K.R. MANGALAM UNIVERSITY

SCHOOL OF ENGINEERING AND TECHNOLOGY

MCA(Master in Computer Application)

Postgraduate Course

2023-25

Preamble

Welcome to the School of Engineering and Technology at K. R. Mangalam University!

At the forefront of innovation and academic excellence, the School of Engineering and Technology is a vibrant hub of learning that nurtures aspiring engineers and technologists. Our commitment to fostering a dynamic learning environment, coupled with a passion for pushing the boundaries of knowledge, empowers our students to embark on a transformative educational journey.

With a blend of cutting-edge curriculum, state-of-the-art facilities, and a distinguished faculty, we are dedicated to equipping our students with the skills, insights, and practical experience they need to thrive in a rapidly evolving technological landscape. Our programs are designed not only to impart technical proficiency but also to cultivate critical thinking, creativity, and ethical leadership.

As we embrace the spirit of innovation and discovery, we invite students to engage in hands-on projects, collaborative research endeavors, and experiential learning opportunities. Through industry partnerships, internships, and exposure to real-world challenges, our students gain a holistic understanding of their fields, preparing them to make meaningful contributions to society.

At the School of Engineering and Technology, we believe in fostering a sense of community and camaraderie among students, faculty, and industry professionals. This collaborative ethos encourages the exchange of ideas, the pursuit of excellence, and the development of lifelong connections.

Whether you aspire to be a trailblazing engineer, a tech-savvy entrepreneur, or a visionary researcher, the School of Engineering and Technology is here to nurture your ambitions and empower you to shape a brighter future.

Welcome to a place where innovation knows no bounds, and where your journey towards academic and professional success begins.

Dean,

School of Engineering and Technology,

K. R. Mangalam University.

Preface

The MCA program is designed to equip students with a comprehensive understanding of computer systems, software development, and emerging technologies. In today's digital age, computer applications have become integral to every aspect of our lives, revolutionizing industries, businesses, and communication. The MCA program aims to empower students with the necessary knowledge and expertise to contribute to this rapidly evolving field.

This comprehensive curriculum combines theoretical foundations with hands-on practical experience, enabling students to develop a strong foundation in programming languages, algorithms, database management, networking, and software engineering. Through a blend of classroom lectures, practical assignments, and industry collaborations, students will be challenged to think critically, solve complex problems, and develop innovative solutions.

In addition to academic rigor, we emphasize the importance of holistic development and encourage students to engage in extracurricular activities, join clubs, and participate in conferences and workshops. These experiences will help you develop leadership skills, foster teamwork, and enhance your communication abilities — all essential qualities for a successful career in the field of computer applications.

We are proud to offer state-of-the-art infrastructure and well-equipped laboratories that provide hands-on experience with the latest tools and technologies. Additionally, our university fosters a vibrant learning environment, with access to extensive libraries, online resources, and industry partnerships that offer opportunities for internships and placements.

As you embark on this exciting journey, we encourage you to make the most of your time here, immerse yourself in the learning process, and actively engage with your peers and faculty. The MCA program will not only equip you with technical skills but also instill in you a passion for lifelong learning and a drive for continuous professional development.

Objectives of the program

After the completion of the degree, students would

- The objectives of a Master's in Computer Application (MCA) program can vary depending on the specific institution or program. However, here are some common objectives that are typically associated with an MCA program:

- Technical Knowledge: The MCA program aims to provide students with a deep understanding of computer science principles, theories, and concepts. It equips them with the technical skills necessary to design, develop, and implement software solutions for various domains.

- Programming and Software Development: The program focuses on developing proficiency in programming languages, software engineering methodologies, and best practices. Students learn to analyze complex problems, design efficient algorithms, and develop robust software applications.

- System Analysis and Design: MCA programs emphasize the importance of system analysis and design techniques. Students learn to gather requirements, analyze system specifications, and design appropriate software architectures. They develop skills in creating system models, designing databases, and ensuring system scalability and security.

- Emerging Technologies: The MCA program keeps students updated with the latest advancements in the field of computer applications. It familiarizes them with emerging technologies such as artificial intelligence, machine learning, data analytics, cloud computing, and Internet of Things (IoT). Students gain hands-on experience with these technologies and understand their practical applications.

- Problem Solving and Critical Thinking: The program aims to enhance students' problem-solving abilities and critical thinking skills. They are exposed to various real-world scenarios and learn to apply their knowledge to identify and solve complex problems in computer applications. Students develop analytical thinking and decision-making skills.

- Communication and Teamwork: MCA programs recognize the importance of effective communication and teamwork in the software development industry. Students are encouraged to collaborate on group projects, participate in discussions, and present their ideas effectively. They develop interpersonal skills, leadership abilities, and learn to work efficiently as part of a team.

- Professional Ethics and Practices: The MCA program emphasizes ethical conduct and professional practices in the field of computer applications. Students learn about legal and ethical considerations related to software development, privacy, and security. They understand the importance of maintaining professional integrity and adhering to industry standards.

- Research and Innovation: Some MCA programs encourage students to engage in research activities and foster an environment of innovation. Students may have opportunities to participate in research projects, publish papers, and contribute to advancements in computer science.

These objectives collectively aim to prepare students for a successful career in computer applications, whether in software development, system analysis, database management, or other related roles. They provide a strong foundation for students to adapt to the ever-changing technological landscape and make valuable contributions to the industry.

Career Avenues

A Master's in Computer Application (MCA) opens up a wide range of career avenues in the field of computer science and technology. Here are some popular career paths that MCA graduates can pursue:

Software Developer/Engineer
System Analyst
Database Administrator
Network Administrator
IT Consultant
Data Scientist/Analyst
Project Manager
Entrepreneurship

These are just a few examples of the diverse career paths available to MCA graduates. The field of computer applications is dynamic and constantly evolving, providing ample opportunities for MCA professionals to contribute to various industries and make a meaningful impact.

Prospective Companies

Microsoft
Google
Amazon
Apple
IBM
Infosys
TCS (Tata Consultancy Services)
Wipro
Cognizant
Accenture.

Duration

2 Years (Full-Time)

Eligibility Criteria

The candidate required to have a bachelor's degree in a relevant field of study, such as Computer Science, Information Technology, or Computer Applications. The degree should be from a recognized university or institution with a minimum of 50% marks in aggregateThe reservation and relaxation for SC/ST/OBC/PWD and other categories shall be as per the rules of central/state government, whichever is applicable.

Table of Contents

University Vision & Mission

Vision

KR Mangalam University aspires to become an internationally recognized institution of higher learning through excellence in inter-disciplinary education, research and innovation, preparing socially responsible life-long learners contributing to nation building.

Mission

- Foster employability and entrepreneurship through futuristic curriculum and progressive pedagogy with cutting-edge technology.
- Instill notion of lifelong learning through stimulating research, Outcomes-based education and innovative thinking;
- Integrate global needs and expectations through collaborative programs with premier universities, research centers, industries and professional bodies;
- Enhance leadership qualities among the youth having understanding of ethical values and environmental realities;

School Vision & Mission

Vision

To create, disseminate, and apply knowledge in science and technology to meet the higher education needs of India and the global society, To serve as an institutional model of excellence in scientific and technical education characterized by integration of teaching, research and innovation.

Mission

- To create an environment where teaching and learning are prioritised, with all support activities being held accountable for their success.
- To strengthen the institution's position as the school of choice for students across the State & Nation.
- To promote creative, immersive, and lifelong learning skills while addressing societal concerns.
- To promote co- and extra-curricular activities for over-all personality development of the students.
- To promote and undertake all-inclusive research and development activities.
- To instill in learners an entrepreneurial mindset and principles.
- Enhance industrial, institutional, national, and international partnerships for symbiotic relationships.
- To help students acquire and develop knowledge, skills and leadership qualities of the 21st Century and beyond.

School Vision & Mission

Vision

To create, disseminate, and apply knowledge in science and technology to meet the higher education needs of India and the global society, to serve as an institutional model of excellence in scientific and technical education characterized by integration of teaching, research and innovation.

Mission

- To create an environment where teaching and learning are prioritised, with all support activities being held accountable for their success.
- To strengthen the institution's position as the school of choice for students across the State & Nation.
- To promote creative, immersive, and lifelong learning skills while addressing societal concerns.
- To promote co- and extra-curricular activities for over-all personality development of the students.
- To promote and undertake all-inclusive research and development activities.
- To instill in learners an entrepreneurial mindset and principles.
- Enhance industrial, institutional, national, and international partnerships for symbiotic relationships.
- To help students acquire and develop knowledge, skills and leadership qualities of the 21st Century and beyond.

About School

The School of Engineering and Technology at K. R. Mangalam University started in 2013 to create a niche of imparting quality education, innovation, entrepreneurship, skill development and creativity. It has excellent infrastructure, state of the art Labs, and a team of qualified and research-oriented faculty members.

School of Engineering & Technology (SOET) brings together outstanding academicians, industry professionals, and experienced researchers to impart hands-on and multi-disciplinary learning experience. The curriculum of the programs caters to the ever-changing needs and demands of industry. The school has state-of-the-art infrastructure and domain-specific labs.

The school is offering undergraduate programs (B. Tech, BCA, B. Sc), postgraduate programs (M. Tech, MCA) and Ph. D (all disciplines). We are offering B. Tech programs in recent areas of specializations like AI & ML, Data Science, Cyber Security, Automotive Designs & Electrical Vehicle, Sustainable Development & Smart Cities, Full stack development, UI/UX development etc.

Our curriculum, being one of our highlights, has been designed in line with the requirements of the new National Education Policy 2020, Pedagogy of Employment, Sustainable Development Goals, IR 4.0 etc. The curriculum focuses on problem-solving, design, development, and application of various emerging technologies with a focus on innovative teaching learning methodologies. It is our endeavor to constantly evolve curriculum support, so our students stay abreast with the latest updates in this technologically developed world.

SOET aims at transforming the students into competitive engineers with adequate analytical skills, making them more acceptable to potential employers in the country. There is a great focus on experiential & project-based learning with Industry collaborations. Our B. Tech programs are in collaborations with Industries like IBM, Siemens, Samatrix, Xebia, ImaginXP etc. Our students get an opportunity to learn directly by professionals from industry.

Program Outcome (PO)

PO1. Advanced Knowledge: Demonstrate advanced knowledge and understanding of computer science, information technology, and related disciplines to solve complex problems and drive innovation.
PO2. Advanced Problem Solving: Apply analytical, critical thinking, and problem-solving skills to identify, analyze, and solve complex computing problems using advanced algorithms, data structures, and software engineering principles.
PO3. Software Development: Design, develop, and deploy software solutions using advanced programming languages, frameworks, and software engineering practices, considering performance, scalability, and security requirements.
PO4. Database Management: Design and manage complex databases, leveraging advanced database management systems and techniques to ensure efficient data storage, retrieval, and manipulation.
PO5. Web and Mobile Applications: Develop advanced web and mobile applications, utilizing contemporary technologies, user interface design principles, and industry best practices.
PO6. Network and Security: Design and implement secure network architectures, protocols, and mechanisms to ensure data integrity, confidentiality, and availability in networked environments.
PO7. Data Science and Analytics: Apply data science techniques, statistical analysis, and machine learning algorithms to extract insights from complex datasets and make data-driven decisions.
PO8. Project Management: Demonstrate proficiency in project planning, execution, and management, including requirements gathering, resource allocation, risk assessment, and project documentation.
PO9. Professional Communication: Communicate effectively, both orally and in writing, with technical and non-technical stakeholders, presenting complex information and ideas clearly and concisely.
PO10. Ethical and Legal Considerations: Understand and adhere to ethical and legal guidelines in computing, including privacy, intellectual property rights, and responsible use of technology.
PO11. Research and Innovation: Conduct research, apply research methodologies, and contribute to the advancement of knowledge in the field of computer applications, exploring new technologies, methods, and solutions.
PO12. Leadership and Collaboration: Exhibit leadership skills and collaborate effectively in multidisciplinary teams, demonstrating teamwork, conflict resolution, and effective interpersonal communication.

Program Educational Objectives (PEO)

PEO1 - Demonstrate expertise in computer applications, programming languages, database management, software engineering, and other key areas of computer science.

PEO2 - Apply knowledge and skills in designing, developing, testing, and maintaining computer applications and systems for various industries and domains.

PEO3 - Pursue higher education, research, and development opportunities in computer science, information technology, or related fields.

PEO4 - Exhibit professionalism, leadership, and teamwork skills to effectively communicate, collaborate, and manage projects with ethical and social responsibilities in the globalized digital society.

PEO5 - Adapt to emerging technologies, changing market demands, and lifelong learning to enhance their career and contribute to the sustainable development of the society.

Program Specific Outcomes (PSO)

PSO1 - Demonstrate proficiency in multiple programming languages, database management systems, software engineering tools, and other key areas of computer science through project-based learning, assignments, and assessments.

PSO2 - Apply analytical and problem-solving skills to design, develop, test, and maintain computer applications and systems for various industries and domains using industry-standard practices and technologies.

PSO3 - Pursue advanced studies, research, or development opportunities in computer science, information technology, or related fields by acquiring the necessary knowledge and skills to contribute to the state-of-the-art.

PSO4 - Exhibit professionalism, leadership, and teamwork skills to effectively communicate, collaborate, and manage projects with ethical and social responsibilities in the globalized digital society through experiential learning, extracurricular activities, and industry exposure.

Program Highlights

- Professionally qualified, competent and committed teaching faculty.
- Industry enabled curriculum and training from industry experts.
- Consistent interaction with renowned academicians and experts.
- Emphasis on project-based learning, techno-pedagogy, field projects, research projects, internships, continuous and comprehensive evaluation.
- Access to certification courses, ability & skill development programs, value-added courses besides core curriculum.
- Effective career counselling, guidance and mentoring program to excel in professional and personal spheres of life.
- Special programs for advanced and slow learners with focus on inclusion and student diversity.
- Focus on career progression through training, placements and preparation for higher studies.
- Centre of excellence in AI ,Machine Learning & Data Science

Program Scheme

Semester I

SN	Category	Course Code	Course Title	L	T	P	C
1	Major	ENMM701	Problem Solving and Advanced Programming Concepts	3	1	0	4
2	Major	ENMM703	Data Structures and Algorithms	3	1	0	4
3	Major	ENMM705	Advanced Database Management Systems	3	1	0	4
4	Minor	ENSP701	AI & Machine Learning	4	0	0	4
5	Major	ENMM707	Data Structure and Algorithms Lab	0	0	2	1
6	Major	ENMM751	Problem Solving Lab	0	0	2	1
7	Major	ENMM753	Data Structures and Algorithms Lab	0	0	2	1
8	Minor	ENSP751	AI & Machine Learning lab	0	0	2	1
9	AEC	AEC011	Life Skills for Professionals-I	3	0	0	3
TOTAL				16	3	8	23

Semester II

SN	Category	Course Code	Course Title	L	T	P	C
1	Major	ENMM702	Advanced Computer Organization & Architecture	3	1	0	4
2	Minor		Department Elective -I	4	0	0	4
3	Major	ENMM704	Advanced Web Technologies	3	1	0	4
4	Major	ENMM706	Introduction of Neural Network and Deep Learning	3	1	0	4
5	Minor		Department Elective -I Lab	0	0	2	1
6	Major	ENMM754	Advanced Web Technologies Lab	0	0	2	1
7	Major	ENMM756	Introduction to Neural Networks & Deep Learning Lab	0	0	2	1
8	AEC	AEC012	Life Skills for Professionals-II	3	0	0	3
9	Open Elective		Open Elective I	3	0	0	3
TOTAL				19	3	6	25

Semester III

SN	Category	Course Code	Course Title	L	T	P	C
1	Minor		Department Elective-II	4	0	0	4
2	Minor		Department Elective-III	4	0	0	4
3	Major	ENMM816	Advance Concepts in Software Engineering & Project Management	3	1	0	4
4	SEC	**SEC036**	Competitive Coding Lab	-	-	4	2
5	Minor		Department Elective-II Lab			2	1
6	Minor		Department Elective-III lab			2	1
7	VAC		Value Added Course	2	0	0	2
8	Open Elective		Open Elective II	3	0	0	3
9	INT	SIMM851	Summer Internship /Minor Project				2
TOTAL				16	1	8	23

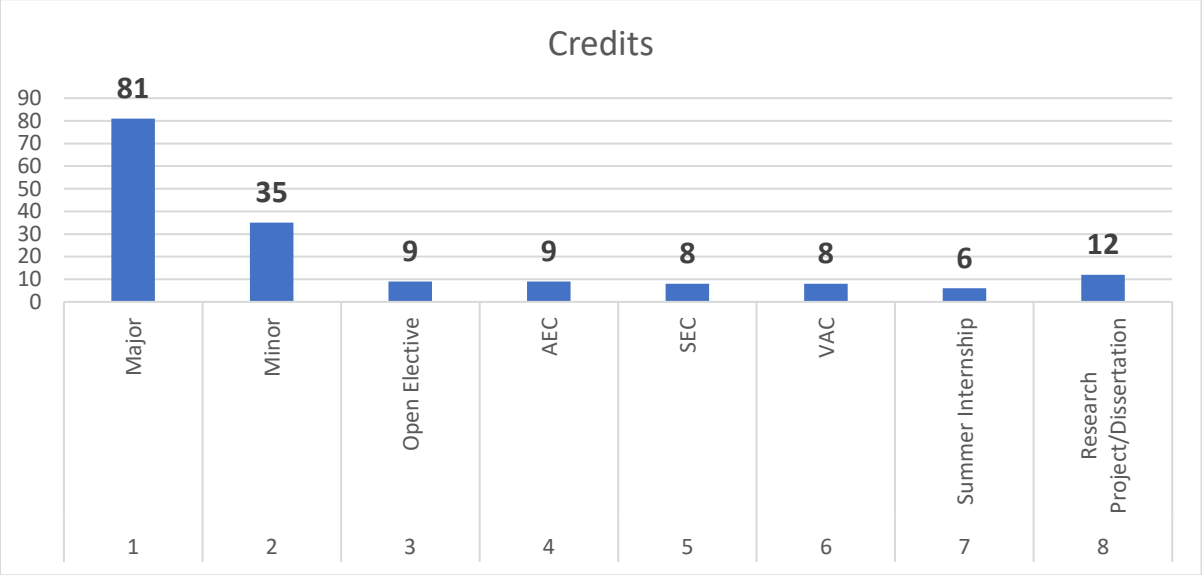
Semester IV

SNo		Course Code	Course Title	L	T	P	C
1	PROJ	SIMM852	Industrial Project/R&D Project/Start-up Project	-	-	-	12
TOTAL				0	0	0	12

Total Credits: 83

The program enables multiple exits & entry options for students as per the guidelines of NEP 2020

Categorization of Courses



Semester: 1

Syllabus

PROBLEM SOLVING AND ADVANCED PROGRAMMING CONCEPTS

Department:	**Department of Computer Science and Engineering**		
Course Name: **Problem Solving and Advanced Programming Concepts**	**Course Code**	**L-T-P**	**Credits**
	ENMM701	3-1-0	4
Type of Course:	**Programme Core** / Programme Elective /Open Elective		
Pre-requisite(s), if any: -			
Brief Syllabus: This subject contains basics of Python programming, data types and formats. Syllabus also includes conditional and course structures and functions and recursion. Various problems that can be solved using recursion like Recursive Fibonacci, Tower of Hanoi etc. Students will able to learn classes and objects in python and how to handle them. Then the final unit will be Introduction to regression and classification and visualization in Python.			
UNIT WISE DETAILS			
Unit Number: 1	**Title:** Basics of Python programming	**No. of hours: 8**	
Content Summary: Python Features, Local Environment Setup, Installing Python, Setting up PATH, Keywords, Understanding Variables, Data Types: Scalar Types, Sequence Type, Mapping Type, Set Types, Mutable Types: List, Manipulation, operation and slicing on Lists, Dictionary, Set, Immutable Types: Numbers, String, tuple, Operators: Arithmetic, Assignment, Comparison, Logical, Identity, Membership, Bitwise, Basic of String, Manipulating strings, Modify Strings, String Concatenation, Format – Strings, Escape Characters, Inbuilt method of Strings, Basic of Regular Expressions, crowd computing			

Unit Number: 2	**Title:** Condition, Control Structures and Functions	**No. of hours: 4**

Content Summary:
Condition: If, If. Else and nested if, Loops: For, while loops, Nested loops, Enumerate, Break, Continue Statement. Sequence and Iterable Objects, Randomization. Function calls, type conversion and coercion, math functions, adding new function, parameters and argument, recursion and its use.

Unit Number: 3	**Title:** Objects and Classes	**No. of hours: 8**

Content Summary:
Basics of Object-Oriented Programming, Creating Class and Object, Constructors in Python – Parameterized and Non-parameterized, Inheritance in Python, In built class methods and attributes, Multi-Level and Multiple Inheritance, Method Overriding and Data Abstraction, Encapsulation and Polymorphism. Exception handling.

Unit Number: 4	**Title**: Python libraries and Database connectivity	**No. of hours: 4**

Content Summary:
Numpy, Pandas ,Data visualization: Introduction to Matplotlib & Seaborn Libraries, basic plotting, various charts .Introduction to Databases ,Connecting to Databases using Python (e.g., SQLite, MySQL)

Unit Number: 5	Classification Techniques	**No. of hours: 8**

Content Summary:

Regression Analysis: Linear & Multiple Linear Regression , Classification: Neural network, Logistic Regression, K-Nearest-Neighbour Classifiers.

Mini Project: Students to work on good size dataset, apply data preprocessing and cleaning, write data retrieval queries, apply various data exploration & visualization techniques.

Self-Learning Component
Students are expected to gain working knowledge of various IDE like Anaconda, Pytorch, Spider etc various components of Anaconda Navigator. Use Github, Kaggle, Google Colab etc

**students will demonstrate the self-learning components through classroom presentations*

Please Note:
1)Students are supposed to learn the components on self-basis
2)Mention open-source tools/ new concepts/technologies that students will be

required to learn and present through presentations in class **3) At least 5-10 % syllabus will be asked in end term exams from self-learning components**
Reference Books: **Text Books:** 1. John V Guttag. "Introduction to Computation and Programming Using Python", Prentice Hall of India **Reference Books** 1. R. Nageswara Rao, "Core Python Programming", Dreamtech 2. Wesley J. Chun. "Core Python Programming, Second Edition", Prentice Hall

Define Course Outcomes (CO)

COs	Statements
CO1	Acquire an **understanding** of Python programming language features, working environment. **Demonstrate** proficiency in Python syntax and effectively utilize keywords in programming.
CO2	**Analyze** and **solve** programming problems that require the use of conditional statements and looping structures
CO3	**Apply** the concepts of functions, recursion & OOPS in problem solving
CO4	Be able to **implement & demonstrate** the concepts of data Pre-processing and data visualization techniques on real datasets.
CO5	**Involve** in self-Learning & be **able to develop** mini-projects in python

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels©	Affective levels(A)	Psychomotor levels(P)
	1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C1		P1, P2

C02	C2		P3
CO3	C3		P3
CO4	C6		P3
CO5	C6		P3

Program Outcome

PO1. Advanced Knowledge: Demonstrate advanced knowledge and understanding of computer science, information technology, and related disciplines to solve complex problems and drive innovation.
PO2. Advanced Problem Solving: Apply analytical, critical thinking, and problem-solving skills to identify, analyze, and solve complex computing problems using advanced algorithms, data structures, and software engineering principles.
PO3. Software Development: Design, develop, and deploy software solutions using advanced programming languages, frameworks, and software engineering practices, considering performance, scalability, and security requirements.
PO4. Database Management: Design and manage complex databases, leveraging advanced database management systems and techniques to ensure efficient data storage, retrieval, and manipulation.
PO5. Web and Mobile Applications: Develop advanced web and mobile applications, utilizing contemporary technologies, user interface design principles, and industry best practices.
PO6. Network and Security: Design and implement secure network architectures, protocols, and mechanisms to ensure data integrity, confidentiality, and availability in networked environments.
PO7. Data Science and Analytics: Apply data science techniques, statistical analysis, and machine learning algorithms to extract insights from complex datasets and make data-driven decisions.
PO8. Project Management: Demonstrate proficiency in project planning, execution, and management, including requirements gathering, resource allocation, risk assessment, and project documentation.
PO9. Professional Communication: Communicate effectively, both orally and in writing, with technical and non-technical stakeholders, presenting complex information and ideas clearly and concisely.
PO10. Ethical and Legal Considerations: Understand and adhere to ethical and legal guidelines in computing, including privacy, intellectual property rights, and responsible use of technology.
PO11. Research and Innovation: Conduct research, apply research methodologies, and contribute to the advancement of knowledge in the field of computer applications, exploring new technologies, methods, and solutions.

PO12. Leadership and Collaboration: Exhibit leadership skills and collaborate effectively in multidisciplinary teams, demonstrating teamwork, conflict resolution, and effective interpersonal communication.

CO-PO Mapping

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	-	-	-	-	1	-	-	1	1	-	-	3
CO2	-	1	1	1	2	-	-	1	1	-	-	3
CO3	-	2	2	1	2	-	-	1	1	-	-	3
CO4	-	2	2	2	3	2	-	1	1	-	-	3
CO5	-	3	3	3	3	3	3	2	3	2	3	3

Please Note:

- Refer to POs while mapping each CO.
- Mark " – " if not applicable
- If attainment of a CO is strongly mapped with a PO, Mark 3
- If attainment of a CO is moderately mapped with a PO, Mark 2
- If attainment of a CO is weakly mapped with a PO, Mark 1

Justification for mapping must be relevant

1=weakly mapped
2= moderately mapped
3=strongly mapped

CO-PSO Mapping

PO	PSO1	PSO2	PSO3	PSO4
CO1	3	1	-	1
CO2	3	1	-	1
CO3	3	2	-	1
CO4	3	2	1	2
CO5	3	3	3	3

Relevance of the Syllabus to various indicators

Unit I	Basics of Python programming
Local	All topics in the syllabus are relevant at the local level as they provide a foundational understanding of Python

	programming.
Regional	All topics in the syllabus can be applicable at the regional level, where individuals and organizations in a specific region utilize Python for various purposes.
National	All topics in the syllabus can be applicable at the national level, as Python is widely used across different industries and sectors.
Global	All topics in the syllabus can be applicable at the global level, as Python is a popular programming language used worldwide.
Employability	All topics in the syllabus, especially Python features, syntax, data types, and string manipulation, contribute to enhancing employability as Python is widely used in the job market.
Entrepreneurship	Understanding Python features, syntax, and data types can be beneficial for entrepreneurs who want to develop software solutions or build web applications.
Skill Development	All topics in the syllabus contribute to skill development in Python programming, including understanding variables, data types, operators, and string manipulation.
Professional Ethics	Professional ethics are not directly addressed in the syllabus topics but are important for ethical behaviour in any professional field.
Gender	NA
Human Values	Human values are not directly addressed in the syllabus topics but can be integrated into the teaching and learning process to emphasize the importance of ethical and responsible programming practices.
Environment & Sustainability	NA
Unit II	
Local	All topics are relevant at local level as programming problems of local nature can be solved using conditional & looping constructs in problem solving
Regional	All topics are relevant at regional level as programming problems of Regional level may utilize conditional & looping constructs in problem solving
National	All topics are relevant at National level as programming problems of national level may utilize conditional & looping constructs in problem solving
Global	All topics are relevant at global level as programming problems of global level may utilize conditional & looping constructs in problem solving
Employability	Conditional & Looping structures are very useful programming tools in problem solving questions during placement drives

Entrepreneurship	Conditional & Looping structures are very useful programming tools in problem solving, any startup involving programming concepts will be useful.
Skill Development	Conditional & Looping structures are very useful programming tools in problem solving and thus skill development.
Professional Ethics	Writing programming codes/solutions will always use conditional & looping structures. There are well defined professional ethics for writing codes.
Gender	NA
Human Values	Any computational solution created with applications for humankind will always involve these programming constructs.
Environment & Sustainability	All computational solutions mapped with environment & sustainability will somehow involve these constructs at the backend.
Unit III	
Local	All topics are relevant at local level as programming solutions at local nature can always use concepts of functions & recursions in problem solving
Regional	All topics are relevant at regional level as programming solutions at regional nature can always use concepts of functions & recursions in problem solving
National	All topics are relevant at national level as programming solutions at national level can always use concepts of functions & recursions in problem solving
Global	All topics are relevant at global level as programming solutions at global nature can always use concepts of functions & recursions in problem solving
Employability	Topics are quite relevant for placements
Entrepreneurship	Working Knowledge of topics can be a advantage for entrepreneurs
Skill Development	Topics can be quite useful for skill development on Programming
Professional Ethics	Writing codes involving functions can be related to coding ethics
Gender	NA
Human Values	NA
Environment & Sustainability	Related applications may be mapped with environment and sustainability
Unit IV	
Local	NA
Regional	Regional reference is possible
National	National reference is possible
Global	This unit will effect globally as global applications can be

	developed
Employability	Skills attained for developing applications using pandas, data visualizations have direct mapping with employability
Entrepreneurship	Mapping of this aspect is also possible. Start up problems can be using these learning aspects
Skill Development	Useful for skill development on regression and classification.
Professional Ethics	NA
Gender	NA
Human Values	NA
Environment & Sustainability	Applications/problem handling based on environment & sustainability issues
Unit V	
Local	Solutions involving Descriptive analytics using data visualization techniques can have a local reference.
Regional	Regional reference is possible
National	National reference is possible
Global	Global aspect is also possible
Employability	Skills attained for developing applications using pandas, data visualizations have direct mapping with employability
Entrepreneurship	Mapping of this aspect is also possible. Startup problems can be using these learning aspects
Skill Development	Topics can be quite useful for skill development on descriptive analytics
Professional Ethics	descriptive analytics is related to presenting thoughts & values in ethical manner
Gender	NA
Human Values	NA
Environment & Sustainability	Applications/problem handling based on environment & sustainability issues

DATA STRUCTURE AND ALGORITHMS

Department:	**Department of Computer Science and Engineering**		
Course Name: Data Structure and Algorithms	**Course Code**	**L-T-P**	**Credits**
	ENMM703	4-0-2	4
Type of Course:	Major		
Pre-requisite(s), if any: Basics of Computer Programming			

Brief Syllabus:
Solving computational problems requires the knowledge of efficient data organization and the ability to make effective choices among multiple solutions. In this course, we will explore several fundamental data structures in computer science and learn to implement them. The course aims to teach the fundamentals of data structures, their design, implementation and effective use in problem solving approach. With the knowledge of data structures and practical experience in implementing them, students can become much more effective designer and developer. The course will start with the basic introduction of linear such as arrays, stack and queues as well as non-linear data structures such as trees and graphs. They will further proceeds with the programming intensive task of implementing them.

UNIT WISE DETAILS

Unit Number: 1	**Title: Introduction to Data Structure**	**No. of hours: 8**

Content Summary:
Introduction to Data Structures: Definition of data structures and abstract data types, Static and Dynamic implementations, Examples and real life applications; Arrays: ordered lists, representation of arrays in memory
Basic Analysis: Differences among best, average, and worst case behaviours of an algorithm, Asymptotic analysis of upper and expected complexity bounds, Big O notation: formal definition and use, big omega and big theta notation , Complexity classes, such as constant, logarithmic, linear, quadratic, and exponential, Time and space trade-offs in algorithms, Recurrence relations , Analysis of iterative and recursive algorithms.

Unit Number: 2	**Title: Stacks, Queues and Linked List**	**No. of hours: 12**

Content Summary:

Stacks: ADT Stack and its operation, Array based implementation of stacks, Examples: Infix, postfix, prefix representation, Conversions of an arithmetic expression from Infix to postfix, Applications: Tower of Hanoi Problem, Algorithms and their complexities

Queues: ADT Queue and its operation, Array based implementation of linear Queues, Circular Queues, Priority queues

Linked List: Definition, Components of linked list, Representation of linked list, Advantages and Disadvantages of linked list. Types of linked list: Singly linked list, Doubly linked list, Circular linked list and circular doubly linked list. Operations on different types of linked list : creation, insertion, deletion, search and display (based on the different position as specified by the user).Linked representation of Stacks & Queues, Algorithms and their complexities

Unit Number: 3	**Title: Trees and Graphs**	**No. of hours: 12**

Content Summary:

Trees: Basic Terminology, Binary Trees and their representation, expression evaluation, Complete Binary trees, Extended binary trees, traversing binary trees, Searching, Insertion and Deletion in binary search trees. AVL trees, Threaded trees, algorithms and their analysis.

Graphs: Terminology and Representations, Graphs & Multigraphs, Directed Graphs, Sequential representation of graphs, Adjacency matrices, Transversal Connected Component and Spanning trees, algorithms and their analysis.

Unit Number: 4	**Title: Sorting ,Searching and File Structure**	**No. of hours: 10**

Content Summary:

Sorting Algorithms: Introduction, Sorting by exchange, selection sort, insertion sort, Bubble sort, Selection sort, Efficiency of above algorithms, Shell sort, Performance of shell sort, Merge sort, Quick sort Algorithm analysis, Heap sort: Heap Construction, Heap sort, bottom – up, Top – down Heap sort approach.

Searching Algorithms: Straight Sequential Search, Binary Search (recursive & non–recursive Algorithm.

File structures: Physical storage media, File Organization, Linked organization of file, Inverted file, Organization records into blocks, Sequential blocks, Hash function, Indexing & Hashing

***Self-Learning Components:**

1. Students should explore Platforms like LeetCode, HackerRank for Data structure
2. Students can refer the following courses as per the **Open Source University**

Curriculum "Algorithms, Part I" by Robert Sedgewick and Kevin Wayne (available on Coursera) "Algorithms, Part II" by Robert Sedgewick and Kevin Wayne (available on Coursera)
Reference Books: 1. E. Horowitz and S. Sahani, "Fundamentals of Data Structures", Galgotia Book source Pvt. Ltd. 2. Data Structures & Algorithms in Python by John Canning, Alan Broder, Robert Lafore Addison-Wesley Professional. 3. "Introduction to Algorithms" by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. 4. Problem Solving with Algorithms and Data Structures Using Python" by Brad Miller and David Ranum.

Define Course Outcomes (CO)

COs	Statements
CO1	**Analyze** the algorithms to determine the time and space complexity and justify the correctness.
CO2	**Understand, Design & Implement** a given Search problem (Linear Search and Binary Search).
CO3	**Articulate** Data Structure concepts such as Stack, Queue, Linked list, Hashing ,Graph and traversal techniques and their use in programs
CO4	**Understand, Design & implement** the algorithm for Selection Sort, Bubble Sort, Insertion Sort, Quick Sort, Merge Sort, Heap sort. Compare their performance in term of Space and time complexity

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels© 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C1,C2	A3	P5
C02	C3	A3	P4

CO3	C4	A4	P3
CO4	C5	A2	P2

***Please Note:**
Map only 1 or 2 Levels in each category. If a higher level is given, no need to mention lower level

CO-PO Mapping

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	2	-	-	1	-	-	-	-	2	-
CO2	3	3	2	-	-	-	-	-	-	-	2	-
CO3	3	3	3	-	-	-	-	-	-	-	2	-
CO4	3	3	3	-	-	-	-	-	-	-	2	-

Please Note:

- Refer to POs while mapping each CO.
- Mark " – " if not applicable
- If attainment of a CO is strongly mapped with a PO , Mark 3
- If attainment of a CO is moderately mapped with a PO , Mark 2
- If attainment of a CO is weakly mapped with a PO , Mark 1

Justification for mapping must be relevant

1=weakly mapped
2= moderately mapped
3=strongly mapped

CO-PSO Mapping

PSO	PSO1	PSO2	PSO3	PSO4
CO1	3	3	2	-
CO2	3	3	1	-
CO3	3	3	-	-
CO4	3	3	-	-

Relevance of the Syllabus to various indicators

Unit I	**Introduction to Data Structure**
Local	-
Regional	-
National	It provides foundational knowledge in data structures and algorithm analysis, which are fundamental oncepts in computer science and software engineering.
Global	The principles taught in this course are applicable worldwide and form the basis of software engineering practices globally.
Employability	Understanding these concepts is crucial for technical interviews and can enhance job prospects in various technology companies.
Entrepreneurship	It helps in developing efficient and scalable software solutions, which are essential for building successful tech startups or innovative ventures.
Skill Development	Develop skills that are fundamental to computer science and software development and can be applied in various programming languages and contexts.
Professional Ethics	Applying good coding practices and software engineering principles align with professional ethics in the field.
Gender	-
Human Values	It encourages students to approach problems analytically and develop efficient solutions that can positively impact human lives.
Environment & Sustainability	-
Unit II	**Stacks, Queues and Linked List**
Local	-
Regional	-
National	These data structures are used extensively in computer science and software engineering, and the skills learned in this course can be applicable to various industries and sectors across the country.
Global	The principles taught in this course are applicable worldwide and form the basis of software engineering practices globally.
Employability	Understanding these data structures and their

	operations is important for solving problems efficiently and implementing optimized algorithms, which are highly sought after skills in the job market.
Entrepreneurship	Knowledge of data structures like stacks, queues, and linked lists is valuable for entrepreneurship in the technology sector.
Skill Development	Understanding and applying concepts related to stacks, queues, and linked lists enhances programming skills and helps in developing efficient algorithms to solve real-world problems.
Professional Ethics	-
Gender	-
Human Values	Fostering critical thinking, problem-solving skills, and logical reasoning, which are important qualities in a technology-driven society which can improve productivity and streamline processes, thus positively impacting human lives.
Environment & Sustainability	-
Unit III	**Trees and Graphs**
Local	-
Regional	-
National	The skills learned in this course can be applicable to various industries and sectors across the country.
Global	The principles taught in this course are applicable worldwide and form the basis of software engineering practices globally.
Employability	Knowledge of data structures such as trees and graphs is highly relevant to employability in the field of software development and computer science.
Entrepreneurship	These data structures are commonly used in designing and developing software solutions, and understanding their implementation and applications can help entrepreneurs build innovative and scalable products.
Skill Development	Understanding and applying concepts related to trees and graphs enhances programming skills and helps in developing efficient algorithms to solve real-world problems.
Professional Ethics	Following best practices in data structure implementation and algorithm design promotes code readability, maintainability, and overall software

	quality.
Gender	-
Human Values	Understanding data structures like trees and graphs enables students to develop efficient algorithms that can improve productivity, streamline processes, and positively impact human lives.
Environment & Sustainability	-
Unit IV	**Sorting , Searching and File Structures**
Local	-
Regional	-
National	It provides foundational knowledge in sorting and searching algorithms.
Global	Sorting and searching algorithms are fundamental building blocks in computer science and software development, used globally.
Employability	Understanding these algorithms and their efficiency helps in developing optimized software solutions, which are highly sought-after skills in the job market.
Entrepreneurship	These algorithms are used extensively in data processing, information retrieval, and optimization problems, which are essential in building innovative and scalable software products.
Skill Development	Understanding and applying sorting and searching algorithms enhances programming skills and helps in developing efficient algorithms to solve real-world problems.
Professional Ethics	Following best practices in algorithm design and implementation promotes code readability, maintainability, and overall software quality.
Gender	-
Human Values	Understanding sorting and searching algorithms enables students to develop efficient solutions that improve productivity, streamline processes, and positively impact human lives.
Environment & Sustainability	-
SDG	SDG 9
NEP 2020	-
POE/4th IR	Aligns with the concepts of Design, Efficiency ,Problem Solving, Abstraction and System Analysis

Benchmarking Universities (atleast 4-5 standard university contents must be reffered):

Name of the University	Program referred for Syllabus contents	Subject which is referred
Amity University Haryana	MCA	Data structure
Benett University	MCA	Data structure
Open Source University	MCA	Data structure
Jawaharlal Nehru Technological University	MCA	Data structure

ADVANCED DATABASE MANAGEMENT SYSTEMS

Department:	**Department of Computer Science and Engineering**		
Course Name: Advanced Database Management Systems	**Course Code**	**L-T-P**	**Credits**
		3-1-0	4
Type of Course:	Programme Core		
Pre-requisite(s), if any:			
Brief Syllabus: This course aims to deliver advanced concepts of database information and management and its implementation. The students will become familiar with the concepts of Transaction Processing and Concurrency Control Techniques, Database Recovery Techniques, Distributed DBMS, Object-Oriented DBMS.			
UNIT WISE DETAILS			
Unit Number: 1	**Title: Database Concurrent Mechanisms**	**No. of hours: 10**	
Content Summary: ER Model, Indexing Dependencies, Transaction processing, Transaction management, Scheduling in Databases, Serial Scheduler, Non-Serial Schedulers, Concurrency Control, Time stamp ordering Protocol, Multi version concurrency control, Validation concurrency control, Recovery Techniques in Databases, Undoing, Deferred Update, Immediate Update, Buffering and Caching in Databases, Design Elements, Cache SQL ResultSet, Cache selected fields and values, Caching serialized applications, Shadow Paging, Data Backup Architecture, Transaction logging, Paging Architecture in Databases, Full Backup, Differential Backup.			
Unit Number: 2	**Title: Distributed DBMS**	**No. of hours: 10**	
Content Summary: Distributed DBMS, Architecture of a DDBMS, Distributed Design of Relational Database, Variant of the 2 Tier Model, Fragmentation and Replication, Update in DDBMS, Update Strategies, Eager update, Distributed Reliability, Phase commit protocol, Asynchronous Update, Distributed transaction management, Distributed concurrency control, Distributed deadlock management.			
Unit Number: 3	**Title: Object-Oriented DBMS**	**No. of hours: 10**	

Content Summary:

Object-Oriented DBMS Architecture, Capturing Semantics, Message Passing, OODB persistence, Object relational database model, Object relational features, Object tables, Nested tables, Varying arrays, Support for large objects, Database Migration, Migration Architecture, Heterogeneous migration versus homogeneous migration, Migration consistency, Database Transformation, Differential Querying, Migration Consistency Verification, Order violation, Consistency violation, Local Transactions

Unit Number: 4	**Title: Database Security**	**No. of hours: 10**

Content Summary:
Flex Cluster Architecture, Policy-Based Cluster, Grid Infrastructure, Kernel Requirements, Data Redaction, Database Auditing, Database Diagnostic Monitor, Global User Authentication, Label Security, Data Masking, Privilege Analysis, Virtual Private Databases.

***Self-Learning Components:**

- MongoDB
- PostgreSQL

Note: Students will give presentations or submit projects based on the self-learning components for evaluation.

Reference Books:

1. Pankaj B. Brahmankar, Advanced Database Management Systems (1 ed.), Tech Neo Publications, 2019. ISBN 9788194154600.
2. Soudamini Patil, Narendra Joshi, Vrushali R. Sonar, Umesh M. Patil, Yogesh B. Patil and Rohit A. Kautkar, Advanced Database Management Systems (1 ed.), Technical Publications, 2020. ISBN 9789389180336.

Define Course Outcomes (CO)

COs	Statements
CO1	Recall fundamental concepts, terminology, and principles of database management systems.
CO2	Examine the architecture and features of object-oriented DBMS, such as capturing semantics and supporting object-relational database models
CO3	Evaluate and optimize buffering and caching strategies to improve database performance

CO4	Analyze and evaluate transaction processing, concurrency control and database recovery protocols in databases.
CO5	Design and implement privilege analysis and virtual private databases to control access and ensure data privacy

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels (C) 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C1	A1	P1
C02	C2	A2	P2
CO3	C5	A3	P3
CO4	C4	A4	P4
CO5	C6	A5	P5

***Please Note:**
Map only 1 or 2 Levels in each category. If a higher level is given, no need to mention lower level

CO-PO Mapping

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	1	1	3	1	1	1	1	1	1	1	1
CO2	2	1	1	2	1	1	1	1	1	1	1	1
CO3	1	2	1	2	1	1	1	1	1	1	1	1
CO4	2	2	1	3	1	1	1	1	1	1	1	1
CO5	1	1	1	3	1	2	1	1	1	2	1	1

Please Note:

- Refer to POs while mapping each CO.
- Mark " – " if not applicable

- If attainment of a CO is strongly mapped with a PO , Mark 3
- If attainment of a CO is moderately mapped with a PO , Mark 2
- If attainment of a CO is weakly mapped with a PO , Mark 1

Justification for mapping must be relevant

1=weakly mapped
2= moderately mapped
3=strongly mapped

CO-PSO Mapping

CO	PSO1	PSO2	PSO3	PSO4
CO1	2	2	1	1
CO2	2	3	1	1
CO3	2	2	1	2
CO4	3	2	2	1
CO5	3	3	1	1

Relevance of the Syllabus to various indicators

Unit I	**Database Concurrent Mechanisms**
Local	The syllabus topics are relevant at the local level as they provide knowledge and skills in database management, transaction processing, concurrency control, and recovery techniques. Local businesses and organizations often use databases for their operations, and professionals with expertise in these areas are in demand.
Regional	The relevance of the syllabus topics extends to the regional level as well. Many regional businesses and industries rely on databases for storing and managing their data. Knowledge of ER modeling, indexing, transaction management, and concurrency control is essential for efficient and secure data processing in these settings.
National	At the national level, the syllabus topics become even more relevant. Database management systems are widely used across various industries and sectors throughout the country. Proficiency in these topics, including transaction processing, recovery techniques, and caching strategies, is crucial for ensuring reliable and scalable data management on a national scale.
Global	In the global context, the syllabus topics hold significant

	relevance. With the increasing digitalization and globalization of businesses, databases play a critical role in managing and processing large amounts of data across international borders. Understanding concepts like transaction management, concurrency control, and recovery techniques becomes essential for organizations operating globally.
Employability	The syllabus topics are highly relevant to employability. Proficiency in database management systems and related concepts enhances the employment prospects of individuals in various sectors, including IT, software development, data analysis, and database administration. These skills are sought after by employers who rely on efficient and secure data management.
Entrepreneurship	Knowledge of database management systems and related topics can be valuable for entrepreneurs. Understanding how to design and manage databases, optimize performance, and ensure data privacy and security can help entrepreneurs build and scale their own businesses, especially those reliant on data-driven applications or services.
Skill Development	The syllabus topics contribute to skill development by providing knowledge and practical skills in database management, transaction processing, concurrency control, and recovery techniques. These skills are highly relevant in the IT industry and contribute to the overall skill development of individuals pursuing careers in this field.
Professional Ethics	Understanding professional ethics is important in the field of database management. Topics such as privacy, data security, and responsible use of technology are relevant in this context. Professionals working with databases need to adhere to ethical guidelines to ensure the integrity and confidentiality of data.
Gender	
Human Values	The syllabus topics indirectly promote human values such as integrity, responsibility, and respect for privacy. These values are essential when dealing with sensitive data and ensuring the ethical use of databases.
Environment & Sustainability	
Unit II	**Distributed DBMS**

Local	The relevance of the syllabus topics at the local level is limited, as distributed database management systems (DDBMS) are more commonly used in larger organizations or those with multiple locations. However, some local businesses with distributed operations may benefit from understanding the concepts of DDBMS and its architecture.
Regional	The relevance of the syllabus topics extends to the regional level. Regional organizations or those operating across multiple locations can benefit from implementing DDBMS to improve data accessibility, scalability, and performance. Understanding the architecture, fragmentation, replication, and update strategies in DDBMS can be valuable for businesses in the region.
National	At the national level, the syllabus topics gain more significance. Large organizations with a national presence often have distributed data centers or multiple branches that require efficient and reliable data management. Knowledge of DDBMS, distributed transaction management, concurrency control, and deadlock management is crucial for ensuring consistent and secure data operations on a national scale.
Global	In the global context, the syllabus topics become highly relevant. Global organizations with distributed operations across different countries or regions need to manage data in a distributed manner. Understanding the architecture, reliability, update strategies, and distributed concurrency control becomes essential for efficient and consistent data management on a global scale.
Employability	Knowledge of DDBMS and related concepts enhances employability in the IT industry. Organizations with distributed data environments seek professionals skilled in designing, implementing, and managing distributed databases. Proficiency in distributed database design, update strategies, transaction management, and concurrency control can significantly increase employability prospects.
Entrepreneurship	Understanding DDBMS and related concepts can be valuable for entrepreneurs looking to build scalable and distributed applications or services. Entrepreneurs can leverage the knowledge of distributed database architecture, fragmentation, replication, and update

	strategies to design and deploy efficient and reliable distributed systems.
Skill Development	The syllabus topics contribute to skill development by providing knowledge and practical skills in distributed database management systems. Understanding the architecture, design, reliability, transaction management, and concurrency control in distributed environments enhances skill development in the field of database management
Professional Ethics	Professional ethics are relevant in the context of distributed databases as well. Topics such as data security, privacy, and responsible data handling are crucial for professionals working with distributed systems. Adhering to ethical guidelines is necessary to ensure the integrity and confidentiality of data in a distributed environment.
Gender	
Human Values	The syllabus topics indirectly promote human values such as integrity, responsibility, and respect for privacy. These values are essential when dealing with distributed data and ensuring ethical practices in managing and securing data across multiple locations.
Environment & Sustainability	
Unit III	**Object-Oriented DBMS**
Local	The relevance of the syllabus topics at the local level is limited, as Object-Oriented DBMS (OODBMS) and related concepts are not as commonly used in local businesses or organizations. However, some local businesses in the IT industry or those dealing with complex data structures may benefit from understanding OODBMS architecture and the object-relational database model.
Regional	The relevance of the syllabus topics extends to the regional level. Regional organizations or those operating in the IT industry can benefit from implementing OODBMS to handle complex data structures and capture semantics. Understanding OODBMS architecture, message passing, and support for large objects can be valuable for businesses in the region.
National	At the national level, the syllabus topics gain more significance. Large organizations in the IT industry, research institutions, and government agencies may require OODBMS capabilities to manage and process

	complex data with a focus on capturing semantics. Knowledge of OODBMS architecture, object-relational features, and database migration can be crucial for ensuring efficient and effective data management on a national scale.
Global	In the global context, the syllabus topics become highly relevant. Global organizations dealing with complex data structures, such as those in scientific research, multimedia, and e-commerce, may rely on OODBMS to handle their data needs. Understanding OODBMS architecture, object-relational features, and database migration becomes essential for efficient and consistent data management on a global scale.
Employability	Knowledge of OODBMS and related concepts can enhance employability in specialized sectors of the IT industry. Organizations working with complex data structures and requiring advanced data modeling capabilities may seek professionals with expertise in OODBMS architecture, object-relational models, and database migration. Proficiency in these topics can increase employability prospects in specific domains.
Entrepreneurship	Understanding OODBMS and its application in handling complex data structures can be valuable for entrepreneurs looking to develop innovative applications or services. Entrepreneurs can leverage the knowledge of OODBMS architecture, object-relational features, and database migration to design and deploy efficient and scalable systems that handle complex data requirements.
Skill Development	The syllabus topics contribute to skill development by providing knowledge and practical skills in OODBMS architecture, object-relational models, and database migration. Understanding these topics enhances skill development in the field of advanced data management and modeling.
Professional Ethics	Professional ethics are relevant in the context of OODBMS as well. Topics such as data security, privacy, and responsible data handling are crucial for professionals working with complex data structures and object-relational models. Adhering to ethical guidelines is necessary to ensure the integrity and confidentiality of data in OODBMS environments.
Gender	

Human Values	The syllabus topics indirectly promote human values such as integrity, responsibility, and respect for privacy. These values are essential when dealing with complex data structures and ensuring ethical practices in managing and securing data in OODBMS environments.
Environment & Sustainability	
Unit IV	**Database Security**
Local	The relevance of the syllabus topics at the local level may vary depending on the specific needs of local businesses and organizations. Flex Cluster Architecture, Policy-Based Cluster, and Grid Infrastructure can be relevant for local businesses that require high availability and scalability in their database systems. Additionally, topics like Data Redaction, Database Auditing, and Privilege Analysis are important for ensuring data security and compliance with local regulations.
Regional	The relevance of the syllabus topics extends to the regional level. Regional organizations in sectors such as finance, healthcare, and government that deal with sensitive data and have complex database requirements can benefit from the implementation of Flex Cluster Architecture, Policy-Based Cluster, and Grid Infrastructure. These topics provide the necessary knowledge to design and deploy robust and scalable database systems at a regional scale.
National	At the national level, the syllabus topics gain further relevance. Large organizations and government agencies with national operations often require advanced database architectures like Flex Cluster Architecture and Policy-Based Cluster to ensure high availability, fault tolerance, and scalability. Additionally, topics like Label Security and Data Masking become important for protecting sensitive data at a national scale.
Global	In the global context, the syllabus topics become highly relevant. Global organizations with distributed operations and complex database requirements can benefit from the concepts covered in the syllabus. Flex Cluster Architecture, Policy-Based Cluster, and Grid Infrastructure enable global scalability and availability of databases. Topics like Global User Authentication and Virtual Private Databases address global authentication and access control needs. Furthermore, Database

	Auditing and Privilege Analysis help organizations maintain data security and compliance across multiple regions.
Employability	Knowledge of these syllabus topics can enhance employability in organizations that require advanced database management skills. Employers seeking professionals with expertise in Flex Cluster Architecture, Policy-Based Cluster, and Grid Infrastructure can value individuals with proficiency in these topics. Additionally, knowledge of Data Redaction, Database Auditing, and Privilege Analysis can be valuable for roles related to data security and compliance.
Entrepreneurship	Understanding the syllabus topics can be valuable for entrepreneurs looking to establish their own database consulting or services businesses. The knowledge of Flex Cluster Architecture, Policy-Based Cluster, and Grid Infrastructure can help entrepreneurs design and implement robust and scalable database solutions for their clients. Moreover, expertise in topics such as Label Security and Data Masking can provide a competitive advantage in offering data security services.
Skill Development	The syllabus topics contribute to skill development by providing knowledge and practical skills in advanced database architectures, security mechanisms, and policy-based management. Understanding these topics enhances skill development in the field of database administration and security.
Professional Ethics	Professional ethics are relevant in the context of database management and security. Topics such as Data Redaction, Database Auditing, and Privilege Analysis address ethical considerations related to data privacy, security, and responsible data management.
Gender	
Human Values	The syllabus topics indirectly promote human values such as integrity, responsibility, and privacy. Concepts like Label Security, Data Masking, and Database Auditing contribute to ensuring the confidentiality and ethical handling of data, aligning with human values.
Environment & Sustainability	
SDG	SDG4, SDG7 and SDG9
NEP 2020	The syllabus covers advanced topics in database management, such as distributed DBMS, object-oriented

	DBMS, and database security. These topics encourage learners to engage in continuous learning and research, fostering a culture of lifelong learning as advocated by the NEP 2020.
POE/4th IR	The mentioned topics and concepts are relevant to the ethical considerations and technological advancements associated with professional ethics and the 4th Industrial Revolution.

Benchmarking Universities (atleast 4-5 standard university contents must be reffered):

Name of the University	**Program referred for Syllabus contents**	**Subject referred**
GGSIPU University	BCA	Advancd Database Management System
BITS University	B.Tech CSE	Database System
Amity University	B.Tech + M.Tech CSE	Database Management System

AI & MACHINE LEARNING

Department:	**Department of Computer Science and Engineering**		
Course Name: AI & Machine Learning	**Course Code**	**L-T-P**	**Credits**
		4-0-1	5
Type of Course:	Minor		
Pre-requisite(s), if any:			

Brief Syllabus:
Artificial Intelligence and Machine Learning is a branch of study or discipline which includes theories, standards, methods and innovations of various different domains like mathematics, cognitive science, electronics and embedded systems to make intelligent systems that mimic human behaviour. Artificial Intelligence (AI) and Machine Learning (ML) focus on collecting, categorizing, strategizing, analyzing and interpreting data. It also incorporates the concepts of machine learning and deep learning model building for solving various computational and real-world business problems.

Unit Number: 1	**Title: : Introduction to AI**	**No. of hours: 08**

Content Summary:
Basic Definitions and Terminology, Foundation and History of AI, Overview of AI problems, Evolution of AI - Applications of AI, Classification/Types of AI. Artificial Intelligence vs Machine learning.

Intelligent Agent: Types of AI Agent, Concept of Rationality, nature of environment, structure of agents. Turing Test in AI.

Unit Number: 2	**Title: Problem Solving, Knowledge and Reasoning**	**No. of hours: 12**

Content Summary:

Search Algorithms in Artificial Intelligence: Terminologies, Properties of search Algorithms, Types of search algorithms: uninformed search and informed search, State Space search Heuristic Search Techniques: Generate-and-Test; Hill Climbing; Properties of A* algorithm, Best-first Search; Problem Reduction.
Constraint Satisfaction problem: Interference in CSPs; Back tracking search for CSPs; Local Search for CSPs; structure of CSP Problem.
Knowledge-Based Agent in Artificial intelligence: Architecture, approaches to designing a knowledge-based agent, knowledge representation: Techniques of knowledge representation, Propositional logic, Rules of

Inference, First-Order Logic, Forward Chaining and backward chaining in AI,
Reasoning in Artificial intelligence: Types of Reasoning and Probabilistic reasoning, Uncertainty.

Unit Number: 3	Title: Introduction to ML	No. of hours: 10

Content Summary:
Introduction to Machine Learning: History of ML Examples of Machine Learning Applications, Learning Types, ML Life cycle, AI & ML, dataset for ML, Data Pre-processing, Training versus Testing, Positive and Negative Class, Cross-validation.

Types of Learning: Supervised, Unsupervised and Semi-Supervised Learning. Supervised: Learning a Class from Examples, Types of supervised Machine learning Algorithms.

Unsupervised: Types of Unsupervised Learning Algorithm, Dimensionality Reduction: Introduction to Dimensionality Reduction, Subset Selection, and Introduction to Principal Component Analysis.

Unit Number: 4	Title: : Classification and Regression	No. of hours: 10

Content Summary:
Binary and Multiclass Classification: Assessing Classification Performance, Handling more than two classes, Multiclass Classification-One vs One, One vs Rest.

Regression: Assessing performance of Regression – Error measures, Overfitting and Underfitting, Catalysts for Overfitting, VC Dimensions.

Self-Learning Component: - The students are expected to choose a topic in discussion with the industry expert and implement the concepts on a real life dataset. The student should present the progress at the end of the semester.
Students must complete one online course of atleast 40 hrs with certification as an outcome of this course

Reference Books:

1. Russell, S. and Norvig, P. 2015. Artificial Intelligence - A Modern Approach, 3rd edition, Prentice Hall.
2. J. Gabriel, Artificial Intelligence: Artificial Intelligence for Humans (Artificial Intelligence, Machine Learning), Create Space Independent Publishing Platform, First edition , 2016.
3. Introduction to Artificial Intelligence & Expert Systems, Dan W Patterson, PHI., 2010 2. S Kaushik, Artificial Intelligence, Cengage

Learning, 1st ed.2011.

Define Course Outcomes (CO)

COs	Statements
CO1	Evaluate Artificial Intelligence (AI) methods and describe their foundations.
CO2	Analyze and illustrate how search algorithms play vital role in problem solving, inference, perception, knowledge representation and learning.
CO3	Demonstrate knowledge of reasoning and knowledge representation for solving real world problems.
CO4	Recognize the characteristics of machine learning that makes it useful to real-world Problems.
CO5	Apply the different supervised learning methods of support vector machine and tree based models.
CO6	Use different linear methods for regression and classification with their optimization through different regularization techniques.

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels© 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C5	A5	P4
CO2	C4	A4	P4
CO3	C2	A2	P2
CO4	C2	A2	P1

CO5	C3	A3	P3
CO6	C3	A3	P3

CO-PO Mapping

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	3								3	3	
CO2	2	3								2	3	
CO3	2	3								3	3	
CO4	2	2	2	2			2			2	2	
CO5	1	3	2		3	1	3		2	2		
CO6	1	3			2	1				2		

Please Note:

- Refer to POs while mapping each CO.
- Mark " – " if not applicable
- If attainment of a CO is strongly mapped with a PO, Mark 3
- If attainment of a CO is moderately mapped with a PO, Mark 2
- If attainment of a CO is weakly mapped with a PO, Mark 1

Justification for mapping must be relevant

1=weakly mapped
2= moderately mapped
3=strongly mapped

CO-PSO Mapping

PO	PSO1	PSO2	PSO3	PSO4
CO1	1	2	3	1
CO2	2	3	1	1
CO3	1	1	3	2
CO4	1	3	2	1
CO5	1	2	1	1
CO6	1	3	1	2

Relevance of the Syllabus to various indicators

Unit I	**Introduction to AI**
Local	-
Regional	-
National	-
Global	Global cooperation and collaboration in AI research, development, and policies are essential to address global challenges, share knowledge, and ensure responsible and ethical use of AI technologies on a global scale.
Employability	AI technologies are increasingly adopted by industries, and professionals with expertise in AI are in high demand. Skills in AI programming, machine learning, data analysis, and AI ethics make individuals valuable assets in the job market, opening up opportunities for employment and career growth.
Entrepreneurship	-
Skill Development	Studying AI fosters skill development in various domains, including programming, data analysis, machine learning, and problem-solving.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit II	**Problem Solving, Knowledge and Reasoning**
Local	-

Regional	-
National	-
Global	Global industries, research institutions, and technology companies rely on professionals with expertise in search algorithms to develop intelligent systems, optimize processes, and make informed decisions.
Employability	Proficiency in search algorithms in artificial intelligence enhances employability prospects, especially in the fields of data science, machine learning, and artificial intelligence.
Entrepreneurship	-
Skill Development	Studying search algorithms in artificial intelligence develops critical thinking, problem-solving, and algorithmic design skills.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	**Introduction to ML**
Local	-
Regional	-
National	-
Global	The application of intelligent systems can drive advancements in healthcare, education, environmental monitoring, and other domains, contributing to global progress and well-being.
Employability	Acquiring knowledge and skills in knowledge-based agents in artificial intelligence enhances employability prospects, especially in the field of data science, machine learning, and artificial intelligence.
Entrepreneurship	-
Skill Development	Studying knowledge-based agents in artificial intelligence develops critical thinking, problem-solving, and programming skills.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-

Unit IV	**Classification & Regression**
Local	-
Regional	-
National	-
Global	Classification and regression techniques have widespread applications in various domains, including finance, marketing, environmental analysis, and social sciences.
Employability	Acquiring knowledge and skills in classification and regression enhances employability prospects, especially in fields such as data analysis, machine learning, and artificial intelligence.
Entrepreneurship	-
Skill Development	Studying classification and regression techniques develops critical thinking, data analysis, and problem-solving skills.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
SDG	SDG 8, 9
NEP 2020	Integration of Emerging Technologies and Skill Development
POE/4th IR	The course on AI and Machine Learning aligns with the Fourth Industrial Revolution (IR 4.0) in the following ways: Data-Driven Decision-Making, Automation and Optimization and Predictive Analytics.

Benchmarking Universities (atleast 4-5 standard university contents must be reffered):

Name of the University	**Program referred for Syllabus contents**	**Subject referred**
Bennett University	B.Tech(CSE)	AI and Machine Learning
Savitribai Phule Pune University	MCA	AI and Machine Learning
JNTU Hyderabad	B.Tech(CSE)	Introduction to AI and Machine Learning

PROBLEM SOLVING LAB

Defined Course Outcomes

COs	
CO 1	Demonstrate the problem-solving skills using variables, basic data types, python data structures, operators, conditional statements, looping structures, functions etc.
CO 2	Solve real world problems involving the concepts based on OOPS
CO 3	Solve real world problems involving the concepts based on regression
CO 4	Work on real world data sets and develop Classifier models in python

Ex. No	Experiment Title	Mapped CO/COs
1	Write a Python function that takes a list as input and returns a new list containing only the unique elements from the original list. The order of the elements should be preserved.	CO1
2	Program to find greatest common divisor (GCD) or highest common factor (HCF) of given two numbers.	CO1
3	Write a Python function that takes two lists as input and returns a new list containing the elements that are common to both input lists, without any duplicates. The order of the elements should be preserved.	CO1
4	Write a Python program that takes a list of tuples as input, where each tuple contains a name and an age. The program should sort the list of tuples based on the age of the individuals in ascending order.	CO1
5	Write a Python function that takes two lists as input and returns a new list that contains the elements from both input lists, in alternating order. For example, if the input lists are [1, 2, 3] and [4, 5, 6], the output should be [1, 4, 2, 5, 3, 6].	CO1
6	Write a Python program that takes a list as input and returns a new list that contains only the elements at even indices (0, 2, 4, etc.) of the input list.	CO1
7	Write a Python function that takes a list and an element as input, and returns the number of occurrences of the element in the list	CO1
8	Write a Python program that takes an integer n as input and	CO1

	prints the following pattern 1 2 2 3 3 3 4 4 4 4 ... n n n ..... n	
9	Create a Python class called BankAccount that represents a bank account. The class should have attributes such as account number, account holder name, and balance. Implement methods to deposit, withdraw, and display the account details	CO2
10	Create a Python class called Library that represents a library. The class should have attributes such as book catalog, member list, and loan history. Implement methods to add books, add members, lend books, return books, and display library statistics.	CO2
11	Create a Python class called Shape that represents a geometric shape. Implement subclasses such as Rectangle, Circle, and Triangle that inherit from the Shape class. Each subclass should have specific methods to calculate area and perimeter, and display shape details.	CO2
12	Create a Python class called Employee that represents an employee. The class should have attributes such as employee ID, name, and salary. Implement methods to calculate the bonus based on performance, display employee details, and compare employees based on salary.	CO2
13	Create a Python class called Product that represents a product in an online store. The class should have attributes such as product ID, name, price, and quantity. Implement methods to calculate the total cost of a product, update the quantity when a product is purchased, and display product details	CO2
14	Create a Python class called Car that represents a car. The class should have attributes such as make, model, and fuel type. Implement methods to calculate the rental cost based on the number of days, display car details, and compare cars based on fuel efficiency.	CO2
15	Create a Python class called Student that represents a student. The class should have attributes such as student ID, name, and grades. Implement methods to calculate the average grade, display student details, and determine if the student is eligible for graduation.	CO2

16	Build a Python program that performs linear regression to predict house prices based on features such as square footage, number of bedrooms, and location. Use a dataset containing historical house prices and corresponding features to train the regression model, and then make predictions on new data.	CO3
17	Create a Python program that uses regression algorithms (e.g., linear regression, support vector regression, or random forest regression) to predict stock prices. Train the model using historical stock prices and related factors (e.g., trading volume, news sentiment), and evaluate its performance on unseen data	CO3
18	Develop a Python program that utilizes regression techniques to forecast energy consumption. Use historical energy usage data along with weather conditions, time of year, and other relevant features to train a regression model. Evaluate the model's accuracy in predicting future energy consumption.	CO3
19	Build a Python program that performs text classification to classify emails as spam or non-spam. Train a classification model (e.g., logistic regression, Naive Bayes, or support vector machines) using a labeled dataset of emails. Evaluate the model's performance by calculating metrics such as accuracy, precision, recall, and F1 score	CO4
20	Create a Python program that uses classification algorithms (e.g., convolutional neural networks, random forests, or k-nearest neighbors) to perform image recognition. Train the model using labeled images of different objects or classes. Test the model's accuracy on new images and display the predicted class for each image	CO4
21	Develop a Python program that applies classification techniques to detect credit card fraud. Train a classification model (e.g., logistic regression, decision trees, or random forests) using a dataset of credit card transactions, where each transaction is labeled as fraudulent or legitimate. Evaluate the model's performance by calculating metrics such as accuracy, precision, recall, and area under the ROC curve	CO4

DATA STRUCTURES AND ALGORITHMS LAB

Defined Course Outcomes

COs	
CO 1	Equip the students with knowledge of algorithms and analysis of space and time complexity of the algorithms
CO 2	Demonstrate the use of stack ,queues and linked list
CO 3	Equip the students with tree and graph data structures and their practical applications
CO 4	Implementing and analysing searching and sorting algorithms

Ex No	Experiment Title	Mapped CO/COs
1	To design, implement and analyze the complexity of Linear search algorithm	CO4
2	To design, implement and analyze the complexity of Binary search algorithm	CO4
3	Implement and compare the time complexity of bubble sort, insertion sort and selection sort. Calculate their running times for best, worst & best cases. Draw the three cases in a single graph to justify its observed time complexities.	CO4
4	Implement and analyse the working of Recursive Algorithms	CO1
5	Implement Quick sort algorithm and calculate its running times for best, worst & best cases. Draw the three cases in a single graph to justify its observed time complexities.	CO4
6	Implement the linear data structure : Stack by performing Push and Pop operation	CO2
7	Implement Postfix and Prefix Expression using Stack	CO2
8	Implement reverse of a String using Stack	
9	Implement the linear data structure : Queue by performing Insertion and Deletion operation	CO2
10	Implement Circular Queue by performing Insertion and Deletion operation	
11	Implement the dynamic data structure : single linked list also analyse their time complexities in three cases: a. Inserting a new node at the beginning	CO2

	b. Inserting a new node at the end c. Deleting a node from the beginning	
12	Consider a linked list L reverse the linked list	CO2
13	Implement the dynamic data structure : doubly linked list also analyse their time complexities in three cases: a. Inserting a new node at the beginning b. Inserting a new node in the middle c. Deleting a node from the end	CO2
14	Implement the dynamic data structure : circular linked list also analyse their time complexities in three cases: a. Inserting a new node at the beginning b. Inserting a new node in the middle c. Deleting a node from the end	CO2
15	Implement and analyse Stack implementation using Linked list	CO2
16	Implement and analyse Queue implementation using Linked list	
17	Implement and analyse the tree traversal algorithms 1. Inorder 2. Preorder 3. Post order	CO3
18	Implement and analyse the following operations of Binary Search tree a. Creating and inserting a new node b. Searching a node c. Deleting an existing node from BST	CO3
19	Implement AVL tree with insertion, deletion and searching operation	CO3
20	Implement the graph traversal techniques: Depth First search and Breadth First search algorithms	CO3
21	To understand and implement the minimum spanning tree in Graphs using Kruskal Algorithm	CO3
22	To understand and implement the minimum spanning tree in Graphs using Prims Algorithm	CO3
23	Implement Merge sort algorithm and calculate its running times for best, worst & best cases. Draw the three cases in a single graph to justify its observed time complexities.	CO4
24	Implement Heap sort algorithm and calculate its running times for best, worst & best cases. Draw the three cases in a single graph to justify its observed time complexities.	CO4

25	Implement a priority queue using a heap and calculate its running times for best, worst & best cases. Draw the three cases in a single graph to justify its observed time complexities.	CO4
	Mini Project 1: Create a student management system that stores and manages student records using various data structures. The system should allow users to perform operations such as adding new students, searching for students, deleting students, and displaying all student records.	
	Mini Project 2: Implement a maze solver using data structures like stacks or queues. The program should take an input maze, find a path from the starting point to the goal, and output the solution. You can use depth-first search (DFS) or breadth-first search (BFS) algorithms to solve the maze.	
	Mini Project 3: Implement a social network analysis tool using data structures like graphs. The tool should be able to read a network of users and their connections, and perform operations like finding the shortest path between two users, identifying influential users, or recommending friends.	

AI & Machine Learning lab

Defined Course Outcomes

COs	
CO 1	Evaluate Artificial Intelligence (AI) methods and describe their foundations.
CO 2	Analyze and illustrate how search algorithms play vital role in problem solving, inference, perception, knowledge representation and learning.
CO 3	Demonstrate knowledge of reasoning and knowledge representation for solving real world problems.
CO 4	Recognize the characteristics of machine learning that makes it useful to real-world Problems.
CO 5	Apply the different supervised learning methods of support vector machine and tree based models.
CO 6	Use different linear methods for regression and classification with their optimization through different regularization techniques.

List of Programs

Ex No	Experiment Title	Mapped CO/COs
1	Implement uninformed search algorithm Breadth-First Search (BFS) on a given problem.	CO1
2	Implement uninformed search Depth-First Search (DFS) on a given problem.	CO1
3	Implement uninformed search algorithm Uniform Cost Search (UCS) on a given problem.	CO1
4	Implement greedy search algorithm Best-First Search on a given problem	CO1
5	Implement greedy search algorithm A* Search on a given problem	CO2
6	Implement a backtracking algorithm to solve a constraint satisfaction problem (CSP).	CO3

7	Apply the backtracking algorithm to a real-world problem, such as the N-Queens problem or the Sudoku puzzle.	CO3
8	Design and implement a knowledge-based agent using a knowledge representation technique like propositional logic or first-order logic.	CO3
9	Implement logistic regression for binary classification tasks and evaluate the performance of the classifier using appropriate evaluation metrics and visualize the results.	CO4
10	Implement decision tree, for binary classification tasks and evaluate the performance of the classifier using appropriate evaluation metrics and visualize the results.	CO4
11	Implement support vector machine for classification tasks and evaluate the performance of the classifier using appropriate evaluation metrics and visualize the results.	CO4
12	Implement random forest for classification tasks and evaluate the performance of the classifier using appropriate evaluation metrics and visualize the results.	CO4
13	Implement linear regression and evaluate the performance of the classifier using appropriate evaluation metrics and visualize the results.	CO5
14	Implement polynomial regression and evaluate the performance of the classifier using appropriate evaluation metrics and visualize the results.	CO5
15	Implement k-NN and evaluate the performance of the classifier using appropriate evaluation metrics and visualize the results.	CO5
16	Implement k-Means and evaluate the performance of the classifier using appropriate evaluation metrics and visualize the results.	CO6
17	Apply dimensionality reduction techniques, such as Principal Component Analysis (PCA), on a dataset with high-dimensional features.	CO6

Life Skills for Professionals – I

Course Code	Course Title	L	T	P	S	Credit
	Life Skills for Professionals - I	2	1			3
Pre-requisites/Exposure						

(L – Lecture T – Tutorial P – Practical S – Studio C – Credits)

COURSE OBJECTIVES: (in bullet points)

- It will define Quantitative aptitude is the skill and ability to solve complex mathematical quantities or equations by analytical and logical means.
- To develop and build upon their abilities in listening reading and speaking skills.
- Improve numerical literacy, data analysis, statistical reasoning, and proficiency in using quantitative tools and techniques to make informed decisions and solve quantitative problems.
- Cultivate leadership qualities such as decision – making, problem – solving, team management skills.
- Recognize the interrelationship between theory and practice apply such knowledge for development.
- Enhance verbal and written communication skills, including clarity, articulation, active listening, and effective presentation skills.

PROGRAMME OUTCOMES (POs)

PO 1 Learner should be able to effectively express their ideas thoughts and opinions both orally and in writing they should demonstrate improved clarity coherence and persuasiveness.

PO 2 Learner should develop active listening skills allowing them to attentively engage in conversation understand the other perspective and respond appropriately

PO 3 The program extends learner become more confident and skilled presentator enabling them to deliver engaging and impactful presentations while in various settings learners should demonstrate improved interpersonal communication.

PO 4 The program enhances learner's analytical and logical reasoning abilities enabling them to solve complex programming problems if efficiently and make informed decisions based on quantitative data.

PO 5 The Learner should demonstrate improved critical thinking skills including analysing inform information evaluating arguments and making sound judgements

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

Develop effective communication skills and enable learners to effectively convey ideas and thoughts and information through verbal and written communication.
It will inculcate interpersonal communication competence which will help learners to develop ability to establish and maintain positive interpersonal relationships including active listening empathy and conflict resolution skills.
It also fosters critical thinking and problem-solving abilities it will develop critical thinking skills including analyzing information evaluating arguments and applying logical reasoning to solve complex problems.

PROGRAM SPECIFIC OUTCOMES (PSOs)

- It will improve learner's quantitative skills including numerical literacy data analysis and ability to interpret in use quantitative information effectively.
- It will have participants to develop confidence in their communication and liabilities enabling to express themselves and overcome communication barriers. The learners will be able to express their ideas thoughts and information clearly effectively both verbally and in writing the learners will demonstrate improved active listening skills allowing them to understand and interrupt information accurately.
- The learner will develop awareness of the non-verbal cues and use them effectively in communication including body language facial expressions and gestures.

- The learner will develop analytical and logical reasoning skills allowing them to identify problems analyze data and develop their effective solutions participants will demonstrate improved numerical literacy and ability to interpret and analyze quantitative information.

LIFE SKILLS FOR PROFESSIONALS - I

1. Department:	**Department of Computer Science and Engineering**		
2. Course Name: Career Development Program-I	**3. Course Code**	**4. L-T-P**	**5. Credits**
		2-0-1	3
6. Type of Course:	Life Skills for Professionals - I		
7. Pre-requisite(s), if any:			
8. Frequency of offering (check one): Odd / Even			

9. Brief Syllabus:

The basics of Quantitative Aptitude are starting from Simplification questions then Number system, Percentage, and Average. If you are just starting learning Quantitative aptitude, you must start with simplification and the number system. The main quantitative Aptitude topics are Number System, Simplification, Percentage, simple interest and compound interest, Profit and loss, Ratio and Proportion, Time and Work, Time, Speed, and Distance, Average, Probability, Permutation, and Combination, Equations and Equalities. Data interpretation questions can be solved by using all quantitative aptitude topics concepts.

Life skills for professionals' course are designed for learnering learner to enhance and develop interpersonal skills that characterize a person's relationships with other professionals. This program will teach skills which will prepare them for a successful career in their industry. The main topics will

include verbal communicational skills, non- verbal communication skills, Active listening skills, written communication skills and presentation skills.

10.Total lecture, Tutorial and Practical Hours for this course:

Lectures: 30	**Practice**	
	Doubt session:7	**PC Lab: 8**

11. Course Outcomes (COs)

Usefulness of this course after its completion i.e. the learner should build self confidence in their communication abilities the learner should exhibit improved communication skills assesses in communication skills, logical reasoning, quantitative skills along with job-specific domain skills.

COs	Learner will develop self confidence in their communication abilities and enabling them to express themselves assertively.
CO 1	Learner will develop the ability for advanced critical thinking and the ability to formulate logical arguments.
CO 2	The learner to apply different value systems and moral dimensions while taking decisions.
CO 3	Apply mathematical techniques to quantitative theory.
CO 4	Soft skills include attributes and personality traits that help learner to interact with others and succeed.
CO 5	Learner will cultivate self-confidence, problem solving and critical thinking abilities

12. UNIT WISE DETAILS

Unit Number: 1	**Title: Introduction to number system**	**No. of hours: 8**

Content Summary:

Number system, Divisibility, Unit digit, Last two-digit, Remainder, Number of zero, Factor, LCM & HCF, Simplification.

Unit Number: 2	**Title: Ratio & its application**	**No. of hours: 8**
Content Summary: Ratio, Mixture, Average, Partnership.		
Unit Number: 3	**Title: communication**	**No. of hours: 7**
Content Summary: Introduction to Communication, Types of communication, Verbal & Nonverbal Communication, Barriers to Communication, Body language, Listening Skills.		
Unit Number: 4	**Title: Personality development**	**No. of hours: 7**
Content Summary: Paragraph writing, Professional Speaking (Elocutions, Debate, describing incidents and developing positive nonverbal communication. Articulation and pronunciation. Communicating with confidence. Using appropriate tone pitch and volume.		
Contents beyond Syllabus: https://www.youtube.com/watch?v=0pNGYM0ltlw https://www.youtube.com/watch?v=0gUgm4zB2F4		
Reference Books: Qunatitative Aptitude by R.S Agarawal Quicker math by M.Tyra Communication skills by G.H. Hook		

Semester II

ADVANCED COMPUTER ORGANIZATION & ARCHITECTURE

Department:	**Department of Computer Science and Engineering**		
Course Name: Advanced Computer Organization & Architecture	**Course Code**	**L-T-P**	**Credits**
		3-1-0	4
Type of Course:	Programme Core		
Pre-requisite(s), if any: Computer Organization and Architecture, Microprocessor			

Brief Syllabus:
Advanced Computer Organization & Architecture (ACOA) covers advanced topics in computer architecture and organization focusing on multicore, graphics-processor unit (GPU), and heterogeneous SOC multiprocessor architectures and their implementation issues (architect's perspective). The objective of the course is to provide in-depth coverage of current and emerging trends in computer organization and architecture focusing on performance and the hardware/software interface. The course emphasis is on analysing fundamental issues in architecture design and their impact on application performance.

UNIT WISE DETAILS

Unit Number: 1	**Title: Theory of Parallelism**	**No. of hours: 10**

Content Summary:
Parallel Computer Models, The State of Computing, Multiprocessors and Multicomputer, Multivector and SIMD Computers ,PRAM and VLSI Models, Program and Network Properties ,Conditions of Parallelism, Program Partitioning and Scheduling, Program Flow Mechanisms, System Interconnect Architectures, Principles of Scalable Performance, Parallel Processing Applications, Speedup Performance Laws, Scalability Analysis and Approaches.

Unit Number: 2	**Title: System Interconnect Architectures, Advanced processors and Pipelining**	**No. of hours: 10**

Content Summary:
Network properties and routing, Static interconnection Networks, Dynamic interconnection Networks, Multiprocessor system Interconnects, Hierarchical bus systems, Crossbar switch and multiport memory, Multistage and combining network.
Advanced processor technology, Instruction-set Architectures, CISC Scalar Processors, RISC Scalar Processors, Superscalar Processors.
Linear pipeline processor, nonlinear pipeline processor, Instruction pipeline Design, Mechanisms for instruction pipelining, Instruction Pipeline Principles, Pipeline Hazards Analysis, Control Hazards & Branch Prediction, MIPS Pipeline for Multi-Cycle Operations, Dynamic instruction scheduling, Branch Handling techniques, branch prediction.

Unit Number: 3	**Title: Memory Hierarchy Design, Multiprocessor architectures**	**No. of hours: 10**

Content Summary:
Cache basics & cache performance, reducing miss rate and miss penalty, multilevel cache hierarchies, main memory organizations, design of memory hierarchies.
Symmetric shared memory architectures, distributed shared memory architectures, models of memory consistency, cache coherence protocols (MSI, MESI, and MOESI), scalable cache coherence, overview of directory based approaches, design challenges of directory protocols, memory based directory protocols, cache based directory protocols, protocol design trade-offs, synchronization.

Unit Number: 4	**Title: Parallel Models and Languages**	**No. of hours: 10**

Content Summary:
Parallel Programming Models(Shared-Variable, Message passing, Data-Parallel, Object-Oriented), Parallel languages & Compilers (language features for parallelism, Parallel Program Development and Environments, Synchronization and Multiprocessing Modes, Parallel language constructs, optimizing compilers for parallelism), Code optimization & partitioning (Scalar optimization, Local & Global optimization, Vectorization, code generation & scheduling, Trace scheduling compilation).

***Self-Learning Components:**

1. GPU Programming with CUDA

References:

1. https://www.udemy.com/course/cuda-gpu-programming-beginner-to-advanced
2. https://www.coursera.org/learn/introduction-to-parallel-programming-with-cuda

Please Note:
At least 5-10 % syllabus will be asked in end term exams from self-learning components

Reference Books:

1. Computer Architecture and Organization, J.P. Hayes, McGraw Hills.
2. Memory System and Pipelined Processors, Harvey G. Cragon, Narosa Publication.
3. Parallel Computer, V.Rajaranam & C.S.R. Murthy, PHI.
4. Scalable Parallel Computers Architecture, Kai Hwang and Zu, MGH.
5. Computer Organization & Architecture, Stalling W, PHI.
6. Computer Architecture, Pipelined and Parallel Processor Design, M.J Flynn, Narosa Publishing.

Online References:

1. https://www.cl.cam.ac.uk/teaching/2021/R265/
2. https://www.ee.iitb.ac.in/~viren/Courses/2015/EE748.htm

Course Outcomes (CO)

COs	Statements
CO1	**Describe** the various architectural concepts that may be applied to optimize and enhance the classical Von Neumann architecture into high performance computing hardware systems.
CO2	**Describe** the design issues relating to the architectural options.
CO3	**Determine** the challenges faced in the implementation of these high-performance systems
CO4	**Understand** pipelining, instruction set architectures, memory addressing.
CO5	**Identify** the various techniques to enhance a processors ability to exploit Instruction-level parallelism (ILP), and its challenges.
CO6	**Understand** the various models to achieve memory consistency.

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels(C) 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C1	A1	P1
C02	C1	A4	P1
CO3	C2	A2	P3
CO4	C2	A1	P1
CO5	C3	A2	P2
CO6	C2	A1	P1

CO-PO Mapping

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	2				2						
CO2	2	2				2						
CO3	2	2		3		2						
CO4		3				3						
CO5		3										
CO6						2						

1=weakly mapped
2= moderately mapped
3=strongly mapped

CO-PSO Mapping

PO	PSO1	PSO2	PSO3	PSO4
CO1	2	2		3
CO2	2	2		2
CO3			3	
CO4			2	
CO5				
CO6				

Relevance of the Syllabus to various indicators

Unit I	Theory of Parallelism
Local	-
Regional	-
National	Contributes to national digital literacy and computer architecture and organization strategies.
Global	Aligns with global trends in computer organization and modern computers
Employability	Develops skills in computer architecture based services.
Entrepreneurship	-
Skill Development	Develops basic knowledge and skills in advanced computer organization and architecture.
Professional Ethics	Develop basic professional ethics in computer architecture
Gender	-
Human Values	-
Environment & Sustainability	-
Unit II	System Interconnect Architectures, Advanced processors and Pipelining
Local	-
Regional	-
National	Provide essential knowledge and skills related to computer architecture and advanced processors. Crucial for the development and advancement of the country's information technology sector.
Global	Form the foundation of modern computer systems and their performance optimization.
Employability	Develops skills for computer science and information technology industries.
Entrepreneurship	-
Skill Development	Develops critical technical skills for individuals aspiring to excel in computer engineering and related fields.
Professional Ethics	Indirectly promotes ethical practices by equipping students with knowledge and skills to design efficient and reliable computer systems.
Gender	-
Human Values	Contribute to the development of technology solutions that align with human values, such as accessibility, inclusivity, and user-centric design.
Environment & Sustainability	Indirectly contribute to energy-efficient computing practices

Unit III	Memory Hierarchy Design, Multiprocessor architectures
Local	-
Regional	-
National	Provides essential knowledge and skills related to the development and optimization of computer systems.
Global	Aligns to computer system design and optimization worldwide.
Employability	Develops highly valuable skills in in the computer science and information technology industries.
Entrepreneurship	-
Skill Development	Develops critical technical skills in computer engineering
Professional Ethics	Indirectly promotes ethical practices by equipping students with the knowledge and skills to design efficient and reliable computer systems.
Gender	-
Human Values	Indirectly supports human values by fostering the development of computer systems.
Environment & Sustainability	Contribute to energy-efficient computing practices
Unit IV	Parallel Models and Languages
Local	-
Regional	-
National	Contributes in development and optimization of computer systems at the national level.
Global	The concepts are applied globally in various domains and industries.
Employability	Highly valued in the computer science and information technology industries.
Entrepreneurship	-
Skill Development	Develops critical technical skills in software development, parallel computing, and optimization.
Professional Ethics	Indirectly by equipping with the knowledge and skills to develop efficient and scalable software solutions
Gender	-
Human Values	Indirectly by fostering the development of software solutions that are efficient, scalable, and responsive to user needs.
Environment & Sustainability	-
SDG	SDG 4, SDG 8, SDG 9
NEP 2020	Promoting universal access to education, holistic development, multidisciplinary approach, skill

	development, critical thinking, creativity, ICT integration, research and development, global competencies, and professional ethics.
POE/4th IR	Aligns with the concepts of parallel computing, advanced processors, and memory architectures.

Benchmarking Universities (atleast 4-5 standard university contents must be reffered):

Name of the University	**Program referred for Syllabus contents**	**Subject which is referred**
Visvesvaraya Technological University, Belagavi	B.Tech CSE	Advanced Computer Architectures
IIT, Guwahati	B.Tech	Advanced Computer Architectures
JC Bose University, Faridabad	B.Tech CSE	Advanced Computer Architectures
University of Warwick	Under graduate Level 2	Advanced Computer Architecture

K.R. MANGALAM UNIVERSITY

ADVANCE WEB TECHNOLOGIES

Department:	**Department of Computer Science and Engineering**		
Course Name: Advance Web Technologies	**Course Code**	**L-T-P**	**Credits**
	ENMM704	3-1-0	4
Type of Course:	Programme Core / Programme Elective /Open Elective		
Pre-requisite(s), if any:			

Brief Syllabus:
This course is an introduction to Web site development and the technologies behind it. Students will learn how to design and develop Web pages using current technologies and tools. Topics covered will include the World Wide Web, HTML, Cascading Style Sheets (CSS) and XML. The focus of this course is on dynamic HTML, a collection of web technologies such as HTML and scripting languages used together to create interactive and animated Web pages. Students will learn to program client-side scripts using JavaScript and the Document Object Model to transform static Web pages created with HTML 5 and CSS 3 into dynamic Web pages.

UNIT WISE DETAILS

Unit Number: 1	**Title:**	**No. of hours: 8**

Content Summary:
Introduction, Web Programming, HTML (Hypertext Markup Language), Document Object Model
(DOM), More HTML Elements, Forms, CSS (Cascading Style Sheets), Responsive Design, Bootstrap,
SASS (Syntactically Awesome Style Sheets), Introduction to Git, GitHub, Commits, Merge Conflicts,
Branching, More GitHub Features

Unit Number: 2	**Title:**	**No. of hours: 12**

Content Summary:
Decorators and Lambda Function in Python, Introduction to Web Applications, HTTP, Django, Routes,
Templates: Conditionals and Styling, Tasks, Forms: Django Forms, Sessions
Introduction to SQL: Databases, Column Types; Tables ; SELECT:

Working with SQL in the Terminal, Functions, UPDATE, DELETE, Other Clauses, Joining Tables: JOIN Query, Indexing, SQL Vulnerabilities; Django Models, Migrations, Shell: Starting our application, Django Admin, Many-to-Many Relationships, Users		
Unit Number: 3	**Title:**	**No. of hours: 12**
Content Summary: Introduction to JavaScript, Events, Variables, querySelector, DOM Manipulation:JavaScript Console, Arrow Functions, TODO List; Intervals, Local Storage, APIs: JavaScript Objects, Currency Exchange.Introduction to User Interfaces,Single Page Applications, Scroll:Infinite Scroll; Animation, React: Addition		
Unit Number: 4	**Title:**	**No. of hours: 8**
Content Summary: Introduction to Testing, Assert: Test-Driven Development, Unit Testing, Django Testing: Client Testing, Selenium, CI/CD, GitHub Actions, Docker Scalability, Scaling, Load Balancing, Autoscaling: Server Failure, Scaling Databases: Database Replication, Caching, Security: Git and GitHub, HTML, HTTPS: Secret-Key Cryptography,Public-Key Cryptography, Databases : APIs, Environment Variables; JavaScript: Cross-Site Request Forgery		
***Self-Learning Components:** mention 4-5 topics for students in bullet points Please Note: 1)Students are supposed to learn the components on self-basis 2)Mention open-source tools/ new concepts/technologies that students will be required to learn and present through presentations in class 3) At least 5-10 % syllabus will be asked in end term exams from self-learning components		
Reference Books: 1.Web Technologies, Uttam K. Roy, Oxford University Press 2. HTML Black Book, Stephen Holzner, Wiley Dreamtech. 3. Web Technology, Rajkamal, Tata McGraw-Hill. 4. Web Technologies: A Computer Science Perspective, Jeffrey C. Jackson, Pearson. 5. XML: How to Program, Deitel&Deitel Nieto		

Define Course Outcomes (CO)

COs	Statements
CO1	Create a well-designed and well-formed, professional Web site utilizing the most current standards and practice
CO2	Demonstrate knowledge in web technologies including HTML, XHTML, CSS, image editing software, web authoring software, and client-side scripting.
CO3	To use GIT and GIT HUB for project management
CO4	To apply Django web framework to create websites
CO5	To create interactive and responsive website using Javascript

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels© 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C2		P1
C02	C3		P2
CO3	C3		P3
CO4	C1		-
CO5	C1		P1

***Please Note:**
Map only 1 or 2 Levels in each category. If a higher level is given, no need to mention lower level

Program outcomes (POs)
Engineering Graduates will be able to:

PO1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO2. **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
PO6. **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO7. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

CO-PO Mapping

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1					1				1		
CO2		1										
CO3			2				3				1	

CO4				2				2				
CO5					3				1			
												1

Please Note:

- Refer to POs while mapping each CO.
- Mark " – " if not applicable
- If attainment of a CO is strongly mapped with a PO , Mark 3
- If attainment of a CO is moderately mapped with a PO , Mark 2
- If attainment of a CO is weakly mapped with a PO , Mark 1

Justification for mapping must be relevant

1=weakly mapped
2= moderately mapped
3=strongly mapped

CO-PSO Mapping

PO	PO1	PO2	PO3	PSO4
CO1	1		3	1
CO2		1		
CO3	1		2	
CO4				1
CO5	1		1	

Relevance of the Syllabus to various indicators

Unit I	Introduction
Local	Addresses local understanding of the Internet and its impact on society
Regional	Addresses regional internet connectivity and network infrastructure requirements
National	Contributes to national digital literacy and internet connectivity strategies
Global	Aligns with global trends in internet technologies and network protocols
Employability	Develops skills in using internet-based services and understanding network protocols
Entrepreneurship	-
Skill Development	Develops basic knowledge and skills in internet technologies and network protocols

Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit II	
Local	Addresses local understanding of the Internet and its impact on society
Regional	-
National	Contributes to national digital literacy and internet connectivity strategies
Global	Aligns with global trends in internet technologies and network protocols
Employability	Develops skills in using internet-based services and understanding network protocols
Entrepreneurship	-
Skill Development	Develops basic knowledge and skills in internet technologies and network protocols
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	
Local	Addresses local network security needs and practices
Regional	-
National	Contributes to national network security strategies and protocols
Global	Aligns with global trends in network security techniques and protocols
Employability	Develops skills in network programming and network security techniques
Entrepreneurship	-
Skill Development	Develops knowledge and skills in client-server programming and network security
Professional Ethics	-
Gender	-
Human Values	-
Environment &	-

Sustainability	
Unit IV	
Local	Addresses local understanding and implementation of internet-based services
Regional	-
National	Contributes to national digital communication strategies and multimedia applications
Global	Aligns with global trends in internet telephony, multimedia applications, and SEO
Employability	Develops skills in internet telephony, multimedia applications, and SEO
Entrepreneurship	-
Skill Development	Develops knowledge and skills in internet telephony, multimedia applications, and SEO
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
SDG	SDG 4
NEP 2020	-
POE/4th IR	Aligns with the concepts of internet telephony, multimedia applications, and SEO

INTRODUCTION OF NEURAL NETWORK AND DEEP LEARNING

<table>
<tr><td>Department:</td><td colspan="3">Department of Computer Science and Engineering</td></tr>
<tr><td rowspan="2">Course Name: Introduction of Neural Network and Deep Learning</td><td>Course Code</td><td>L-T-P</td><td>Credits</td></tr>
<tr><td>ENMM706</td><td>4-0-0</td><td>4</td></tr>
<tr><td>Type of Course:</td><td colspan="3">Major</td></tr>
<tr><td colspan="4">Pre-requisite(s), if any:</td></tr>
<tr><td colspan="4">Brief Syllabus:

The course begins with key concepts of neural networks, feed-forward neural network, and back propagation. The student gets an opportunity to learn the programming languages (Tensor Flow) to train the deep learning models. The student learns the concepts behind deep learning algorithms and its use cases.</td></tr>
<tr><td colspan="4">UNIT WISE DETAILS</td></tr>
<tr><td>Unit Number: 1</td><td colspan="2">Title: Basics of Neural Network</td><td>No. of hours: 10</td></tr>
<tr><td colspan="4">Content Summary: Basic ideas behind the Neural Network, Social Engineering about the data, Importance, and applications of Neural network Neural network concepts, Information flow in a neural network, understanding the basic structure of biological Neural Networks and ANN. Activation functions and their uses.(Sigmoid, Relu, Tanh etc.)</td></tr>
<tr><td>Unit Number: 2</td><td colspan="2">Title: Feedforward neural network</td><td>No. of hours: 10</td></tr>
<tr><td colspan="4">Content Summary: Linear Models, Training a Neural network, how to determine hidden layers, recurrent neural, multi-layer neural network, Risk minimization, regularization, model selection, and practical optimization.</td></tr>
</table>

Unit Number: 3	Title: Deep Learning	No. of hours: 10
Content Summary: Deep Feed Forward network, bias-variance dilemma, Overfitting, dropouts, Gradient decent algorithm, Convolutional Neural Network, Recurrent Neural Network, Deep Belief Network. Challenges in designing the best Neural Network ---**Self-learning**		
Unit Number: 4	**Title: Probabilistic Neural Network and Deep Learning Research**	**No. of hours: 10**
Content Summary: Hopfield Net, Boltzmann machine, RBMs, Need of Encoders and Auto encoders, Object recognition, computer vision, natural language processing. Research areas in Probabilistic Neural Networks		
Reference Books: 1. Yegnanarayana, B., Artificial Neural Networks PHI Learning Pvt. Ltd, 2009. 2. Golub, G., H. and Van Loan, C, F, Matrix Computations, JHU Press,2013. 3. Satish Kumar, Neural Networks: A Classroom Approach, Tata McGraw-Hill Education, 2004. **Text Books** 1. Goodfellow, I., Bengio,Y., and Courville, A., Deep Learning, MIT Press, 2016.		

Define Course Outcomes (CO)

COs	Statements
CO1	**Understand** neural network and Biological Neural Network
CO2	**Express** proficiency in the handling of feedforward Neural Network
CO3	**Determine** methods to create and manipulate Deep Neural Network
CO4	**Identify** commonly used operations involving in designing Deep Neural Network

CO5	**Articulate** Neural Network, such as Backpropagation, Drop out, overfitting and their use in programs.

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels© 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C2		P1
C02	C3		P2
CO3	C3		P2
CO4	C1		-
CO5	C1		P4

CO-PO Mapping

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	–	–	–	–	1	1	1	–	–	–	1
CO2	3	3	3	–	3	–	–	–	–	2	–	–
CO3	3	3	3	2	3	–	–	–	–	–	–	2
CO4	2	2	2	–	2	–	–	–	–	–	–	–
CO5	2	2	–	2	2	–	–	–	–	2	–	–

CO-PSO Mapping

PO	PSO1	PSO2	PSO3	PSO4
CO1	2	–	–	–
CO2	3	3	3	–
CO3	3	3	–	–
CO4	2	2	–	2
CO5	2	2	2	2

Relevance of the Syllabus to various indicators

Unit I	**The Neural Network**
Local	Information flow in a neural network
Regional	understanding the basic structure of biological Neural Networks and ANN.
National	Training a network
Global	loss functions, activation functions
Employability	Develops skills in using internet-based services and understanding network protocols
Entrepreneurship	-
Skill Development	Develops basic knowledge and skills in Neural networks.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit II	**Feedforward neural network**
Local	Understanding of how to train the neural networks
Regional	-
National	-
Global	Aligns with global trends in Neural networks technologies
Employability	Develops skills in using Neural network
Entrepreneurship	-
Skill Development	Develops basic knowledge and skills in Deep learning.

Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	**Deep Learning**
Local	Understanding different algorithms used in deep learning
Regional	-
National	Contributes to deep learning technology strategies
Global	Aligns with global trends in Deep neural network
Employability	Develops skills in programming and other techniques
Entrepreneurship	-
Skill Development	Develops knowledge and skills in Deep learning
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	**Probabilistic Neural Network and Deep Learning research**
Local	
Regional	-
National	Contributes to national business strategies.
Global	Aligns with global trends in deep learning research and

	applications
Employability	Develops skills in predictive modeling
Entrepreneurship	-
Skill Development	Develops knowledge and skills in data analytics as well as in predictions
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
SDG	SDG 4
NEP 2020	-
POE/4th IR	Aligns with the concepts of Neural network and Deep Learning applications

K.R. MANGALAM UNIVERSITY

ADVANCE WEB TECHNOLOGIES LAB

Department:	**Department of Computer Science and Engineering**		
Course Name: **Web Technologies lab**	**Course Code**	**L-T-P**	**Credits**
	ENMM754	0-0-2	1
Type of Course:	Major Course		

Defined Course Outcomes

COs	Course Outcomes (COs)
CO 1	To design simple webpages using HTML and CSS.
CO 2	To use GIT and GIT HUB for project management.
CO 3	To apply Django web framework to create websites.
CO 4	To create interactive and responsive website using JavaScript.
CO5	To test and deploy application web applications.

Ex. No	Experiment	Mapped COs
1	Implement User Registration and Login form using JavaScript.	CO1
2	Designing a Responsive Navigation Bar with CSS	CO1
3	CSS: Inline Style , Internal Style ,and External Style Sheets	CO1
4	Developing a Real-time Chat Application with Flask and WebSocket	CO1
5	Building an Interactive To-Do List Using JavaScript and Flask.	CO1
6	Validating User Inputs in HTML Forms.	CO1
7	Dynamically Modifying HTML Elements and Content.	CO2
8	Designing an e-commerce website with product listings, shopping cart, and secure payments.	CO2
9	Integrate charting libraries (e.g., Chart.js) to visualize data from your Flask app.	CO3
10	Build a simple video streaming application where users can upload and view videos.	C03
11	Design a responsive navigation bar using HTML and CSS.	CO2

12	Implement a JavaScript function to change the background color of a button on hover.	Co3
13	Build a form with validation using HTML5 attributes.	CO1
14	Create a JavaScript countdown timer that triggers an event when the timer reaches zero.	CO2
15	Use JavaScript to dynamically add elements to the DOM.	CO3
16	Implement a feature to hide and show content on button click using JavaScript.	CO4
17	Develop a simple image gallery with clickable thumbnails and larger image display.	CO3
18	Set up a Flask application and create a route that displays a welcome message.	CO4
19	Build a user registration form using Flask-WTF for form handling.	CO5
20	Implement user login functionality with session management using Flask.	CO3
21	Develop a basic to-do list application using Flask for backend and JavaScript for frontend interaction.	CO2
22	Add functionality to mark tasks as completed and delete tasks from the list.	CO2
23	Make an AJAX request to an external API and display the data on your webpage.	CO3
24	Set up Flask-Socket IO for real-time chat functionality between users.	CO4
25	Create a database model for blog posts and implement CRUD operations (Create, Read, Update, and Delete).	CO3
26	Allow users to submit comments on blog posts and display them dynamically.	CO4
27	Design and build a personal portfolio website showcasing your projects and skills.	CO3

INTRODUCTION TO NEURAL NETWORKS & DEEP LEARNING LAB

Department:	**Department of Computer Science and Engineering**		
Course Name: Introduction to Neural Networks & Deep Learning Lab	**Course Code**	**L-T-P**	**Credits**
	ENMM756	0-0-2	1
Type of Course:	Major		
Pre-requisite(s), if any:			

Lab Experiments

Defined Course Outcomes

COs	
CO 1	Acquire a practical understanding of neural networks and deep learning algorithms through hands-on lab experiments.
CO 2	Develop proficiency in implementing feedforward neural networks and understanding their underlying principles.
CO 3	Demonstrate the ability to create and manipulate deep neural networks for solving complex real-world problems.
CO 4	Analyze and evaluate the performance of neural network models using appropriate evaluation metrics and techniques.

Ex. No	Experiment Title	Mapped CO/COs
1	Familiarize students with the lab environment, software, and tools.	CO1
2	Compare the information flow in a simple biological neural network (such as a single neuron) with a corresponding ANN architecture. Analyze how information is processed and propagated through each system and identify similarities and differences.	CO1
3	Implement a basic feedforward neural network using a library/framework of choice.	CO2
4	Train and test the neural network using a simple dataset to classify inputs.	CO2
5	Review the architecture and working principles of feedforward neural networks.	CO2
6	Explore different activation functions and their effects on network performance.	CO1
7	Implementation of different Learning Rules.	CO2
8	Implementation of Perceptron Networks.	CO2
9	Introduce the concepts of neural networks, artificial neurons, and activation functions.	CO1
10	Analyze and implement a pre-trained Neural Network	CO3
11	Common issues and errors encountered during deep learning experiments,	CO3
12	Experiment with different regularization techniques (e.g., regularization, dropout) to mitigate overfitting and evaluate their impact on model performance.	CO3
13	Troubleshooting strategies and debugging techniques for deep learning experiments.	CO3
14	Compare different optimization algorithms (e.g., stochastic gradient descent, Adam, RMS prop) and analyse their effects on model convergence and	CO3

	performance.	
15	Investigate the impact of batch normalization on model convergence and performance.	CO3
16	Compare the performance of different loss functions (e.g., cross-entropy, mean squared error) for a specific task.	CO3
17	Implement a sequence-to-sequence model for machine translation using an encoder-decoder architecture.	CO3
18	Study the concept of information bottleneck in neural networks by systematically varying the network's capacity	CO4
19	Investigate the impact of network architecture on information flow and learning capabilities.	CO4
20	Project related to the application of machine learning in healthcare.	CO4
21	Project related to the application of machine learning in business analysis.	CO4
22	Analyze the overall network and check how different layers or units contribute to the network's decision-making process.	CO3

K.R. MANGALAM UNIVERSITY

LIFE SKILLS FOR PROFESSIONALS-II

Department:	**Department of Computer Science and Engineering**		
Course Name: **Life Skills for Professionals - II**	**Course Code** **AEC012**	**L-T-P** 3-0-0	**Credits** 3
Type of Course:	AEC		
Pre-requisite(s), if any:			

Brief Syllabus:

This course is a multifaceted initiative designed to enhance and optimize learner's communication practices across various platforms. This program integrates a range of strategies, tools, and techniques to foster effective communication, facilitate collaboration, and promote a cohesive information flow within the learner's area. This course is structured and comprehensive initiative designed to develop and improve individuals' aptitude across various cognitive and behavioral domains. This course incorporates a range of assessments, training modules, and activities to enhance critical thinking, problem-solving, decision-making, and other essential aptitudes required for personal and professional success.

UNIT WISE DETAILS

Unit Number: 1	**Title: Personality Improvement**	**No. of hours: 4**

Content Summary: Asking for and giving information, Offering and responding to offers, Requesting and responding to requests, Congratulating people on their success, Asking questions and responding politely, Apologizing and forgiving

Unit Number: 2	**Title: Ratio & its application**	**No. of hours: 8**

Content Summary: Time & Work, Time & Distance, Train, Boat & Stream, Permutation & combination, Probability

Unit Number: 3	Title: Arithmetic	No. of hours: 8
Content Summary: Inequalities, Log, progression, Mensuration, BODMAS		
Unit Number: 4	**Title: Presentation Skills**	**No. of hours: 8**
Content Summary: Presentation Skills, Telephone etiquettes, LinkedIn Profile and professional networking, Video resumes & Mock interview sessions.		
Unit Number: 5	**Title: Leadership skills**	**No. of hours: 4**
Content Summary: Nurturing future leaders, Increasing productivity of the workforce, Imparting Self-leadership, Executive leadership		

***Self-Learning Components:**

https://onlinecourses.nptel.ac.in/noc21_hs02/preview

Please Note:

1) Students are supposed to learn the Interactive Learning Modules on the internet.

2) Webinars and Podcasts/ Self-Assessment Tools/Case Studies and Projects

3) At least 5-10 % syllabus will be asked in end term exams from self-learning components

Reference Books:

Aggarwal, R. S. (2014). Quantitative aptitude (Revised edition).
Gladwell, M. (2021). Talking to strangers.
Scott, S. (2004). Fierce conversations.

Define Course Outcomes (CO)

COs	Statements
CO1	**Understand** and apply the fundamental theories, models, and principles of communication.
CO2	**Apply** ability to communicate effectively through spoken and written forms. It includes developing skills in public speaking, interpersonal communication, professional writing, and persuasive communication.
CO3	**Evaluate** the development of teamwork and collaboration skills. It includes activities such as group projects, team-building exercises, and simulations that allow students to practice effective communication and collaboration within diverse teams
CO4	**Improve** their communication skills in different professional and personal contexts, such as interviews, networking events, customer interactions, and interpersonal relationships
CO5	**Analyze** ideas and information clearly and concisely through spoken language. They will develop the ability to articulate their thoughts, use appropriate vocabulary, and convey their message with clarity.

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels© 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C2		-
C02	C3		-
CO3	C5		
CO4			P5
CO5	C5		P5

CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO 1	3	3	-	-	-	-	-	-	-	-	3	3
CO 2	3	2	-	-	-	-	-	-	-	-		3
CO 3	-	2	3	-	-	2	-	-	-	-	-	2
CO 4	-	3	2	-	-	-	-	2	-	-	-	3
CO 5	-	-	3	2	-	-	-		-	-	-	3

1=weakly mapped
2= moderately mapped
3=strongly mapped

CO-PSO Mapping

CO	PSO1	PSO2	PSO3	PSO4
CO 1	3	2	-	-
CO 2		2	-	-
CO 3	1	-	-	3
CO 4		2	-	-
CO 5	-	-	3	-

Relevance of the Syllabus to various indicators

Unit I	**Personality Improvement**
Local	Improve personality, enhance basic communication skills.
Regional	Recognize the importance of continuous learning and practice to maintain and further develop interpersonal ability.
National	Practice leadership strategies for solving problems within time constraints, as in competitive exams.
Global	Aligns with global trends in employment
Employability	Develop skills in real-life situations, such as academic exams, job interviews, and problem-solving scenarios.
Entrepreneurship	Learn to share ideas, listen to others, build consensus, and manage conflicts to achieve common goals in collaborative settings.

Skill Development	Develops Skills in public speaking, interpersonal communication, professional writing, and persuasive communication.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit II	**Ratio & its application**
Local	Recognize the importance of continuous learning and practice to maintain and further develop mental ability.
Regional	Practice attentive listening techniques, such as paraphrasing and asking clarifying questions.
National	Attentively listen to others, understand their perspectives, and respond appropriately.
Global	Aligns with global trends in employment
Employability	Develop skills in participating and contributing to group discussions, meetings, or presentations.
Entrepreneurship	Learn to share ideas, listen to others, build consensus, and manage conflicts to achieve common goals in collaborative settings.
Skill Development	Apply skills in real-life situations, such as academic exams, job interviews, and problem-solving scenarios.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	**Arithmetic**

Local	Improve number sense, enhance basic arithmetic skills and strengthen mental math abilities and speed.
Regional	-
National	Learn about Inequalities, Log, progression, Mensuration, BODMAS
Global	Recognize the importance of continuous learning.
Employability	Develop skills in participating and contributing to group discussions, meetings, or presentations.
Entrepreneurship	-
Skill Development	Recognize the importance of continuous learning and practice to maintain and further develop mental ability.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	**Presentation Skills**
Local	Attentively listen to others, understand their perspectives, and respond appropriately with timelines
Regional	-
National	Contributes to develop skill and improved productivity
Global	Aligns with global trends in understanding the deadlines.
Employability	Enhance the employability of individuals by developing essential skills and competencies sought by employers
Entrepreneurship	-
Skill Development	Strengthening critical thinking, problem-solving, memory, and other cognitive functions to improve overall mental agility and performance.

Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
SDG	SDG 4
NEP 2020	-
POE/4th IR	Aligns with the concepts employability

(DEPARTMENT ELECTIVE -I)
NATURAL LANGUAGE PROCESSING

Department:	**Department of Computer Science and Engineering**		
Course Name: NATURAL LANGUAGE PROCESSING	**Course Code**	**L-T-P**	**Credits**
	ENSP302	4-0-0	4
Type of Course:	Department Elective I (Minor)		
Pre-requisite(s), if any: Strong programming skills, particularly in Python.			
Brief Syllabus: The ultimate objective of NLP is to read, decipher, understand, and make sense of the human languages in a manner that is valuable It helps resolve ambiguity in language and adds useful numeric structure to the data for many downstream applications, such as speech recognition or text analytics.			
Total lecture, Tutorial and Practical Hours for this course:			
Lectures: 40	**Practice**		
	Tutorials:	**Lab Work:**	
Course Outcomes (COs) The ultimate objective of NLP is to read, decipher, understand, and make sense of the human languages in a manner that is valuable It helps resolve ambiguity in language and adds useful numeric structure to the data for many downstream applications, such as speech recognition or text analytics.			

COs	
CO 1	NLP helps resolve ambiguity in language and adds useful numeric structure to the data for many downstream applications, such as speech recognition or text analytics.

CO 2	Its goal is to build systems that can make sense of text and automatically.
CO 3	Perform tasks like translation, spell check, or topic classification

UNIT WISE DETAILS

Unit Number: 1	**Title: Introduction to NLP**	**No. of hours: 8**

Content Summary: Natural Language Processing in real world, What is language, Approached to NLP,

Build NLP model: Eights Steps for building NLP Model, Web Scrapping

Unit Number: 2	**Title: Text Representation**	**No. of hours: 8**

Content Summary: Basic Vectorization, One-Hot Encoding, Bag of Words, Bag of N Grams, TF-IDF, Pre-trained Word Embedding, Custom Word Embeddings, Vector Representations via averaging, Doc2Vec Model, Visualizing Embeddings using TSNW and Tensorbaord

Text Classification: Application of Text Classification, Steps for building text classification system, Text classification using Naïve Bayes Classifier, Logistic Regression, and Support Vector Machine, Neural embedding for Text Classification, text classification using deep learning, interpret text classification model

Unit Number: 3	**Title: Information Extraction**	**No. of hours: 8**

Content Summary: Applications of Information Extraction, Processes for Information Extraction. Key phrase Extraction, Named Entity Recognition, Disambiguation and linking of named entity, Relationship extraction

Chatbot: Real life applications of chatbot, Chatbot Taxonomy, Dialog Systems, Process of building a dialog, Components of Dialog System, End to End Approach, Rasa NLU

Unit Number: 4	**Title: NLP for social media**	**No. of hours: 8**

Content Summary: Application of NLP in social media, challenges with social media, Natural Language Processing for Social Data, Understanding Twitter Sentiments, Identifying memes and Fake News **NLP for E-Commerce**: E-commerce catalog, Search in E-Commerce, How to build an e-commerce catalog, Review and Sentiment Analysis, Recommendations for E-Commerce
Self-Learning Components:
Reference Books: Natural Language Processing with Python by Steven Bird, Ewan Klein and Edward Loper Foundations of Statistical Natural Language Processing by Christopher Manning and Hinrich Schütze

K.R. MANGALAM UNIVERSITY

NATURAL LANGUAGE PROCESSING LAB

Department:	**Department of Computer Science and Engineering**		
Course Name:	**Course Code**	**L-T-P**	**Credits**
NATURAL LANGUAGE PROCESSING LAB	**ENSP352**	0-0-2	1
Type of Course:	Department Elective I (Minor)		
Pre-requisite(s), if any: Strong programming skills, particularly in Python.			

List of Experiments

1. Write a program to scrap website
2. Write a program to inspect website using developer tool
3. Write a program to request permission to scrap website
4. Write a program to inspect H1 element
5. Write a program to inspect table element
6. Write a program to create column list
7. Write a program to clean column list
8. Write a program to word tokenization
9. Write a program to implement RegEx for word tokenization
10. Write a program to implement stopwords
11. Write a program to implement LSTM

IMAGE PROCESSING & COMPUTER VISION

Department:	**Department of Computer Science and Engineering**		
Course Name: **Image Processing & Computer Vision**	**Course Code**	**L-T-P**	**Credits**
	ENSP304	4-0-0	4
Type of Course:	Minor		
Pre-requisite(s), if any: (1) Linear Algebra and (2) programming in python			
Brief Syllabus: The syllabus for the subject "Image Processing and Computer Vision using Python" covers the following topics: introduction to image processing and computer vision, Python programming basics for image processing, image acquisition and manipulation using Python libraries, image enhancement techniques, image filtering and convolution, feature extraction and object detection, image segmentation and boundary detection, image registration and alignment, camera calibration and 3D reconstruction, deep learning for image classification and object recognition, and applications of computer vision in fields like robotics, healthcare, and autonomous systems. The syllabus emphasizes hands-on programming exercises and projects to develop practical skills in implementing image processing and computer vision algorithms using Python.			
UNIT WISE DETAILS			
Unit Number: 1	**Title: Introduction to Basic Concepts of Image Formation**	**No. of hours: 8**	
Content Summary: Fundamentals and Applications of image processing, Image processing system components, Image sensing and acquisition, Sampling and quantization, Neighbors of pixel adjacency connectivity, regions and boundaries ,Distance measures. Image Enhancement: Frequency and Spatial Domain, Contrast Stretching, Histogram Equalization, Low pass and High pass filtering.			

Unit Number: 2	Title: Image Restoration and coloring	No. of hours: 10

Content Summary:

Model of The Image Degradation Restoration Process, Noise Models, Restoration in the presence of Noise Only Spatial Filtering, Periodic Noise Reduction by Frequency Domain Filtering, Linear Position Invariant Degradations, Estimation of Degradation Function, Inverse filtering, Wiener filtering, Constrained Least Square Filtering, Geometric Mean Filter, Geometric Transformations.

Colour Image Processing, Image Segmentation, Texture Descriptors, Colour Features, Edges/Boundaries, Object Boundary and Shape Representations, Interest or Corner Point Detectors, Speeded up Robust Features, Saliency.

Unit Number: 3	Title: Image Compression and Segmentation	No. of hours: 8

Content Summary:

Data Redundancies, Image Compression models, Elements of Information Theory, Lossless and Lossy compression, Huffman Coding, Shanon-Fano Coding, Arithmetic Coding, Golomb Coding, LZW Coding, Run Length Coding, Loss less predictive Coding, Bit Plane Coding, Image compression standards.

Image Segmentation and Morphological Image Processing: Discontinuity based segmentation, similarity-based segmentation, Edge linking and boundary detection, Threshold, Region based Segmentation Introduction to Morphology, Dilation, Erosion, Some basic Morphological Algorithms Object

Unit Number: 4	Title: Object Representation and Computer Vision Techniques	No. of hours: 12

Content Summary:

Representation and description and Computer Vision Techniques: Introduction to Morphology, Some basic Morphological Algorithms, Representation, Boundary Descriptors, Regional Descriptors, Chain Code, and Structural Methods. Review of Computer Vision applications; Artificial Neural Network for Pattern Classification, Convolutional Neural Networks, Machine Learning Algorithms and their Applications

in Image Segmentation, Motion Estimation and Object Tracking, Gesture Recognition, Face and Facial Expression Recognition, Image Fusion

***Self-Learning Components:**

Please Note:

1. **Concepts of Huffman coding, arithmetic coding, and other compression algorithms.**
2. **Presenting an overview of image compression standards (e.g., JPEG, JPEG2000) and their performance characteristics.**
3. **Presentation on a specific computer vision application (e.g., gesture recognition, facial expression recognition) and the underlying algorithms used**

Reference Books:

1. **Gonzalez Rafael C. and Woods Richard E., Digital Image Processing, New Delhi: Prentice– Hall of India.**
2. **M.K. Bhuyan , " Computer Vision and Image Processing: Fundamentals and Applications", CRC Press, USA, ISBN 9780815370840 - CAT# K338147**
3. **MOOCs course by Prof. M. K. Bhuyan, "Computer Vision and Image Processing - FundamentalsandApplications"https://onlinecourses.nptel.ac.in/noc21_ee23/course**
4. **Richard Szeliski, Computer Vision: Algorithms and Applications (1 ed.), Springer, 2011. ISBN 978-1848829350.**
5. **D. Forsyth and J. Ponce, Computer Vision: A Modern Approach (2 ed.), Prentice Hall, 2015. ISBN 978-9332550117.**

Define Course Outcomes (CO)

COs	Statements
CO1	**Understand** the fundamental concepts and techniques of image processing.
CO2	**Apply** image enhancement techniques for improving image quality.
CO3	**Analyze** the impact of different image enhancement techniques on image quality and visual perception.
CO4	**Evaluate** the strengths and limitations of computer vision techniques in various applications
CO5	**Develop** innovative image fusion techniques for combining multiple images to enhance visual perception

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels© 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C2	-	-
C02	C3	-	P2
CO3	C4	-	P3
CO4	C5	A3	-
CO5	C6	A4	P5

CO-PO Mapping

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	1	-	-	1	-	-	-	1	1	-	2
CO2	1	1	3	-	2	-	-	-	1	-	-	2
CO3	1	2	1	3	2	-	-	-	1	-	-	2
CO4	-	2	3	-	3	-	-	-	1	-	-	2
CO5	-	-	3	1	1	1	-	-	2	1	1	2

1=weakly mapped
2= moderately mapped
3=strongly mapped

CO-PSO Mapping

PO	PSO1	PSO2	PO3	PSO4
CO1	3	1	-	-
CO2	2	3	-	-
CO3	2	2	1	1
CO4	2	3	2	2
CO5	2	3	2	2

Relevance of the Syllabus to various indicators

Unit I	Introduction
Local	The fundamental concepts and applications of image processing can be relevant locally for various industries and research institutions that utilize image processing techniques.
Regional	-
National	-
Global	Image processing has global significance as it is used worldwide in multiple domains
Employability	Knowledge of image processing is in demand in the job market, and this syllabus can contribute to enhancing

	employability in related fields.
Entrepreneurship	-
Skill Development	Developing proficiency in image processing techniques
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit II	Image Restoration and coloring
Local	Local industries such as photography studios, design agencies, and media production houses can benefit from students' knowledge of image restoration techniques to improve image quality and enhance visual content.
Regional	-
National	Contributes to national digital literacy and internet connectivity strategies
Global	Aligns with globally relevant as image restoration and computer vision techniques are used worldwide in various domains such as healthcare, surveillance, augmented reality, and autonomous systems.
Employability	Proficiency in image restoration, coloring, and computer vision techniques enhances the employability in industries such as image processing, computer vision, multimedia, animation, and gaming.
Entrepreneurship	-
Skill Development	-
Professional	-

Ethics	
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	Image Compression and Segmentation
Local	-
Regional	-
National	-
Global	Image compression is a globally relevant field as the efficient transfer and storage of visual data is essential for various applications, including video streaming, social media, cloud services, and remote sensing.
Employability	Proficiency in image compression and segmentation techniques enhances the employability in industries such as multimedia, telecommunications, data analysis, and software development, where efficient handling of visual data is required.
Entrepreneurship	Students equipped with knowledge of image compression and segmentation techniques can explore entrepreneurial opportunities in areas such as image compression software development, image processing services, and innovative applications that rely on efficient image storage and transmission.
Skill Development	Develops critical skills in image compression algorithms, information theory, coding techniques, and image segmentation methodologies, empowering students to analyze, process, and optimize visual data effectively.
Professional Ethics	-
Gender	-
Human Values	-

Environment & Sustainability	-
Unit IV	Object Representation and Computer Vision Techniques
Local	The knowledge of object representation and computer vision techniques can contribute to the development of local industries and sectors that utilize computer vision technologies, such as surveillance systems, robotics, autonomous vehicles, and augmented reality applications.
Regional	-
National	-
Global	Object representation and computer vision techniques have global relevance due to their widespread applications in fields like image and video processing, computer graphics, virtual reality, and human-computer interaction, impacting global technological advancements.
Employability	The knowledge of object representation and computer vision techniques enhances the employability in industries related to computer vision, machine learning, and artificial intelligence, where the ability to develop and deploy computer vision algorithms and systems is highly valued.
Entrepreneurship	Students equipped with object representation and computer vision skills can explore entrepreneurial opportunities by developing innovative computer vision-based products, services, or solutions for industries such as retail, entertainment, healthcare, and security
Skill Development	Develops knowledge and skills in object representation, feature extraction, pattern recognition, and machine learning algorithms
Professional Ethics	-
Gender	-
Human Values	-

Environment & Sustainability	-
SDG	SDG 4,9
NEP 2020	Emphasizes skill development in areas such as image enhancement, restoration, compression, segmentation, and computer vision techniques, which are highly relevant in the digital era and align with the policy's focus on skill-based education.
POE/4th IR	Aligns with the concepts of Technological Advancement, Innovation and Entrepreneurship ETCR

K.R. MANGALAM UNIVERSITY

IMAGE PROCESSING & COMPUTER VISION LAB

Department:	**Department of Computer Science and Engineering**		
Course Name: **Image Processing & Computer Vision Lab**	**Course Code**	**L-T-P**	**Credits**
	ENSP354	0-0-2	1
Type of Course:	Minor		
Pre-requisite(s), if any: (1) Linear Algebra and (2) programming in python			

Proposed Lab Experiments

Defined Course Outcomes

COs	
CO 1	**Apply** image processing techniques using Python libraries.
CO 2	**Analyze** and evaluate the effectiveness of different image enhancement algorithms
CO 3	**Implement** image restoration algorithms and evaluate their performance in the presence of noise.
CO 4	**Develop** image compression algorithms and analyze their impact on image quality.
CO 5	**Formulate** computer vision techniques such as object detection and tracking, gesture recognition, and facial expression recognition using Python.

Ex. No	Experiment Title	Mapped CO/COs
1	Image acquisition and display using OpenCV library	CO 1
2	Image enhancement techniques: contrast stretching, histogram equalization	CO 2
3	Low-pass and high-pass filtering for image smoothing and sharpening	CO 2
4	Image degradation and restoration: modeling degradation process, noise reduction	CO 3
5	Inverse filtering and Wiener filtering for image restoration	CO 3
6	Geometric mean filter for image denoising	CO 3
7	Geometric transformations: translation, rotation, scaling	CO 1
8	Color image processing: color space conversion, histogram-based operations	CO 2
9	Image segmentation using thresholding techniques	CO 1
10	Texture analysis and feature extraction	CO 2
11	Edge detection and boundary extraction	CO 2
12	Interest point detection using Harris corner detector	CO 2
13	Speeded up robust features (SURF) for feature extraction	CO 2
14	Saliency detection in images	CO 2
15	Lossless and lossy image compression using Huffman coding	CO 4
16	Shanon-Fano coding and arithmetic coding for image compression	CO 4
17	Golomb coding and LZW coding for data compression	CO 4
18	Run-length coding for image compression	CO 4
19	Lossless predictive coding for image compression	CO 4
20	Bit plane coding for image compression	CO 4
21	Image segmentation based on discontinuity and similarity	CO 1

22	Edge linking and boundary detection in images	CO 1
23	Morphological operations: dilation and erosion	CO 1
24	Object representation and description using morphological algorithms	CO 1
25	Introduction to computer vision applications	CO 1
26	Pattern classification using artificial neural networks	CO 5
27	Convolutional neural networks for image classification	CO 5
28	Machine learning algorithms for image segmentation	CO 5
29	Motion estimation and object tracking	CO 5
30	Gesture recognition and face/facial expression recognition	CO 5

Detailed syllabus

Session 1: Image acquisition and display using OpenCV library
• Session: Introduction to image acquisition and display using OpenCV library
• Exercise: Write a Python code to capture and display images using OpenCV
• Project: Build a simple application to capture images from a webcam and display them in real-time
Session 2: Image enhancement techniques: contrast stretching, histogram equalization
• Session: Introduction to image enhancement techniques
• Exercise: Implement contrast stretching and histogram equalization algorithms in Python
• Project: Apply image enhancement techniques on a set of images and compare the results
Session 3: Low-pass and high-pass filtering for image smoothing and sharpening
• Session: Understanding low-pass and high-pass filters for image processing
• Exercise: Implement low-pass and high-pass filters in Python for image smoothing and sharpening
• Project: Apply filters on a set of images and analyze the effects of smoothing and sharpening
Session 4: Image degradation and restoration: modeling degradation

process, noise reduction
• Session: Introduction to image degradation and restoration
• Exercise: Model image degradation process and implement noise reduction techniques
• Project: Restore a set of degraded images using various restoration methods
Session 5: Inverse filtering and Wiener filtering for image restoration
• Session: Understanding inverse filtering and Wiener filtering for image restoration
• Exercise: Implement inverse filtering and Wiener filtering algorithms in Python
• Project: Apply restoration techniques on a set of images and evaluate the performance
Session 6: Geometric mean filter for image denoising
• Session: Introduction to geometric mean filter for image denoising
• Exercise: Implement geometric mean filter in Python for denoising images
• Project: Apply the filter on noisy images and compare the results with other denoising techniques
Session 7: Geometric transformations: translation, rotation, scaling
• Session: Introduction to geometric transformations in image processing
• Exercise: Implement translation, rotation, and scaling operations on images using OpenCV
• Project: Apply geometric transformations on a set of images and analyze the transformations' effects
Session 8: Color image processing: color space conversion, histogram-based operations
• Session: Understanding color image processing techniques
• Exercise: Perform color space conversion and histogram-based operations on images
• Project: Apply color image processing techniques on a set of images and analyze the results
Session 9: Image segmentation using thresholding techniques
• Session: Introduction to image segmentation using thresholding techniques
• Exercise: Implement thresholding algorithms for image segmentation in Python
• Project: Segment images using various thresholding methods and evaluate the segmentation results
Session 10: Texture analysis and feature extraction
• Session: Understanding texture analysis and feature extraction methods
• Exercise: Extract texture features from images using texture analysis

algorithms
• Project: Apply texture analysis and feature extraction techniques on images and analyze the extracted features
Session 11: Edge detection and boundary extraction
• Session: Introduction to edge detection and boundary extraction
• Exercise: Implement edge detection algorithms in Python
• Project: Detect edges and extract boundaries from a set of images using different edge detection methods
Session 12: Interest point detection using Harris corner detector
• Session: Understanding interest point detection using Harris corner detector
• Exercise: Implement Harris corner detection algorithm in Python
• Project: Detect interest points in images and analyze their properties using the Harris corner detector
Session 13: Speeded up robust features (SURF) for feature extraction
• Session: Introduction to SURF (Speeded Up Robust Features) algorithm
• Exercise: Implement SURF algorithm for feature extraction in Python
• Project: Extract features from images using SURF and evaluate their robustness and speed
Session 14: Saliency detection in images
• Session: Understanding saliency detection in images
• Exercise: Implement saliency detection algorithm in Python
• Project: Detect salient regions in images and analyze their significance using the implemented algorithm
Session 15: Lossless and lossy image compression using Huffman coding
• Session: Introduction to image compression using Huffman coding
• Exercise: Implement Huffman coding for lossless image compression in Python
• Project: Compress a set of images using Huffman coding and evaluate the compression ratio and quality
Session 16: Shanon-Fano coding and arithmetic coding for image compression
• Session: Understanding Shanon-Fano coding and arithmetic coding for image compression
• Exercise: Implement Shanon-Fano coding and arithmetic coding algorithms in Python
• Project: Compare the performance of Shanon-Fano coding and arithmetic coding for image compression
Session 17: Golomb coding and LZW coding for data compression
• Session: Introduction to Golomb coding and LZW (Lempel-Ziv-Welch)

coding for data compression
• Exercise: Implement Golomb coding and LZW coding algorithms in Python
• Project: Apply Golomb coding and LZW coding on data and analyze the compression efficiency
Session 18: Run-length coding for image compression
• Session: Understanding run-length coding for image compression
• Exercise: Implement run-length coding algorithm in Python
• Project: Compress images using run-length coding and analyze the compression performance
Session 19: Lossless predictive coding for image compression
• Session: Introduction to lossless predictive coding for image compression
• Exercise: Implement lossless predictive coding algorithm in Python
• Project: Apply predictive coding on images and evaluate the compression results
Session 20: Bit plane coding for image compression
• Session: Understanding bit plane coding for image compression
• Exercise: Implement bit plane coding algorithm in Python
• Project: Apply bit plane coding on images and analyze the compression efficiency
Session 21: Image segmentation based on discontinuity and similarity
• Session: Introduction to image segmentation based on discontinuity and similarity
• Exercise: Implement image segmentation algorithms using discontinuity and similarity measures
• Project: Segment images based on different segmentation criteria and evaluate the results
Session 22: Edge linking and boundary detection in images
• Session: Understanding edge linking and boundary detection in images
• Exercise: Implement edge linking algorithms for boundary detection in Python
• Project: Detect and link edges to extract boundaries from images using various edge linking methods
Session 23: Morphological operations: dilation and erosion
• Session: Introduction to morphological operations in image processing
• Exercise: Implement dilation and erosion operations using morphological algorithms
• Project: Apply morphological operations on images to analyze their effects on different objects
Session 24: Object representation and description using morphological algorithms

• Session: Understanding object representation and description using morphological algorithms
• Exercise: Implement object representation and description techniques using morphological operations
• Project: Represent and describe objects in images using morphological algorithms and analyze the results
Session 25: Introduction to computer vision applications
• Session: Overview of computer vision applications and use cases
• Exercise: Explore different computer vision applications and their functionalities
• Project: Choose a specific computer vision application, implement it, and demonstrate its capabilities
Session 26: Pattern classification using artificial neural networks
• Session: Introduction to pattern classification using artificial neural networks
• Exercise: Implement an artificial neural network for pattern classification in Python
• Project: Train a neural network model to classify patterns and evaluate its performance
Session 27: Convolutional neural networks for image classification
• Session: Understanding convolutional neural networks (CNNs) for image classification
• Exercise: Implement a CNN architecture in Python for image classification
• Project: Train a CNN model on a dataset for image classification and evaluate its accuracy
Session 28: Machine learning algorithms for image segmentation
• Session: Introduction to machine learning algorithms for image segmentation
• Exercise: Implement machine learning algorithms for image segmentation in Python
• Project: Apply machine learning techniques for image segmentation and analyze the segmentation results
Session 29: Motion estimation and object tracking
• Session: Understanding motion estimation and object tracking techniques
• Exercise: Implement motion estimation and object tracking algorithms in Python
• Project: Track objects in video sequences using motion estimation and analyze the tracking performance
Session 30: Gesture recognition and face/facial expression recognition
• Session: Introduction to gesture recognition and face/facial expression recognition

• Exercise: Implement gesture recognition and face/facial expression recognition algorithms in Python
• Project: Develop a system that can recognize gestures and facial expressions from video input

K.R. MANGALAM UNIVERSITY

INTRODUCTION TO GENERATIVE AI

Department:	**Department of Computer Science and Engineering**		
Course Name: **Introduction to Generative AI**	**Course Code**	**L-T-P**	**Credits**
	ENSP306	4-0-0	4
Type of Course:	Minor		
Pre-requisite(s), if any:			

Brief Syllabus:

This course introduces students to the fundamental concepts and techniques of Generative Artificial Intelligence (AI). Generative AI is an emerging field that focuses on developing algorithms and models capable of generating new content, such as images, music, and text. The course will cover the theoretical foundations of generative models and provide hands-on experience with open-source tools for creating and exploring generative AI applications.

UNIT WISE DETAILS

Unit Number: 1	**Title: Foundations of Generative AI**	**No. of hours: 12**

Content Summary:

Introduction to Generative AI: Definition, working and applications of generative AI, Historical overview and recent advancements, Ethical considerations and societal impact.
Probability and Statistics for Generative AI: Probability distributions and random variables, Maximum likelihood estimation, Bayesian inference and generative models.
Fundamentals of Deep Learning: Neural networks and their architectures, Backpropagation and optimization algorithms, Transfer learning and pre-trained models.

Unit Number: 2	**Title: Generative Models**	**No. of hours: 8**

Content Summary:

Overview of generative models: Gaussian Mixture Models, Hidden Markov Models; Representation learning and latent variables; Autoencoders: Basics of autoencoders and their applications, Encoder and decoder architectures, Reconstruction loss and latent space representation; Variational autoencoders (VAEs): Introduction to VAEs, reparameterization;

Unit Number: 3	Title: Generative Adversarial Networks and Flow-based Models	No. of hours: 8

Content Summary:

Generative Adversarial Networks (GANs)**:** Introduction, Architecture of GANs, Training GANs and understanding the loss functions; Autoregressive Models (including information-theoretic foundations)

Flow-based generative models and their advantages, Normalizing flows and invertible transformations, Training and sampling from flow-based models, Evaluation of Generative Models: Metrics for evaluating generative models (log-likelihood, Inception Score)

Unit Number: 4	Title: Applications and Future Directions	No. of hours: 12

Content Summary:

Real-World Applications of Generative AI: Image synthesis and editing, Data augmentation and data generation, Generative AI in healthcare, gaming, and art; **Ethical Considerations and Challenges:** Bias and fairness in generative models, Deepfakes and misinformation, Responsible AI practices; Emerging **Trends and Future Directions:** Reinforcement learning and generative models, Meta-learning and few-shot generation, OpenAI's DALL-E.

***Self-Learning Components:**

- Students are encouraged to explore and familiarize themselves with the tools of Python programming language for machine learning (NumPy, Pandas, PyTorch)
- Experiment with popular open-source tools: TensorFlow and Keras
- Presentation on current research areas like: style transfer, multimodal generation, and unsupervised learning.
- Open source tools for image: CycleGAN for image translation, StyleGAN and StyleGAN2 for high-quality image synthesis, OpenAI's CLIP for cross-model understanding
- Course on "Introduction to Generative AI" with Google Cloud

Reference Books:

1. Generative Deep Learning, by David Foster, 2nd Edition, O'Reilly Media, Inc.
2. Deep Learning by Ian Goodfellow, Yoshua Bengio and Aaron Courville , The MIT Press
3. PATTERN RECOGNITION AND MACHINE LEARNING by Christopher M. Bishop
4. Natural Language Processing with Python" by Steven Bird, Ewan Klein, and Edward Loper

Reference Links:

- Deep Learning Specialization on Coursera (includes a course on generative models):https://www.coursera.org/specializations/deep-learning
- TensorFlow Tutorials on Generative Models: https://www.tensorflow.org/tutorials/generative
- OpenAI's Generative Models page: https://openai.com/research/generative-models/

Define Course Outcomes (CO)

COs	Statements
CO1	**Understand** the foundational concepts of Generative AI
CO2	**Apply** probability distributions and random variables in generative models
CO3	**Employ** various generative models, such as VAEs, GANs, and flow-based models, to generate new data samples in different domains.
CO4	**Implement** and **Analyze** generative models
CO5	**Evaluate** emerging trends and future directions in the field of Generative AI
CO6	**Develop** sequence generation models using recurrent neural networks (RNNs) and LSTM

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels© 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C2	A1	P1
CO2	C3	A2	P2
CO3	C3	A3	P3
CO4	C4	A4	P4
CO5	C5	A4	P5
CO6	C6	A5	P5

CO-PO Mapping

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	2	-	-	-	2	-	2	-	2	-	3
CO2	3	3	-	2	2	-	-	-	-	-	-	3
CO3	-	-	3	-	-	-	2	2	2	3	-	2
CO4	-	3	2	3	3	-	2	-	-	-	-	2
CO5	-	-	-	2	3	-	2	-	2	-	1	2
CO6	-	-	1	2	-	3	-	-	3	3	2	-

1=weakly mapped
2= moderately mapped
3=strongly mapped

CO-PSO Mapping

PO	PO1	PO2	PO3	PSO4
CO1	3	-	-	-

CO2	3	3	-	-
CO3	3	3	3	-
CO4	-	3	-	-
CO5	3	3	-	-
CO6	-	3	3	-

Relevance of the Syllabus to various indicators

Unit I	**Foundations of Generative AI**
Local	-
Regional	-
National	-
Global	Understanding generative AI enables participation in global technological development and collaboration.
Employability	Proficiency in generative AI enhances employability in AI-related fields and industries.
Entrepreneurship	-
Skill Development	Develops technical skills in deep learning, probabilistic modeling
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit II	**Generative Models**
Local	-
Regional	-
National	-
Global	Technological development and innovation.

Employability	-
Entrepreneurship	-
Skill Development	Develops technical skills in autoencoders, VAEs, GANs, and autoregressive models.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	**Generative Adversarial Networks and Flow-based Models**
Local	-
Regional	-
National	-
Global	Developing algorithms and models
Employability	-
Entrepreneurship	-
Skill Development	Develops technical skills in Generative Adversarial Networks and Flow-based Models
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	**Applications and Future Directions**
Local	-

Regional	-
National	-
Global	Knowledge of emerging trends and future directions in the field of Generative AI
Employability	-
Entrepreneurship	-
Skill Development	Develops skills in applying generative models, analyzing their performance, and exploring future directions.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
SDG	4,8,9
NEP 2020	Development of a knowledge-based society and promotes interdisciplinary learning.
POE/4th IR	Advancement of AI technologies.

INTRODUCTION TO GENERATIVE AI LAB

Department:	**Department of Computer Science and Engineering**		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Introduction to Generative AI Lab	**ENSP356**	0-0-2	1
Type of Course:	Minor		
Pre-requisite(s), if any:			

Proposed Lab Experiments

Defined Course Outcomes

COs	
CO 1	**Utilize** Python programming to generate random samples from various probability distributions
CO 2	**Apply** knowledge of generative AI models and frameworks
CO 3	**Develop** proficiency in building and training feedforward neural networks and deep learning frameworks
CO 4	**Implement** basic autoencoder models and train them on datasets
CO5	**Evaluate** the performance metrics of trained models, such as accuracy and loss

Ex. No	Experiment Title	Mapped CO/COs
1	Generate random samples from various probability distributions (e.g., normal distribution, uniform distribution) using Python	CO1
2	Implement maximum likelihood estimation (MLE) for a given dataset and estimate the parameters of a selected probability distribution.	CO1
3	Explore and experiment with existing generative AI models and frameworks (e.g., TensorFlow, PyTorch).	CO2
4	Implement a basic generative AI model (e.g., a simple image generator) using a chosen framework.	CO2
5	Implement a feedforward neural network using a deep learning framework (e.g., TensorFlow, PyTorch).	CO3
6	Train the neural network on a benchmark dataset (e.g., MNIST, CIFAR-10) using backpropagation and a chosen optimization algorithm (e.g., stochastic gradient descent).	CO3
7	Evaluate the trained model's performance metrics (e.g., accuracy, loss) on a separate validation set.	CO5
8	Compare and analyze the performance of the trained model with and without transfer learning.	CO5
9	Train an autoencoder on a dataset of images.	CO4
10	Encode a set of images using the trained encoder and visualize their corresponding latent space representations	CO4
11	Build an encoder and a decoder architecture for a VAE using a deep learning framework.	CO4
12	Train the VAE on a dataset of images (e.g., MNIST, CIFAR-10) using a chosen loss function	CO4
13	Implement a basic autoencoder model and train it on a dataset.	CO4

14	Implement an autoregressive model, such as PixelCNN or PixelRNN, using a deep learning framework.	CO3
15	Implement a GAN architecture using a deep learning framework.	CO3
16	Train the GAN on a dataset of images (e.g., MNIST, CIFAR-10) and monitor the generator and discriminator losses.	CO5
17	Analyze the loss functions used in GAN training (e.g., adversarial loss, feature matching loss)	CO5
18	Train an RNN-based model to generate sequences (e.g., text or music)	CO3
19	Train the RNN on a dataset of sequences (e.g., text corpus, MIDI data) using backpropagation through time (BPTT)	CO3
20	Implement a flow-based generative model using a deep learning framework	CO3
21	Fine-tune a pre-trained deep learning model on a new task or dataset.	CO3
22	Implement the Tacotron model using a deep learning framework.	CO3
23	Implement the CycleGAN model using a deep learning framework.	CO3
24	Implement the evaluation metrics using appropriate libraries or frameworks.	CO5
25	Evaluate the performance of different generative models using the implemented metrics.	CO5

Projects:

Project 1: Random Data Analysis
Description: In this project, you will generate random samples from various probability distributions using Python and analyze the generated data.
Tasks:
1. Choose three different probability distributions (e.g., normal distribution, uniform distribution, exponential distribution).
2. Write Python code to generate random samples from each distribution using appropriate libraries or functions.
3. Plot histograms or density plots to visualize the generated samples for each distribution.
4. Compute and display summary statistics (e.g., mean, standard deviation) for each generated sample.
5. Compare the characteristics of the generated samples from different distributions and analyze their differences.
Project 2: Parameter Estimation for a Probability Distribution
Description: In this project, you will implement the maximum likelihood estimation (MLE) algorithm to estimate the parameters of a selected probability distribution based on a given dataset.
Task:
1. Choose a probability distribution (e.g., normal distribution, exponential distribution) and specify its probability density function (PDF) or probability mass function (PMF).
2. Generate a synthetic dataset based on the chosen distribution with known parameters.
3. Implement the MLE algorithm using Python to estimate the parameters of the distribution from the synthetic dataset.
4. Compare the estimated parameters with the known true parameters and calculate the estimation error.
5. Repeat the process for multiple iterations with different dataset sizes to analyze the performance of the MLE algorithm with varying sample sizes.
Project 3: Exploring Pre-trained Generative AI Models
Description: In this mini project, you will explore and experiment with existing pre-trained generative AI models using popular frameworks such as TensorFlow

or PyTorch.
Task:
1. Choose a specific generative AI model, such as a pre-trained image generation model (e.g., GAN, VAE) or a text generation model (e.g., language model).
2. Set up the chosen deep learning framework (e.g., TensorFlow or PyTorch) and load the pre-trained model.
3. Understand the input and output requirements of the model and experiment with generating new samples.
4. Analyze and interpret the generated samples in terms of their quality, diversity, or other relevant metrics.
5. Explore different input variations or techniques to generate customized or specific samples.
Project 4: Image Classification using Feedforward Neural Network
Description: In this mini project, you will implement a feedforward neural network using a deep learning framework and train it for image classification on a benchmark dataset.
Task:
1. Choose a benchmark dataset for image classification, such as MNIST or CIFAR-10.
2. Set up the chosen deep learning framework (e.g., TensorFlow or PyTorch) and define the architecture of the feedforward neural network for image classification.
3. Preprocess the dataset, including normalization and splitting into training and validation sets.
4. Implement the training loop for the feedforward neural network using backpropagation and a chosen optimization algorithm (e.g., stochastic gradient descent).
5. Evaluate the trained model's performance metrics (e.g., accuracy, loss) on the separate validation set and analyze the results.

K.R. MANGALAM UNIVERSITY

TRANSFER LEARNING

Department:	**Department of Computer Science and Engineering**		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Transfer Learning	**ENSP308**	4-0-0	4
Type of Course:	Minor		
Pre-requisite(s), if any:			

Brief Syllabus:

The "Transfer Learning" syllabus covers an Introduction to transfer learning, pretrained models, fine-tuning, and feature extraction. Deep transfer learning techniques, including domain adaptation and multi-task learning. Practical implementation with data preprocessing, model adaptation, and experimentation. Applications in computer vision, NLP, and challenges like negative transfer and ethical considerations.

UNIT WISE DETAILS

Unit Number: 1	**Title: Introduction to Transfer Learning**	**No. of hours: 8**

Content Summary:

Fundamentals of transfer learning, motivation, and applications. Pretrained models and their usage. Fine-tuning and feature extraction. Transfer learning frameworks and libraries. Transfer learning strategies and techniques. Evaluation and performance metrics for transfer learning models. Case studies and real-world applications.

Unit Number: 2	**Title: Transfer Learning Techniques and Algorithms**	**No. of hours: 8**

Deep transfer learning methods, including domain adaptation, instance transfer, and multi-task learning. Transfer learning in convolutional neural networks (CNNs), recurrent neural networks (RNNs), and generative models. Ensemble-based transfer learning approaches. Handling domain shift and dataset bias. Advanced transfer

learning algorithms and architectures.

Unit Number: 3	Title: Practical Implementation of Transfer Learning	No. of hours: 12

Data preprocessing and feature extraction for transfer learning. Fine-tuning and model adaptation techniques. Transferring knowledge across different domains and tasks. Implementing transfer learning in popular deep learning frameworks. Experimentation, analysis, and fine-tuning of transfer learning models.

Unit Number: 4	Title: Transfer Learning Applications and Challenges	No. of hours: 12

Content Summary:

Transfer learning in computer vision, natural language processing (NLP), and speech recognition. Transfer learning for specific domains like healthcare, finance, and social media analysis. Challenges in transfer learning, such as negative transfer and overfitting. Ethical considerations and fairness in transfer learning. Latest research trends and future directions in transfer learning.

***Self-Learning Components:**

- Students can leverage online platforms like Coursera, edX, and Udemy to access transfer learning courses. These resources provide in-depth knowledge, practical examples, and hands-on exercises to enhance their understanding
- Students can explore and experiment with open-source libraries and frameworks like TensorFlow, PyTorch, and Scikit-learn. These tools provide pre-trained models, sample code, and documentation that facilitate self-learning and experimentation.
- https://www.coursera.org/lecture/convolutional-neural-networks/transfer-learning

**students will demonstrate the self-learning components through classroom presentations*

Reference Books:

- "Transfer Learning" by Sinno Jialin Pan and Qiang Yang (Morgan & Claypool Publishers)
- "Transfer Learning for Natural Language Processing" by Shervin Minaee and Amirali Abdolrashidi (Springer)

- "Domain Adaptation in Computer Vision Applications" by Gabriela Csurka (Morgan & Claypool Publishers)
- "Transfer Learning in Reinforcement Learning" by Panpan Cai, Yang Yu, and Xuewen Yao (Springer)

Text Books

- "Transfer Learning: Algorithms and Applications" by B. K. Tripathy and Sandipan Roy (Springer)
- "Transfer Learning and Domain Adaptation in NLP" by Yoshua Bengio, Jian-Yun Nie, and Geoffrey J. Gordon (Morgan & Claypool Publishers)
- Transfer Learning: Methods, Applications, and Challenges" by R. Sathya and K. S. Devi (CRC Press)

Define Course Outcomes (CO)

COs	Statements
CO1	Understand transfer learning concepts and principles.
CO 2	Apply transfer learning techniques to real-world problems.
CO 3	Analyze and evaluate the performance of transfer learning algorithms
CO 4	Synthesize and adapt pre-trained models for specific tasks.
CO 5	Critically think, propose innovative approaches, and effectively communicate transfer learning concepts

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels© 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C1,C2	A1	P1
CO2	C3	A2	P2
CO3	C4	A3	P3
CO4	C5	A4	-
CO5	C5	-	P5

CO-PO Mapping

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	2	1	1	-	-	-	1	-	-	1
CO2	3	2	3	2	3	3	-	-	2	2	-	2
CO3	3	3	3	2	3	-	-	-	2	2	-	3
CO4	3	-	1	-	1	-	-	-	1	-	-	2
CO5	3	1	1	3	2	-	-	-	1	3	-	2

Relevance of the Syllabus to various indicators

Unit I	**Introduction to Transfer Learning**
Local	Applying transfer learning to address specific local challenges, utilizing local frameworks and libraries, and evaluating the performance of transfer learning models on local datasets
Regional	Exploring regional datasets and case studies to understand the effectiveness of transfer learning in addressing regional challenges

National	Understanding the national context and specific needs for transfer learning applications
Global	Exploring transfer learning applications and advancements on a global scale, collaborating with international research communities, and addressing global challenges through the development of transfer learning models and frameworks
Employability	Developing practical skills in transfer learning, enhancing job prospects in various domains such as computer vision, natural language processing, and data analysis,
Entrepreneurship	Empowering students to explore innovative applications of transfer learning, fostering an entrepreneurial mindset to create new products, services, and solutions
Skill Development	Enhancing technical skills, problem-solving abilities, critical thinking, and collaboration in transfer learning
Professional Ethics	
Gender	
Human Values	
Environment & Sustainability	
Unit II	**Transfer Learning Techniques and Algorithms**
Local	Addressing local challenges and requirements with advanced transfer learning techniques
Regional	Leveraging transfer learning to address regional data patterns and challenges.
National	Using transfer learning to tackle national-level data characteristics and requirements.
Global	Leveraging transfer learning for global-scale data analysis and knowledge sharing.
Employability	Enhancing employability through the application of transfer learning in real-world scenarios

Entrepreneurship	Exploring entrepreneurial opportunities in transfer learning for developing innovative solutions
Skill Development	Enhancing technical skills and expertise in transfer learning methodologies and applications.
Professional Ethics	Adhering to ethical principles and responsible conduct in transfer learning research and applications.
Gender	
Human Values	
Environment & Sustainability	
Unit III	**Practical Implementation of Transfer Learning**
Local	Developing expertise in data preprocessing and feature extraction techniques relevant to local data and domain-specific tasks
Regional	Implementing fine-tuning and model adaptation techniques suitable for regional datasets and specific regional tasks
National	Transferring knowledge across different domains and tasks to address national-level challenges and improve performance on national datasets.
Global	Understanding and implementing transfer learning in popular deep learning frameworks used globally to solve diverse problems
Employability	Acquiring practical skills in experimentation, analysis, and fine-tuning of transfer learning models, enhancing employability in the field of machine learning and artificial intelligence.
Entrepreneurship	Exploring real-world applications and identifying opportunities for entrepreneurial ventures using transfer learning techniques
Skill Development	Developing proficiency in data preprocessing, feature extraction, model adaptation, and experimentation, fostering overall skill development in the field of transfer learning.
Professional	

Ethics	
Gender	
Human Values	
Environment & Sustainability	
Unit IV	**Transfer Learning Applications and Challenges**
Local	Applying transfer learning techniques in computer vision, NLP, and speech recognition tasks specific to the local context
Regional	Exploring transfer learning applications in specific regional domains like healthcare, finance, and social media analysis
National	Addressing national challenges and opportunities by leveraging transfer learning in various domains.
Global	Staying updated with the latest research trends and future directions in transfer learning, contributing to the global knowledge base
Employability	Acquiring skills in transfer learning for computer vision, NLP, and speech recognition, enhancing employability in diverse industries.
Entrepreneurship	Identifying innovative applications of transfer learning in domains like healthcare, finance, and social media to create entrepreneurial opportunities.
Skill Development	Developing expertise in addressing challenges like negative transfer and overfitting, advancing overall skill development in transfer learning.
Professional Ethics	
Gender	
Human Values	
Environment & Sustainability	

SDG	SDG 4
NEP 2020	Under NEP 2020, collaborative learning environments are promoted to foster interactive and engaging teaching practices, which are supported by the implementation of transfer learning.
POE/4th IR	Transfer learning is highly relevant in the context of the Fourth Industrial Revolution (4th IR), facilitating the adaptation and application of existing knowledge and models to drive innovation, efficiency, and social impact across various domains.

K.R. MANGALAM UNIVERSITY

TRANSFER LEARNING LAB

Department:	**Department of Computer Science and Engineering**		
Course Name: **Transfer Learning Lab**	**Course Code**	**L-T-P**	**Credits**
	ENSP358	0-0-2	1
Type of Course:	Minor		
Pre-requisite(s), if any:			

Proposed Lab Experiments

Defined Course Outcomes

COs	
CO 1	Understand the fundamentals of transfer learning, including its motivation and applications.
CO 2	Gain proficiency in implementing transfer learning techniques and algorithms using Python.
CO 3	Apply fine-tuning and model adaptation techniques to transfer knowledge across different domains and tasks
CO 4	Evaluate and measure the performance of transfer learning models using appropriate metrics in Python.
CO5	Analyze case studies and real-world applications of transfer learning to understand its practical implications

Ex. No	Experiment Title	Mapped CO/COs

1	Fine-tuning Pretrained Models using Python	CO1
2	Introduction to Transfer Learning using Python	CO1
3	Exploring Pretrained Models in Python	CO2
4	Implementing Transfer Learning with Pretrained Models in Python	CO2
5	Feature Extraction from Pretrained Models using Python	CO2
6	Implementing a deep transfer learning model using Python libraries like TensorFlow or PyTorch to address domain shift in image classification tasks.	CO3
7	Fine-tuning a pre-trained CNN architecture, such as VGG or ResNet, in Python to transfer knowledge from a source task to a target task.	CO2
8	Implementing a multi-task learning approach using Python libraries like Keras or TensorFlow to train an RNN model for multiple related tasks in natural language processing.	CO3
9	Building an ensemble of transfer learning models in Python by combining predictions from multiple pre-trained models to improve performance in image recognition tasks.	CO3
10	Utilizing Python libraries like TensorFlow or PyTorch to implement generative models, such as GANs, to handle dataset bias and generate synthetic data for transfer learning.	CO3
11	Preprocess a dataset using Python libraries like NumPy and Pandas, perform data augmentation techniques, and extract relevant features for transfer learning tasks.	CO4
12	Implement fine-tuning techniques using Python and deep learning frameworks like TensorFlow or PyTorch to adapt pre-trained models to new target tasks.	CO3
13	Transfer knowledge from a pre-trained model in one domain to a different domain or task using Python and transfer learning techniques like domain adaptation or	CO2

	multi-task learning.	
14	Implement transfer learning using popular deep learning frameworks like TensorFlow or PyTorch. Students can choose a specific framework, load pre-trained models, and fine-tune them for their desired tasks.	CO3
15	Design experiments to evaluate different transfer learning strategies and techniques, analyze the performance of transfer learning models using Python, and fine-tune the models based on the analysis results	CO4
16	Implement transfer learning techniques using Python and deep learning frameworks like TensorFlow or PyTorch for computer vision tasks such as image classification, object detection, or image segmentation.	CO3
17	Apply transfer learning methods in Python and NLP libraries like NLTK or spaCy for tasks like sentiment analysis, text classification, or named entity recognition.	CO3
18	Utilize Python and speech recognition libraries like SpeechRecognition or PyAudio to develop transfer learning models for speech recognition tasks.	CO3
19	Choose a specific domain like healthcare, finance, or social media analysis and implement transfer learning techniques in Python to address domain-specific tasks such as medical image classification, financial sentiment analysis, or social media text classification.	CO5
20	Apply transfer learning concepts to time-series data using Python and deep learning frameworks like TensorFlow or PyTorch to solve tasks like forecasting, anomaly detection, or sensor data analysis	CO4
21	Study recent research papers on transfer learning and implement cutting-edge transfer learning algorithms or architectures in Python to stay updated with the latest trends in the field	CO5

Semester III

<table>
<tr><td>Department:</td><td colspan="3">Department of Computer Science and Engineering</td></tr>
<tr><td rowspan="2">Course Name: Advance Concepts in Software Engineering & Project Management</td><td>Course Code</td><td>L-T-P</td><td>Credits</td></tr>
<tr><td>ENMM816</td><td>3-1-0</td><td>4</td></tr>
<tr><td>Type of Course:</td><td colspan="3">Major</td></tr>
<tr><td colspan="4">Pre-requisite(s), if any: Basics of Computer Programming</td></tr>
<tr><td colspan="4">Brief Syllabus:
Solving computational problems requires the knowledge of efficient data organization and the ability to make effective choices among multiple solutions. In this course, we will explore several fundamental data structures in computer science and learn to implement them. The course aims to teach the fundamentals of data structures, their design, implementation and effective use in problem solving approach. With the knowledge of data structures and practical experience in implementing them, students can become much more effective designer and developer. The course will start with the basic introduction of linear such as arrays, stack and queues as well as non-linear data structures such as trees and graphs. They will further proceeds with the programming intensive task of implementing them.</td></tr>
<tr><td colspan="4">UNIT WISE DETAILS</td></tr>
<tr><td>Unit Number: 1</td><td colspan="2">Title: Software Engineering Principles and Paradigms</td><td>No. of hours: 8</td></tr>
<tr><td colspan="4">Content Summary:
Overview of software engineering paradigms: Waterfall, Agile, DevOps, etc. Software process models: Rational Unified Process (RUP), Scrum, Kanban, etc. Requirements engineering: Elicitation, analysis, specification, validation. Software architecture and design principles. Software quality assurance and testing methodologies. Case study: Comparing software development methodologies.</td></tr>
<tr><td>Unit Number: 2</td><td colspan="2">Title: Advanced Software Design and Development</td><td>No. of hours: 12</td></tr>
<tr><td colspan="4">Content Summary:
Object-oriented design principles and design patterns. Component-based development and service-oriented architecture. Designing for scalability, performance, and maintainability. Software reusability and code quality metrics. Emerging technologies and their impact on software development (e.g.,</td></tr>
</table>

microservices, serverless). Case study: Designing and developing a complex software system.

Unit Number: 3	Title: Software Project Management Strategies	No. of hours: 12

Content Summary:
Project planning and estimation techniques (PERT, CPM, agile estimation). Risk management and mitigation in software projects. Resource allocation, team management, and conflict resolution. Change management and handling scope creep. Agile project management tools (e.g., JIRA, Trello) and techniques. Case study: Managing a software project using Agile methodologies**.**

Unit Number: 4	Title: Software Quality Management and Verification	No. of hours: 10

Content Summary:
Software testing strategies: Unit, integration, system, acceptance testing. Test automation and continuous integration/continuous deployment (CI/CD). Code reviews, static analysis, and dynamic analysis. Ensuring software security and addressing vulnerabilities. Monitoring and improving software quality over the development lifecycle. Case study: Implementing a comprehensive software testing and quality assurance process.

Unit Number: 5	Emerging Trends and Project Management Challenges	No. of hours: 10

Cloud computing and its impact on software engineering. Big data and analytics in software development and project management. DevOps practices and their integration with project management. Ethical considerations in software engineering (privacy, bias, etc.). Handling distributed teams and global project management challenges. Case study: Adapting to new technologies and managing global software teams.

***Self-Learning Components:**
1. Students should explore Platforms like Coursera, edX, and Udemy offer courses on software engineering principles and Agile practices.
2. Choose an open-source project and analyze its software development process. Compare the methodologies used and their impact on the project's success.

Reference Books:

2. Software Engineering: A Practitioner's Approach" by Roger S. Pressman or "Agile Estimating and Planning" by Mike Cohn for an in-depth understanding of software engineering and Agile

methodologies.

Define Course Outcomes (CO)

COs	Statements
CO1	**Analyze** various software development methodologies and paradigms, comparing their strengths and weaknesses in different project scenarios.
CO2	**Understand, Design & Implement** Understand advanced software design principles, including design patterns and component-based development, and their application in solving complex software challenges.
CO3	**Articulate** Articulate the benefits and challenges of emerging technologies in software development and project management, including cloud computing, microservices, and DevOps practices.
CO4	**Understand, Design & implement** Understand the principles of software architecture and apply them to design and implement modular and maintainable software systems.

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels© 2. Knowledge 3. Understand 4. Apply 5. Analyze 6. Evaluate 7. Create	Affective levels(A) 2. Receiving 3. Responding 4. Valuing 5. Organizing 6. Characterizing	Psychomotor levels(P) 2. Imitation 3. Manipulation 4. Precision 5. Articulation 6. Improving
CO1	C1,C2	A3	P5
C02	C3	A3	P4
CO3	C4	A4	P3
CO4	C5	A2	P2

***Please Note:**
Map only 1 or 2 Levels in each category. If a higher level is given, no need

to mention lower level

CO-PO Mapping

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	2	-	-	1	-	-	-	-	2	-
CO2	3	3	2	-	-	-	-	-	-	-	2	-
CO3	3	3	3	-	-	-	-	-	-	-	2	-
CO4	3	3	3	-	-	-	-	-	-	-	2	-

Please Note:

- Refer to POs while mapping each CO.
- Mark " – " if not applicable
- If attainment of a CO is strongly mapped with a PO , Mark 3
- If attainment of a CO is moderately mapped with a PO , Mark 2
- If attainment of a CO is weakly mapped with a PO , Mark 1

Justification for mapping must be relevant

1=weakly mapped
2= moderately mapped
3=strongly mapped

CO-PSO Mapping

PSO	PSO1	PSO2	PSO3	PSO4
CO1	3	3	2	-
CO2	3	3	1	-
CO3	3	3	-	-
CO4	3	3	-	-

Relevance of the Syllabus to various indicators

Unit I	**Software Engineering Principles and Paradigms**
Local	-

Regional	-
National	The syllabus covers a wide range of software development methodologies, which are applicable both nationally and globally, allowing students to understand and adapt to diverse project environments.
Global	The inclusion of emerging trends and technologies ensures that students are prepared to contribute effectively in both national and global software development scenarios.
Employability	The course equips students with advanced software engineering skills, making them more attractive to employers seeking professionals with a deep understanding of software design, development, and project management.
Entrepreneurship	Understanding software architecture, design patterns, and project management strategies empowers students to envision and create innovative software products, fostering an entrepreneurial mindset.
Skill Development	The syllabus emphasizes practical skill development through hands-on coding, design exercises, and software testing. These skills are directly transferable to real-world software engineering roles.
Professional Ethics	The inclusion of ethical considerations in software engineering underscores the importance of ethical decision-making, instilling a sense of responsibility and integrity in future software professionals.
Gender	-
Human Values	The syllabus includes ethical discussions, emphasizing the importance of empathy, respect, and social responsibility in software engineering and project management.
Environment & Sustainability	-
Unit II	**Advanced Software Design and Development**
Local	-
Regional	-
National	The syllabus covers a wide range of software development methodologies, which are applicable both nationally and globally, allowing students to understand and adapt to diverse project environments.
Global	The inclusion of emerging trends and technologies ensures that students are prepared to contribute

	effectively in both national and global software development scenarios.
Employability	The course equips students with advanced software engineering skills, making them more attractive to employers seeking professionals with a deep understanding of software design, development, and project management.
Entrepreneurship	Understanding software architecture, design patterns, and project management strategies empowers students to envision and create innovative software products, fostering an entrepreneurial mindset.
Skill Development	The syllabus covers a wide range of software development methodologies, which are applicable both nationally and globally, allowing students to understand and adapt to diverse project environments.
Professional Ethics	The inclusion of ethical considerations in software engineering underscores the importance of ethical decision-making, instilling a sense of responsibility and integrity in future software professionals.
Gender	-
Human Values	The syllabus includes ethical discussions, emphasizing the importance of empathy, respect, and social responsibility in software engineering and project management.
Environment & Sustainability	-
Unit III	**Software Project Management Strategies**
Local	-
Regional	-
National	The syllabus covers a wide range of software development methodologies, which are applicable both nationally and globally, allowing students to understand and adapt to diverse project environments.
Global	The inclusion of emerging trends and technologies ensures that students are prepared to contribute effectively in both national and global software development scenarios.
Employability	The course equips students with advanced software engineering skills, making them more attractive to employers seeking professionals with a deep understanding of software design, development, and project management.

Entrepreneurship	Understanding software architecture, design patterns, and project management strategies empowers students to envision and create innovative software products, fostering an entrepreneurial mindset.
Skill Development	The syllabus covers a wide range of software development methodologies, which are applicable both nationally and globally, allowing students to understand and adapt to diverse project environments.
Professional Ethics	The inclusion of ethical considerations in software engineering underscores the importance of ethical decision-making, instilling a sense of responsibility and integrity in future software professionals.
Gender	-
Human Values	The syllabus includes ethical discussions, emphasizing the importance of empathy, respect, and social responsibility in software engineering and project management.
Environment & Sustainability	-
Unit IV	**Software Quality Management and Verification**
Local	-
Regional	-
National	The syllabus covers a wide range of software development methodologies, which are applicable both nationally and globally, allowing students to understand and adapt to diverse project environments.
Global	The inclusion of emerging trends and technologies ensures that students are prepared to contribute effectively in both national and global software development scenarios.
Employability	The course equips students with advanced software engineering skills, making them more attractive to employers seeking professionals with a deep understanding of software design, development, and project management.
Entrepreneurship	Understanding software architecture, design patterns, and project management strategies empowers students to envision and create innovative software products, fostering an entrepreneurial mindset.
Skill Development	The syllabus covers a wide range of software development methodologies, which are applicable both nationally and globally, allowing students to understand and adapt to diverse project environments.

Professional Ethics	The inclusion of ethical considerations in software engineering underscores the importance of ethical decision-making, instilling a sense of responsibility and integrity in future software professionals.
Gender	-
Human Values	The syllabus includes ethical discussions, emphasizing the importance of empathy, respect, and social responsibility in software engineering and project management.
Environment & Sustainability	-
SDG	SDG 9
NEP 2020	-
POE/4th IR	Aligns with the concepts of Design, Efficiency ,Problem Solving, Abstraction and System Analysis
Unit V	**Emerging Trends and Project Management Challenges**
Local	-
Regional	-
National	The syllabus covers a wide range of software development methodologies, which are applicable both nationally and globally, allowing students to understand and adapt to diverse project environments.
Global	The inclusion of emerging trends and technologies ensures that students are prepared to contribute effectively in both national and global software development scenarios.
Employability	The course equips students with advanced software engineering skills, making them more attractive to employers seeking professionals with a deep understanding of software design, development, and project management.
Entrepreneurship	Understanding software architecture, design patterns, and project management strategies empowers students to envision and create innovative software products, fostering an entrepreneurial mindset.
Skill Development	The syllabus covers a wide range of software development methodologies, which are applicable both nationally and globally, allowing students to understand and adapt to diverse project environments.
Professional Ethics	The inclusion of ethical considerations in software engineering underscores the importance of ethical decision-making, instilling a sense of responsibility and

	integrity in future software professionals.
Gender	-
Human Values	The syllabus includes ethical discussions, emphasizing the importance of empathy, respect, and social responsibility in software engineering and project management.
Environment & Sustainability	-
SDG	SDG 9
NEP 2020	-
POE/4th IR	Aligns with the concepts of Design, Efficiency ,Problem Solving, Abstraction and System Analysis

Benchmarking Universities (atleast 4-5 standard university contents must be reffered):

Name of the University	Program referred for Syllabus contents	Subject which is referred

(DEPARTMENT ELECTIVE-II)

COMPUTATIONAL SERVICES IN THE CLOUD

Department:	**Department of Computer Science and Engineering**		
Course Name: Computational Services in The Cloud	**Course Code**	**L-T-P**	**Credits**
	ENSP401	4-0-0	4
Type of Course:	Minor (Department Elective III)		
Pre-requisite(s), if any:			
Brief Syllabus: This course covers evolutionary computing paradigms from multi-processor systems to Cloud, Edge, and Fog Computing. The course design follows cloud deployment models, service models, virtualization techniques and cloud architectural solutions. It also elaborates cloud compliances and security at fine-grained level by following a shared responsibility model. It reveals a design pattern to the students, enabling them to think through the process of designing and implementing cloud infrastructure and optimal IT solutions			
Unit Number: 1	**Title: Introduction to Cloud Computing**		**No. of hours: 10**
Content Summary: Cloud Computing, Adoption of cloud-based IT resources, Service Models: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), Software-as-a-Service(SaaS), Deployment models: Public Cloud, Private Cloud, Hybrid Cloud, Community Cloud, Cloud Computing Characteristics, Challenges of cloud computing, Virtualization concept, Types of virtualizations, Demo of virtualization, Virtualization Merits, Role of virtualization in cloud computing, Virtualization Demerits, VMPlacement, VM Migration, VM Migration Demo, VM clustering, Design Issues in VM Clustering, Need of Dockers and Containers, Docker Eco-System, Hypervisor vs			

Docker.

Unit Number: 2	Title: Microservices	No. of hours: 10

Content Summary:

Microservices, Service-Oriented Architecture, REST API, IP Addressing, Subnetting, Supernetting, Designing of Virtual Private Cloud, Demo of VPC, VPC Peering,VPC Case Study, Cloud Storage, Serverless Computing, Cloud API Gateway, Cloud Databases, Resource Provisioning, Time shared and space shared, Efficient VMConsolidation on cloud server, Task/DAG Scheduling Algorithms, Min-Min, Max-Min, MET, B-level Demo, T-level Demo, Task-VM Mapping, Auto Scaling, Load Balancing.

Unit Number: 3	Title: Case Study	No. of hours: 10

Content Summary:

Case Study: Cloud Market analysis, Security and Compliances, Shared security model in IAAS/PAAS/SAAS, Shared technology issues, Data loss or leakage, Accountor service hijacking, Implementation of cloud security, Security Groups, Network Access Control Lists, Cloud databases, Parallel Query Execution with NoSQLDatabase, Big Data, Handling Big Data on Cloud Platform, Map- Reduce framework for large clusters using Hadoop, Design of data applications based on Map Reducein Apache Hadoop.

Unit Number: 4	Title: Comparative study/analysis of public clouds	No. of hours: 10

Content Summary:

Comparative study/analysis of public clouds, Edge Computing, Fog Computing, Data Offloading, Cloud-Based DevOps Tools, Task Partitioning, Data Partitioning, Data Synchronization, Distributed File System, Data center, Ongoing Research Topics.

***SELF-LEARNING COMPONENTS:**

The students are expected to choose a topic in discussion with the industry expert and implement the concepts of cloud computing. The student should present the progress at the end of the semester.

Please Note:

1)Students are supposed to learn the components on self-basis

2) At least 5-10 % syllabus will be asked in end term exams from self-learning components.

Reference Books:

1. Lizhe Wang, Rajiv Ranjan, Jinjun Chen and Boualem Benatallah, Cloud Computing (1 ed.), CRC Press, 2017. ISBN 978-1351833097.
2. Judith S. Hurwitz and Daniel Kirsch, Cloud Computing For Dummies (2 ed.), Hoboken: John Wiley & Sons, 2020. ISBN 978-1119546658.
3. Prerna Sharma, Moolchand Sharma and Mohamed Elhoseny, Applications of Cloud Computing (1 ed.), CRC Press, 2020. ISBN 9780367904128.

Define Course Outcomes (CO)

COs	**Statements**
CO1	Explain the core concepts of the cloud computing paradigm: how and why this paradigm shift came about, the characteristics, advantages and challenges brought about by the various models and services in cloud computing.
CO2	Apply the fundamental concepts in datacenters to understand the tradeoffs in power, efficiency and cost.
CO3	Identify resource management fundamentals, i.e. resource abstraction, sharing and sandboxing and outline their role in managing infrastructure in cloud computing.

CO4	Analyze various cloud programming models and apply them to solve problems on the cloud.

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels© 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C3	A3	P4
CO2	C3	A4	P2
CO3	C2	A2	P1
CO4	C4	A5	P5

CO-PO Mapping

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	1	1	-	-		2	1	2	2	1
CO2	2	2	2	1	-	-	1	3	-	1	2	2

CO3	3	3	3	-	1	2	-	2	1	-	1	1
CO4	3	3	2	1	-	2	-	2	2	-	2	1

Justification for mapping must be relevant.

1=weakly mapped
2= moderately mapped
3=strongly mapped

CO-PSO Mapping

PO	PSO1	PSO2	PSO3	PSO4
CO1	2	2	1	3
CO2	3	3	2	3
CO3	1	1	3	1
CO4	1	1	1	2

Relevance of the Syllabus to various indicators

Unit I	**Introduction to Design thinking**
Local	-
Regional	-
National	-
Global	It promotes international collaboration, data sharing, and connectivity, facilitating global trade, research, and innovation.
Employability	Proficiency in cloud computing technologies, service models, and deployment models enhances employability and career prospects.

Entrepreneurship	-
Skill Development	This includes skills in virtualization, cloud service models, deployment models, and containerization.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit II	**Case studies in Design thinking**
Local	-
Regional	-
National	-
Global	In the global context, microservices, service-oriented architecture, and cloud computing are essential technologies driving digital innovation and transformation. Global organizations heavily rely on these technologies to deliver scalable and resilient applications and services to a global user base.
Employability	Acquiring skills in microservices, service-oriented architecture, and cloud computing enhances employability prospects in the IT industry. Proficiency in these technologies is in high demand as organizations increasingly adopt cloud-based architectures and microservice-oriented approaches
Entrepreneurship	-
Skill Development	The course on microservices, service-oriented architecture, and cloud computing promotes skill development in areas such as cloud infrastructure design, API development, virtualization, storage management, and resource provisioning

Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	**Design Frameworks**
Local	-
Regional	-
National	-
Global	It helps address global challenges such as data privacy, security breaches, and international data transfer regulations.
Employability	Knowledge of cloud market analysis, security, and compliance enhances employability in various roles, including cloud architects, cloud security specialists, and cloud consultants.
Entrepreneurship	-
Skill Development	Studying cloud market analysis, security, and compliance develops critical skills such as risk assessment, security implementation, and compliance management.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	**Innovation & Creativity**
Local	-
Regional	-

National	-
Global	Cloud computing is a global phenomenon, and a course on comparative study/analysis of public clouds and related topics provides students with a global perspective on cloud technologies.
Employability	Proficiency in cloud computing technologies and understanding the different deployment models, such as public clouds and edge computing, enhances employability prospects in the IT industry.
Entrepreneurship	-
Skill Development	A course on comparative study/analysis of public clouds and related topics enhances various skills such as research, critical analysis, problem-solving, and technical proficiency.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
SDG	SDG 4, 9
NEP 2020	Integration of Emerging Technologies and Skill Development
POE/4th IR	The course on cloud computing aligns with the Fourth Industrial Revolution (IR 4.0) in the following ways: Advanced Infrastructure, Data Management and Analytics and Digital Transformation.

COMPUTATIONAL SERVICES IN THE CLOUD LAB

Department:	**Department of Computer Science and Engineering**		
Course Name: Computational Services in The Cloud Lab	**Course Code**	**L-T-P**	**Credits**
	ENSP451	0-0-2	1
Type of Course:	Minor (Department Elective III)		
Pre-requisite(s), if any: Nil			

Proposed Lab Experiments

Defined Course Outcomes

COs	
CO 1	**Apply** the concept cloud computing to solve practical use cases.
CO 2	**Analyzing** different services in cloud computing
CO 3	**Evaluate** different available services provided by cloud vendors
CO 4	**Design** Cloud based application

List of Programs

Ex No	Experiment Title	Mapped CO/COs
1	Install Virtualbox / VMware Workstation with different flavours of linux or windows OS	CO1
2	Install a C compiler in the virtual machine created using virtual box and execute Simple Programs	CO1
3	Install Google App Engine. Create hello world app and other simple web applications using python/java.	CO1
4	Use GAE launcher to launch the web applications.	CO1
5	Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim.	CO2
6	Find a procedure to transfer the files from one virtual machine to another virtual machine	CO2
7	Find a procedure to launch virtual machine using trystack (Online Openstack Demo Version)	CO2
8	Install Hadoop single node cluster and run simple applications like wordcount.	CO1
9	Deploy a simple web application using a Platform as a Service (PaaS) offering like AWS Elastic Beanstalk, Azure App Service, or Google App Engine	CO4
10	Create and manage databases using services like AWS RDS, Azure SQL Database, or Google Cloud SQL.	CO3
11	Containerize an application using Docker and create a container registry on a cloud platform.	CO3
12	Deploy and manage containers using container orchestration tools like AWS ECS, Azure Kubernetes Service (AKS), or Google Kubernetes Engine (GKE)	CO4
13	Develop and deploy a serverless function using AWS Lambda, Azure Functions, or Google Cloud Functions	CO4
14	Configure event triggers, access permissions, and monitoring for serverless functions	CO2
15	Utilize cloud-based data analytics tools like AWS Athena, Azure Data Lake Analytics, or Google BigQuery to query and analyze large datasets	CO2
16	Build and train machine learning models using cloud-based services like AWS SageMaker, Azure Machine Learning, or Google Cloud AutoML	CO2
17	Use infrastructure provisioning tools such as AWS CloudFormation, Azure Resource Manager, or Google Cloud	CO1

	Deployment Manager to define and deploy infrastructure components	
18	Set up cloud monitoring and logging services like AWS CloudWatch, Azure Monitor, or Google Cloud Monitoring to track the performance and health of cloud resources	CO3
19	Implement access control policies and roles using AWS IAM, Azure Active Directory, or Google Cloud Identity and Access Management (IAM)	CO4
20	Enable encryption for data at rest and in transit using cloud security services	CO3
21	Explore cost optimization techniques like auto-scaling, spot instances, or reserved instances to optimize cloud resource usage and reduce costs	CO2

K.R. MANGALAM UNIVERSITY

MICROSOFT AZURE CLOUD FUNDAMENTALS

Department:	**Department of Computer Science and Engineering**		
Course Name: Microsoft Azure Cloud Fundamentals	**Course Code**	**L-T-P**	**Credits**
	ENSP403	4-0-0	4
Type of Course:	Minor (Department Elective III)		
Pre-requisite(s), if any:			

Brief Syllabus:

The Microsoft Azure Cloud Fundamentals course introduces the concepts of cloud computing and the Azure platform. It covers Azure services such as Virtual Machines, Storage, Networking, Identity, App Services, and Databases. The course focuses on security, monitoring, and management in Azure. Real-world case studies and hands-on labs enable practical application. The subject equips students to design trustworthy intrusion detection systems and enhances security in IoT networks using Azure.

UNIT WISE DETAILS

Unit Number: 1	**Title: Introduction to Cloud Computing**	**No. of hours: 10**

Content Summary: Introduction to Cloud Computing and its Characteristics, Benefit, and Challenges of cloud computing. Cloud service models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). Cloud Deployment Models, for example, Public, private, hybrid, and community clouds. shared responsibility model. Identify appropriate use cases for each cloud model. Consumption-based model. Comparison of cloud pricing models.

Unit Number: 2	**Title: Introduction to Microsoft Azure**	**No. of hours: 10**

Content Summary: Microsoft Azure cloud platform and its significance in the industry, Azure regions, and availability zones for understanding global data center distribution. Introduction of various Azure services and solutions available for different scenarios.

Unit Number: 3	Title: Azure Virtual Machines (VMs) and Storage	No. of hours: 10

Content summary: Create and manage virtual machines using Azure. Different VM sizes and types based on performance requirements. VM scaling and load balancing for optimizing application performance. Azure storage services: Blob Storage, Table Storage, File Storage, and Disk Storage.

Unit Number: 4	Title: Azure Networking, Identity and Access Management	No. of hours: 10

Content Summary: creation and configuration of virtual networks and subnets in Azure. Azure Load Balancer for distributing incoming network traffic, VPN Azure Active Directory (Azure AD) for managing identities and authentication. Gateway for secure communication between on-premises networks, and Azure. Azure Active Directory (Azure AD) for managing identities and authentication. Database Services, Azure Storage Account for data storage and retrieval.

***SELF-LEARNING COMPONENTS:**

1. https://nptel.ac.in/courses/106105167
2. https://www.mygreatlearning.com/academy/learn-for-free/courses/microsoft-azure-essentials

Please Note:

1)Students are supposed to learn the components on self-basis

2) At least 5-10 % syllabus will be asked in end term exams from self-learning components.

Reference Books:

1. Microsoft Azure Essentials Fundamentals of Azure by Michael S. Collier and Robin E. Shahan
2. Mastering Microsoft Azure Infrastructure Services by John Savill
3. Azure for Architects by Ritesh Modi

Text Books

Microsoft Azure For Dummie by Timothy L. Warner and Katrina Dow, and Michael Washam

Define Course Outcomes (CO)

COs	Statements
CO1	**Understand** Microsoft Azure concepts
CO2	**Express** proficiency in the handling of Azure services
CO3	**Determine** methods to create and manipulate virtual machines
CO4	**Identify** commonly used models to implement cloud network
CO5	**Articulate** Azure database services

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels© 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C2	-	P1
C02	C3	-	P2
CO3	C3	-	P4
CO4	C1	-	-

CO5	C1	-	P4

CO-PO Mapping

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	–	–	–	–	1	1	1	–	–	–	1
CO2	3	3	3	–	3	–	–	–	–	2	–	–
CO3	3	3	2	3	3	–	–	–	–	–	–	2
CO4	2	2	2	2	2	–	–	–	–	–	–	–
CO5	2	2	–	2	2	–	–	–	–	2	–	–

CO-PSO Mapping

PO	PSO1	PSO2	PSO3	PSO4
CO1	2	–	–	–
CO2	3	3	3	–
CO3	3	3	–	–
CO4	2	2	–	2
CO5	2	2	2	2

Relevance of the Syllabus to various indicators

Unit I	**Introduction to Cloud Computing**
Local	-
Regional	-
National	-
Global	Develops skills in using cloud-based services
Employability	Develops skills in using cloud-based services
Entrepreneurship	-
Skill Development	Develops basic knowledge and skills regarding Cloud concepts as well as in cloud computing, network management, and Azure services
Professional Ethics	-

Gender	-
Human Values	-
Environment & Sustainability	-
Unit II	**Introduction to Microsoft Azure**
Local	Understanding of Microsoft Azure services
Regional	-
National	-
Global	Aligns with global trends of Cloud
Employability	Develops cloud-based Knowledge for cloud architects
Entrepreneurship	-
Skill Development	Develops conceptual knowledge of Cloud computing
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	**Azure Virtual Machines (VMs) and Storage**
Local	
Regional	-
National	-
Global	Designing Virtual machines in Azure and storage management
Employability	Develops skills regarding the cloud security and cloud architecture
Entrepreneurship	-

Skill Development	-
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	**Azure Networking, Identity, and Access Management**
Local	
Regional	-
National	-
Global	Access management of data
Employability	Develops skills in cloud security and network management
Entrepreneurship	-
Skill Development	Develops knowledge and skills in data storage and access management
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
SDG	SDG 4
NEP 2020	-
POE/4th IR	Aligns with the concepts of technology and coding

K.R. MANGALAM UNIVERSITY

MICROSOFT AZURE CLOUD FUNDAMENTALS LAB

Department:	**Department of Computer Science and Engineering**		
Course Name: Microsoft Azure Cloud Fundamentals Lab	**Course Code**	**L-T-P**	**Credits**
	ENSP453	0-0-2	1
Type of Course:	Minor (Department Elective III)		
Pre-requisite(s), if any: Nil			

Proposed Lab Experiments

Defined Course Outcomes

COs	
CO 1	**Acquire** a practical understanding of cloud computing through hands-on lab experiments.
CO 2	**Develop** proficiency in using global data centers.
CO 3	**Demonstrate** the ability to create virtual machines and handle data storage
CO 4	**Analyze** and maintain the security and access management system.

Ex. No.	**Experiment Title**	**Mapped CO/COs**
1	Familiarize students with the lab environment, software, and tools.	CO1
2	Creating and Managing Virtual Machines with Virtual Box	CO1

3	Create and configure virtual machines running different operating systems (e.g., Windows and Linux).	CO2
4	Install and configure Next Cloud on a local server or cloud-based virtual machine.	CO2
5	Set up your (user) accounts and storage quotas.	CO2
6	Try to implement -Test file uploads, sharing, and collaboration features	CO1
7	Install pfSense as a virtual router/firewall.	CO2
8	Configure virtual networks, subnets, and VLANs	CO2
9	Network Setup with pf Sense: for Test routing, port forwarding, and firewall rules.	CO1
10	Identity and Access Management with Key Cloak: Explore user authentication methods, roles, and permissions.	CO3
11	Identity and Access Management with Key Cloak Configure single sign-on (SSO) for different applications.	CO3
12	Install Key Cloak as an identity provider on a virtual machine.	CO3
13	Install Azure CLI and PowerShell on your machines.	CO3
14	Configure single sign-on (SSO) for different applications.	CO3
15	Use Azure CLI and PowerShell to create and manage Azure resources (e.g., VMs, storage accounts).	CO3
16	Create a simple web app (e.g., using HTML/CSS/JS or a web framework).	CO3
17	Deploy the web app to Azure App Service using Azure portal or Azure CLI.	CO3
18	Test the app's accessibility and scalability	CO4

19	Create an Azure SQL Database instance.	CO4
20	Monitor security alerts and take remedial actions.	CO4

K.R. MANGALAM UNIVERSITY

STORAGE AND DATABASES ON CLOUD

Department:	**Department of Computer Science and Engineering**		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Storages and Databases on Cloud	**ENSP405**	4-0-0	4
Type of Course:	Minor (Department Elective III)		
Pre-requisite(s), if any:			

Brief Syllabus:

The course on cloud databases and storage provides a comprehensive understanding of the principles, technologies, and best practices associated with storing and managing data in the cloud. The syllabus covers various topics, starting with an introduction to cloud computing and an exploration of different types of cloud storage and databases, including object storage, block storage, file storage, relational databases, NoSQL databases, and more. Students delve into popular cloud storage and database services, such as Amazon S3, Google Cloud Storage, and Azure Blob Storage gaining practical knowledge of their features, deployment options, scalability, and high availability. The curriculum also includes essential aspects like database design, data migration, security measures, backup and recovery strategies, performance optimization, and monitoring techniques. Real-world case studies provide insights into organizations utilizing cloud storage and databases effectively.

UNIT WISE DETAILS

Unit Number: 1	**Title: Introduction to Storage on cloud**	**No. of hours: 10**

Content Summary:

Introduction to Cloud Computing, Overview of cloud databases and cloud storages, types of cloud storages(Object, block and file), different types of cloud database management systems, Gartner Magic Quadrant for Cloud Database Management Systems, Advantages of Working with Cloud Databases, Considerations for Cloud Databases, Top Cloud Database, Factors that help in choosing the right cloud database, Challenges involved in using cloud storages and databases.

Unit	**Title: Data Integration, Migration,**	**No. of hours: 10**

Number: 2	Security and performance on cloud	

Content Summary:

Techniques, tool, methods and considerations for migrating from premise database to cloud databases

Backup, Recovery, and Disaster Planning including automated backups, point-in-time recovery and replication

Performance Optimization and Monitoring including query optimization, indexing, caching, and monitoring tools

Scalability and High Availability: load balancing, replication, sharding, and auto-scaling, Cloud Data Warehousing

Unit Number: 3	Title: AWS Cloud Storage	No. of hours: 10

Content Summary:

Introduction to AWS cloud storage, AWS management console, AWS Storage Services, Uploading files and images , Creating a web server, Overview of Amazon S3, Storage Classes, EC2 Instance Storage, network file system Amazon Elastic Block Store, Amazon Elastic file system, Amazon Cloud Front.

Brief introduction to Google Cloud Storage, and Azure Blob Storage.

Unit Number: 4	Title: Case Study	No. of hours: 10

Content Summary: Case Studies and Real-world Examples of Netflix , Airbnb, Pinterest, spotify, coca-cola etc. Analyzing real-world use cases of organizations using cloud storage and databases, discussing architecture decisions, challenges, and lessons learned.

***Self-Learning Components:**

1) **E**xplore open-source projects related to cloud databases and storage. discover open-source projects from GitHub, learn from the codebase, and contribute to the development of cloud-based storage and database solutions. (https://github.com/topics/cloud-database)

2) Join the course : https://www.coursera.org/specializations/cloud-computing

2) Join Online Courses like

- https://www.udemy.com/topic/cloud-computing/
- https://www.coursera.org/courses?query=data%20storage
- https://www.mygreatlearning.com/academy/learn-for-free/courses/databases-and-files-systems-in-aws
- https://www.youtube.com/watch?v=EN4fEbcFZ_E
- https://www.codecademy.com/catalog/subject/cloud-computing
- https://www.simplilearn.com/tutorials/cloud-computing-tutorial

2) Read case studies of organizations that have successfully implemented cloud storage and database solutions.

- **Case Study of Delloite:** https://www2.deloitte.com/us/en/pages/consulting/articles/cloud-computing-case-studies.html
- **Case Study of Amazon:** https://aws.amazon.com/solutions/case-studies/amazon/

Learn about their challenges, architectural decisions, and best practices they adopted.

Please Note:

1)Students are supposed to learn the components on self-basis

2) At least 5-10 % syllabus will be asked in end term exams from self-learning components.

Reference Books:

1) "Database Cloud Storage: The Essential Guide to Oracle Automatic Storage Management" by Nitin Vengurlekar, 2013

2) "Cloud Database Development and Management" by Lee chao, 2013

3) "Advancing Cloud Database Systems and Capacity Planning with Dynamic Applications" by Narendra Kumar Kamila, 2017

Define Course Outcomes (CO)

COs	Statements
CO1	**Understand** & Recall the fundamental concepts and principles of cloud storage and databases. Interpret the security considerations and best practices for cloud storage and databases.
CO2	**Express** ideas and solutions for optimizing performance and improving efficiency in cloud storage and databases through indexing, caching, and query optimization techniques.
CO3	**Determine** the requirements of an application or system and determine the most suitable cloud storage and database solutions to meet those requirements.
CO4	**Identify** different types of cloud storage and database services available.
CO5	**Articulate** the best practices and considerations for designing scalable, reliable, and secure cloud storage and database architectures.
CO6	**Design** data storage architecture for cloud-based applications.

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels© 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C2	-	P1
C02	C3	-	P2
CO3	C3	-	P3
CO4	C2	-	-
CO5	C3	-	P1

CO-PO Mapping

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1	-	1	-	1	1	2	2	-	3
CO2	2	3	1	1	1	1	1	1	2	2	1	3
CO3	2	2	3	2	2	3	2	2	3	3	3	3
CO4	1	2	-	3	1	1	-	-	2	2	1	2
CO5	-	2	1	1	3	2	-	-	2	3	2	3
CO6	1	1	2	-	2	3	1	1	3	3	2	3

Justification for mapping must be relevant.

1=weakly mapped
2= moderately mapped
3=strongly mapped

CO-PSO Mapping

	PSO1	PSO2	PSO3	PSO4
CO1	2	3	1	3
CO2	3	2	2	2
CO3	3	2	3	3
CO4	1	1	2	1
CO5	2	3	2	2
CO6	1	1	3	1

Relevance of the Syllabus to various indicators

Unit I	Introduction to cloud databases and storages
Local	local considerations may include factors like local network connectivity, data centre locations, and latency between local systems and cloud databases.
Regional	regional considerations can include regulatory compliance specific to a particular region, data sovereignty issues, and availability zones
National	national considerations such as national data protection laws and regulations, government policies related to cloud computing, and national cybersecurity guidelines that may impact the usage and storage of data in cloud databases.
Global	Aligns with global trends in internet technologies and network protocols accessibility of cloud databases and storage options across multiple regions, data replication and backup strategies for disaster recovery on a global scale
Employability	The unit syllabus may indirectly contribute to employability by introducing students to cloud computing concepts, cloud databases, and storage technologies
Entrepreneurship	-
Skill Development	The unit syllabus primarily focuses on developing skills

	related to cloud computing, cloud databases, and storage
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit II	Data Integration, Migration, Security and performance on cloud
Local	local considerations may include factors like local network connectivity, data centre locations, and latency between local systems and cloud databases.
Regional	regional considerations can include regulatory compliance specific to a particular region, data sovereignty issues, and availability zones
National	national considerations such as national data protection laws and regulations, government policies related to cloud computing, and national cybersecurity guidelines that may impact the usage and storage of data in cloud databases.
Global	Aligns with global trends in internet technologies and network protocols accessibility of cloud databases and storage options across multiple regions, data replication and backup strategies for disaster recovery on a global scale
Employability	local considerations may include factors like local network connectivity, data centre locations, and latency between local systems and cloud databases.
Entrepreneurship	regional considerations can include regulatory compliance specific to a particular region, data sovereignty issues, and availability zones
Skill Development	national considerations such as national data protection laws and regulations, government policies related to cloud computing, and national cybersecurity guidelines that

	may impact the usage and storage of data in cloud databases.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	AWS Cloud Storage
Local	Primarily focuses on cloud storage technologies and their local implementation within the AWS ecosystem.
Regional	-
National	Contributes to national network security strategies and protocols
Global	AWS, Google Cloud, and Azure are global cloud service providers, enabling businesses worldwide to access their storage services and infrastructure.
Employability	Acquiring knowledge and skills in these cloud storage platforms enhances employability in the field of cloud computing and storage management.
Entrepreneurship	Knowledge of AWS, Google Cloud, and Azure storage services can empower entrepreneurs to build and scale their businesses by utilizing scalable and reliable cloud storage solutions for their applications and data.
Skill Development	The unit syllabus focuses on developing skills related to AWS cloud storage, including uploading files and images, creating a web server, utilizing different storage services
Professional Ethics	-
Gender	-
Human Values	-
Environment &	-

Sustainability	
Unit IV	Case Studies
Local	Addresses local understanding and implementation of internet-based services
Regional	-
National	Contributes to national digital communication strategies and multimedia applications
Global	Aligns with global trends in internet telephony, multimedia applications, and SEO
Employability	Develops skills in internet telephony, multimedia applications, and SEO
Entrepreneurship	-
Skill Development	Develops knowledge and skills in internet telephony, multimedia applications, and SEO
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
SDG	SDG 4
NEP 2020	It supports NEP's objective of enhancing the use of technology in education and research by providing students with knowledge and skills in cloud databases and storage, which are essential components of modern digital infrastructure
POE/4th IR	The course can incorporate discussions on ethical considerations in cloud databases and storage, addressing issues such as data privacy, security, and responsible data management.

STORAGE AND DATABASES ON CLOUD LAB

Department:	**Department of Computer Science and Engineering**		
Course Name: **Storages and Databases on Cloud Lab**	**Course Code**	**L-T-P**	**Credits**
	ENSP455	0-0-2	1
Type of Course:	Minor (Department Elective III)		
Pre-requisite(s), if any: Nil			

Proposed Lab Experiments

Defined Course Outcomes

COs	**Understand** the principles and concepts of cloud storage and databases.
CO 1	**Gain** hands-on experience in using cloud storage services for data storage and retrieval.
CO 2	**Implement** backup and recovery strategies for cloud-based storage and databases.
CO 3	**Optimize** database performance in the cloud through indexing and query optimization techniques.
CO 4	**Analyse** real-world case studies to understand the challenges and solutions related to cloud storage and databases.

Ex. No	Experiment Title	Mapped CO/COs
1	Create accounts and configure cloud storage services such as Amazon S3, Google Cloud Storage, or Azure Blob Storage: Practice creating buckets/containers, uploading files, setting access permissions, and managing storage resources.	CO1, CO4, CO5
2	Perform operations on object storage, including uploading, downloading, and deleting files: Explore advanced features like versioning, lifecycle policies, and metadata management.	CO1, CO2, CO5
3	Perform Block Storage Configuration: Create and attach storage volumes to virtual machines, perform formatting and mounting, and understand snapshotting and resizing operations.	CO1, CO3, CO5
4	File Storage Implementation: Work with shared file systems such as Amazon EFS, Google Cloud Filestore, or Azure Files. They configure file shares, mount them on virtual machines, and explore features like file locking and access control.	CO1, CO4, CO5
5	Database Provisioning and Management: • learn to set up and configure cloud databases like Amazon RDS, Google Cloud SQL, or Azure Database Services • create database instances, manage security settings, and perform basic administration tasks such as backups and restores.	CO1, CO3, CO4, CO5
6	Data Migration to the Cloud: • Practice migrating databases from on-premises or other cloud providers to the selected cloud database service. • Explore different migration methods, tools, and validate the successful transfer of data	CO1, CO3, CO4, CO5

7	Database Design and Schema Management: • Work on designing and creating database schemas for various scenarios. • Implement tables, relationships, indexes, and constraints to support specific use cases and understand database modelling concepts.	CO1, CO3, CO5, CO6
8	Data Security and Access Control: Configure security measures for cloud databases, including setting up user accounts, managing roles and permissions, and implementing encryption techniques to protect data at rest and in transit	CO1, CO5
9	Performance Tuning and Optimization: Learn performance tuning techniques for cloud databases, including query optimization, index creation, caching strategies, and monitoring tools to identify and resolve performance bottlenecks	CO1, CO2, CO5
10	High Availability and Scalability: • Explore features like replication, automatic scaling, and load balancing to ensure high availability and scalability of databases in the cloud • Configure and test failover scenarios and evaluate the performance impact of scaling operations.	CO1, CO3, CO5
11	Project Backup and Recovery Strategies: • Practice implementing backup and recovery mechanisms for cloud databases • perform regular backups, schedule automated backups, and simulate recovery scenarios to restore databases to a consistent state.	CO1, CO5
12	Project Cloud Data Warehousing: • Gain hands-on experience with cloud data warehousing platforms like Amazon Redshift • load data, run complex queries, and analyze performance optimization techniques for data warehousing.	CO1, CO2, CO5, CO6

13	**Project** Design and implement an inventory management system that utilizes cloud storage for storing inventory data. Include features such as real-time inventory tracking, order management, and reporting capabilities.	CO1, CO2, CO6
14	**Project** Build a document management system that leverages cloud storage for storing and organizing documents. Include features like full-text search, document tagging, and access control for secure document sharing.	CO1, CO2, CO5s

K.R. MANGALAM UNIVERSITY

APPLICATION DEVELOPMENT AND DEVOPS ON CLOUD

Department:	**Department of Computer Science and Engineering**		
Course Name: **APPLICATION DEVELOPMENT AND DEVOPS ON CLOUD**	**Course Code**	**L-T-P**	**Credits**
	ENSP407	4-0-0	4
Type of Course:	Minor (Department Elective III)		
Pre-requisite(s), if any: Nil			
Brief Syllabus: The syllabus aims to equip students with practical skills and theoretical knowledge to design, develop, and deploy applications in cloud environments while implementing DevOps practices to enhance software development, delivery, and operations on the cloud. It prepares them for a career in the dynamic and rapidly growing field of cloud computing and DevOps, where demand for skilled professionals is high due to the increasing adoption of cloud technologies in various industries.			
UNIT WISE DETAILS			
Unit Number: 1	**Title: Introduction to Cloud Computing**	**No. of hours: 10**	
Content Summary: Overview of Cloud Computing: Definition, Characteristics, Advantages, and Disadvantages. Cloud Service Models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS). Cloud Deployment Models: Public Cloud, Private Cloud, Hybrid Cloud, Community Cloud. Cloud Providers: Introduction to popular cloud service providers (e.g., AWS, Azure, Google Cloud).			

Cloud Architecture: Understanding cloud components, virtualization, and multi-tenancy.

Introduction to DevOps: Principles, Practices, and Benefits.

Unit Number: 2	Title: Cloud-Based Application Development	No. of hours: 10

Content Summary:

Cloud Native Applications: Principles and architecture for building applications optimized for cloud environments.

Containerization: Docker and Kubernetes for application deployment, scaling, and management.

Microservices Architecture: Designing, developing, and deploying microservices-based applications.

Serverless Computing: Introduction to serverless architecture and Function as a Service (FaaS).

Application Security in the Cloud: Best practices for securing cloud-based applications.

Cloud-Based Databases: Database options and considerations in cloud environments.

Unit Number: 3	Title: DevOps Practices in Cloud	No. of hours: 10

Content Summary:

Continuous Integration (CI) and Continuous Deployment (CD) in Cloud: Setting up CI/CD pipelines.

Infrastructure as Code (IaC): Automating infrastructure provisioning using tools like Terraform and CloudFormation.

Configuration Management: Managing application configurations in a cloud-based environment.

Monitoring and Logging in Cloud: Tools and techniques for monitoring application performance and gathering logs.

Auto-scaling and Load Balancing: Scaling applications dynamically based on demand.

High Availability and Disaster Recovery: Designing and implementing resilient applications in the cloud.

Unit Number: 4	Title: Cloud-Based DevOps Tools and Best Practices	No. of hours: 10

Content Summary:

Cloud-Based Version Control: Using Git and other version control tools in cloud-based development.

Collaboration and Communication Tools: Utilizing cloud-based collaboration tools for distributed teams.

Cloud-Based Testing and Quality Assurance: Strategies for testing applications in cloud environments.

Cost Management in Cloud: Optimizing cloud resource usage and cost control.

Performance Optimization in Cloud: Techniques for improving application performance in cloud environments.

Case Studies and Real-World Projects: Analyzing successful cloud-based application development and DevOps projects.

***SELF-LEARNING COMPONENTS:**

- https://elearn.nptel.ac.in/shop/iit-workshops/completed/azure-devops-and-micro-services-azure-kubernetes-deployment-models/
- https://nptel.ac.in/courses/106105167

Please Note:

1)Students are supposed to learn the components on self-basis

2) At least 5-10 % syllabus will be asked in end term exams from self-learning components.

Reference Books:

1. Jez Humble and David Farley, Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation, Pearson Education, Inc., 2011.
2. Thomas Erl, Ricardo Puttini, and Zaigham Mahmood, Cloud Computing: Concepts, Technology & Architecture, Prentice Hall, 2013.
3. Arun Eapen, Docker on Amazon Web Services: Build, deploy, and manage

your container applications at scale on AWS, Packt Publishing, 2017.
4. Sam Newman, Building Microservices: Designing Fine-Grained Systems, O'Reilly Media, Inc., 2015.
5. Mark Richards and Neal Ford, Fundamentals of Software Architecture: An Engineering Approach, O'Reilly Media, Inc., 2020.

Define Course Outcomes (CO)

COs	Statements
CO1	**Understand** the fundamental concepts and principles of cloud computing, including different cloud service models and deployment models. Gain insights into various cloud providers and their offerings to make informed decisions for application development and deployment.
CO2	**Acquire** practical knowledge and hands-on experience in developing cloud-native applications, utilizing containerization with Docker and orchestration using Kubernetes. Demonstrate proficiency in designing and implementing microservices-based architectures.
CO3	**Master** the principles and practices of DevOps in a cloud environment. Learn how to set up Continuous Integration (CI) and Continuous Deployment (CD) pipelines, automate infrastructure provisioning with Infrastructure as Code (IaC), and manage application configurations effectively.
CO4	**Develop** the skills to monitor, log, and optimize the performance of cloud-based applications. Explore auto-scaling and load balancing techniques to ensure high availability and disaster recovery strategies for resilient applications.
CO5	**Apply** cloud-based DevOps tools and best practices to collaborate effectively in distributed teams. Gain knowledge of cloud-based version control, testing, cost management, and performance optimization techniques to deliver efficient and cost-effective cloud solutions.

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels© 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C2	-	P1
C02	C3	-	P2
CO3	C3	-	P3
CO4	C1	-	-
CO5	C1	-	P1

CO-PO Mapping

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	2	-	2	1	-	3	3	-	1	3
CO2	3	3	2	-	3	-	-	-	-	-	-	3
CO3	3	3	2	2	3	1	-	-	-	-	-	3
CO4	-	3	2	2	3	-	-	-	-	-	1	3
CO5	-	3	2	3	2	-	-	-	-	-	-	3

Justification for mapping must be relevant.

1=weakly mapped
2= moderately mapped
3=strongly mapped

CO-PSO Mapping

PO	PSO1	PSO2	PSO3	PSO4
CO1	3	1	2	-
CO2	2	1	-	-
CO3	2	-	-	1
CO4	-	-	1	-
CO5	2	1	-	1

Relevance of the Syllabus to various indicators

Unit I	Introduction
Local	Addresses local understanding of the Internet and its impact on society
Regional	Addresses regional internet connectivity and network infrastructure requirements
National	Contributes to national digital literacy and internet connectivity strategies
Global	Aligns with global trends in internet technologies and network protocols
Employability	Develops skills in using internet-based services and understanding network protocols
Entrepreneurship	-
Skill Development	Develops basic knowledge and skills in internet technologies and network protocols
Professional Ethics	-
Gender	-
Human Values	-

Environment & Sustainability	-
Unit II	
Local	Addresses local understanding of the Internet and its impact on society
Regional	-
National	Contributes to national digital literacy and internet connectivity strategies
Global	Aligns with global trends in internet technologies and network protocols
Employability	Develops skills in using internet-based services and understanding network protocols
Entrepreneurship	-
Skill Development	Develops basic knowledge and skills in internet technologies and network protocols
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	
Local	Addresses local network security needs and practices
Regional	-
National	Contributes to national network security strategies and protocols
Global	Aligns with global trends in network security techniques and protocols

Employability	Develops skills in network programming and network security techniques
Entrepreneurship	-
Skill Development	Develops knowledge and skills in client-server programming and network security
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	
Local	Addresses local understanding and implementation of internet-based services
Regional	-
National	Contributes to national digital communication strategies and multimedia applications
Global	Aligns with global trends in internet telephony, multimedia applications, and SEO
Employability	Develops skills in internet telephony, multimedia applications, and SEO
Entrepreneurship	-
Skill Development	Develops knowledge and skills in internet telephony, multimedia applications, and SEO
Professional Ethics	-
Gender	-
Human Values	-

Environment & Sustainability	-
SDG	SDG 4
NEP 2020	-
POE/4th IR	Aligns with the concepts of internet telephony, multimedia applications, and SEO

K.R. MANGALAM UNIVERSITY

APPLICATION DEVELOPMENT AND DEVOPS ON CLOUD LAB

Department:	**Department of Computer Science and Engineering**		
Course Name: **APPLICATION DEVELOPMENT AND DEVOPS ON CLOUD LAB**	**Course Code**	**L-T-P**	**Credits**
	ENSP457	0-0-2	1
Type of Course:	Minor (Department Elective III)		
Pre-requisite(s), if any: Nil			

Proposed Lab Experiments

Defined Course Outcomes

COs	**Course Outcomes (COs)**
CO 1	**Understand** the fundamental concepts of cloud computing, cloud service models, and cloud-based application development
CO 2	**Acquire** hands-on experience in deploying and managing applications on cloud platforms using various cloud services and tools
CO 3	**Implement** DevOps practices in cloud-based application development, including continuous integration, continuous deployment, and version control
CO 4	**Demonstrate** proficiency in using containerization (Docker and Kubernetes), Infrastructure as Code (IaC), and serverless computing for cloud-based application deployment and scaling

CO 5	**Develop** skills in application monitoring, logging, security, and performance optimization for cloud-based applications, and apply cost optimization strategies for cloud resources

Experiment No.	**Experiment Title**	**Mapped COs**
1	Setting up a cloud-based development environment	CO1
2	Deploying a web application on a cloud platform	CO1, CO2
3	Containerizing and deploying with Docker and Kubernetes	CO2, CO4
4	Implementing Continuous Integration (CI)	CO2, CO3
5	Implementing Infrastructure as Code (IaC)	CO2, CO4
6	Configuring auto-scaling for an application	CO2
7	Implementing high availability and disaster recovery	CO2
8	Building a serverless application	CO2, CO4
9	Implementing application monitoring and logging	CO2
10	Creating and managing a cloud-based database	CO2
11	Integrating third-party APIs with an application	CO2
12	Implementing security best practices	CO2
13	Managing application configurations	CO2

14	Performing load testing on an application	CO2
15	Migrating an application to the cloud	CO2
16	Implementing DevOps practices	CO3
17	Deploying a multi-tier application architecture	CO1, CO2
18	Integrating CI/CD pipeline with version control	CO2, CO3
19	Implementing blue-green deployment	CO3
20	Implementing Canary deployment	CO3
21	Performance optimization in cloud	CO2, CO4
22	Implementing feature flags	CO2, CO3
23	Implementing cost optimization strategies	CO2, CO5
24	Deploying a microservices-based application	CO2, CO4
25	Real-world project showcasing cloud-based development	CO1, CO2, CO3, CO4, CO5

1. **Exercise for Setting up a cloud-based development environment**: Exercise: Students set up an account with a cloud service provider (e.g., AWS, Azure) and create a virtual machine instance to host a basic web application.

2. **Project for Deploying a web application on a cloud platform**: Project: Students develop a simple web application and deploy it on a cloud platform using PaaS. They configure auto-scaling based on application demand and perform load testing to observe scaling behavior.

3. **Exercise for Containerizing and deploying with Docker and Kubernetes**: Exercise: Students containerize an existing application using Docker and deploy it on a Kubernetes cluster. They configure Kubernetes services, pods, and replicas for the application.

4. **Exercise for Implementing Continuous Integration (CI)**: Exercise: Students set up a CI/CD pipeline using a cloud-based CI/CD tool (e.g.,

Jenkins) to automatically build and test their application whenever code changes are pushed to the repository.

5. **Exercise for Implementing Infrastructure as Code (IaC)**: Exercise: Students use Terraform or CloudFormation to define and provision cloud resources for their application, such as virtual machines, storage, and networking.

6. **Project for Configuring auto-scaling for an application**: Project: Students design a scalable architecture for a multi-tier application and implement auto-scaling based on CPU utilization. They monitor application performance during auto-scaling events.

7. **Exercise for Implementing high availability and disaster recovery**: Exercise: Students configure a highly available architecture using load balancers and multiple availability zones to ensure application resilience. They test disaster recovery scenarios.

8. **Project for Building a serverless application**: Project: Students develop a serverless application using FaaS. They implement serverless functions to handle specific application features and integrate them with event triggers.

9. **Exercise for Implementing application monitoring and logging**: Exercise: Students set up monitoring and logging solutions (e.g., CloudWatch, Stackdriver) for their applications to collect metrics, logs, and perform analysis.

10. **Project for Creating and managing a cloud-based database**: Project: Students design a database schema and implement a cloud-based database (e.g., Amazon RDS, Azure SQL Database). They perform data migration and backup strategies.

11. **Project for Integrating third-party APIs with an application**: Project: Students integrate a popular third-party API (e.g., Google Maps API, Twitter API) into their application to enhance its functionality and showcase real-time data retrieval.

12. **Exercise for Implementing security best practices**: Exercise: Students configure Identity and Access Management (IAM) roles and policies to enforce security for their cloud-based applications. They implement Network Security Groups and security groups to control traffic flow.

13. **Exercise for Managing application configurations**: Exercise: Students use cloud-native configuration management tools (e.g., AWS Systems

Manager, Azure Configuration Management) to manage application configurations across multiple instances.

14. **Project for Performing load testing on an application**: Project: Students use load testing tools (e.g., JMeter, Gatling) to simulate high user loads on their cloud-based applications and analyze performance metrics under heavy traffic.

15. **Project for Migrating an application to the cloud**: Project: Students migrate an existing on-premises application to a cloud platform, considering factors like data transfer, application dependencies, and post-migration testing.

16. **Project for Implementing DevOps practices**: Project: Students collaborate in teams to develop a complete cloud-based application and implement DevOps practices, including version control, CI/CD, and automated testing.

17. **Exercise for Deploying a multi-tier application architecture**: Exercise: Students design and deploy a multi-tier application on the cloud, comprising frontend web servers, application servers, and database servers.

18. **Project for Integrating CI/CD pipeline with version control and issue tracking tools**: Project: Students set up integration between a CI/CD pipeline and version control (e.g., Git) and issue tracking tools (e.g., Jira) for automated build and issue tracking.

19. **Exercise for Implementing blue-green deployment**: Exercise: Students implement blue-green deployment strategies to minimize downtime during application updates, allowing seamless rollback if issues arise.

20. **Exercise for Implementing Canary deployment**: Exercise: Students deploy a new feature of their application to a small subset of users using Canary deployment to gather feedback and monitor performance.

21. **Project for Performance optimization in the cloud**: Project: Students optimize the performance of their cloud-based application by tuning parameters, caching, and optimizing database queries for better response times.

22. **Project for Implementing feature flags**: Project: Students implement feature flags (feature toggles) in their application to enable/disable features dynamically and gradually release new functionalities.

23. **Project for Implementing cost optimization strategies**: Project: Students analyze the cost of their cloud resources and implement cost optimization techniques, such as using reserved instances and rightsizing resources.

24. **Project for Deploying a microservices-based application**: Project: Students design and deploy a microservices-based application architecture on the cloud, using containers and orchestrators like Kubernetes.

25. **Project for Real-world project showcasing cloud-based development**: Project: Students work on a real-world cloud-based development project, applying the concepts learned throughout the course to design, develop, deploy, and manage an application in the cloud.

K.R. MANGALAM UNIVERSITY

(DEPARTMENT ELECTIVE-IV)

MOBILE APPLICATION DEVELOPMENT USING IOS

<table>
<tr><td>Department:</td><td colspan="3">Department of Computer Science and Engineering</td></tr>
<tr><td rowspan="2">Course Name:

Mobile Application Development using iOS</td><td>Course Code</td><td>L-T-P</td><td>Credits</td></tr>
<tr><td>ENSP409</td><td>4-0-0</td><td>4</td></tr>
<tr><td>Type of Course:</td><td colspan="3">Minor (Department Elective IV)</td></tr>
<tr><td colspan="4">Pre-requisite(s), if any: Basics of Android</td></tr>
<tr><td colspan="4">Brief Syllabus:

The objective of the course is to provide skills to develop applications for OS X and iOS. It includes introduction to development framework Xcode. Objective-C is used as programming language to develop the applications. Objective-C is the superset of the C programming language and provides object-oriented capabilities and a dynamic runtime. Objective-C inherits the syntax, primitive types, and flow control statements of C and adds syntax for defining classes and methods.</td></tr>
<tr><td colspan="4">UNIT WISE DETAILS</td></tr>
<tr><td>Unit Number: 1</td><td colspan="2">Title: Introduction to IDE and SDK of iOS App Development</td><td>No. of hours: 10</td></tr>
<tr><td colspan="4">Content Summary:

Xcode-The SDK environment, Supporting tools, Advance settings. Development Technique, Fundamental of Object-Oriented Programming, The MVC architecture.</td></tr>
</table>

Unit Number: 2	Title: Objective-C	No. of hours: 10

Content Summary:

Introduction to Objective C, Primitive Data Types, Conditions, Loops, Functions, Arrays, Pointers, Structures, Classes, Objects, Foundation, Memory Management, Inheritance, Categories, Protocols, Predicates, Blocks, Multi-Threading.

Objects Send and Receive Messages concept, Use of Pointers to Keep Track of Objects, Methods - Return Values.

Unit Number: 3	Title: Encapsulating Data	No. of hours: 10

Content Summary:

Properties of Encapsulation of an Object's Values, Declare Public Properties for Exposed Data, Use Accessor Methods to Get or Set Property Values, Concept of Dot Syntax, Properties Are Backed by Instance Variables.

Dealing with Errors: Use NSError for Most Errors, Some Delegate Methods Alert You to Errors, Some Methods Pass Errors by Reference

Unit Number: 4	Title: Developing iOS Applications	No. of hours: 10

Content Summary:

iOS App Anatomy, Design Principles, Creating a Basic Hello World App with interface elements, UI View & Controller, UI Elements, Trigger Actions, Storyboard, Device Orientations, Using Gestures, Popovers and Modal Dialogs, Creating Universal Apps, Status Bar, Navigation Bar, Tab Bar, Content Views (e.g. Image view, Map View etc.), UI Table View and Table View Controller, Core Data, Test your App, Publishing your App.

***Self-Learning Components:**

1. XCode Documentation

References:

1. https://www.tutorialspoint.com/objective_c/objective_c_quick_guide.htm

2. https://www.coursera.org/learn/introduction-to-ios-mobile-application-development
3. https://www.geeksforgeeks.org/classes-objects-in-objective-c/

Please Note:

1)Students are supposed to learn the components on self-basis

2) At least 5-10 % syllabus will be asked in end term exams from self-learning components.

Textbook:

1. Effective objective C 2.0, Matt Galloway, Effective software development series, Scott Meyers.

Reference Books:

1. Programming in Objective-C (5th Edition) (Developer's Library) by Stephen G. Kochan.

2. iOS 6 Development Unleashed: Developing Mobile Applications for Apple iPhone, iPad, and iPod Touch by Robert McGovern

Online References:

1. https://developer.apple.com/library/archive/documentation/Cocoa/Conceptual/ProgrammingWithObjectiveC/Introduction/Introduction.html
2. https://www.digitalocean.com/community/tutorials/objective-c-hello-world-tutorial

Course Outcomes (CO)

COs	Statements
CO1	**Create** iPhone apps using Objective-C and Apple's new programming language, use industry tools and frameworks such as Cocoa, Xcode, UIKit, Git.
CO2	**Understand** and know how to use properly UIKit, asynchronous code, Core Image, NSURL Session and JSON Map Kit and Core Location, Auto Layout, Source Control, Core Data, Animation, and the app submission process.

CO3	Read and write programs based on Objective-C, also have a strong grasp of Objective-C objects
CO4	**Organize** their code professionally using objects and blocks, prototype several entries- level apps and try to publish on App store.

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels(C) 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C2	A1	P1
C02	C3	A4	P1
CO3	C5	A2	P2
CO4	C6	A1	P4

CO-PO Mapping

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	1	2	-	-	-	-	-	-	-	-	3
CO2	-	3	2	-	2	-	-	-	-	-	-	3
CO3	-	2	2	3	1	-	-	-	-	-	-	3
CO4	-	2	3	-	-		-	-	-	-	-	2

1=weakly mapped
2= moderately mapped
3=strongly mapped

CO-PSO Mapping

PO	PSO1	PSO2	PSO3	PSO4
CO1	2		-	3
CO2	2	2	-	2
CO3	-	2	-	-
CO4	-	-	-	3

Relevance of the Syllabus to various indicators

Unit I	Introduction to IDE and SDK of iOS App Development
Local	-
Regional	-
National	-
Global	Xcode is the official IDE provided by Apple for iOS app development. It is available globally and widely used by developers worldwide. Xcode includes a suite of tools, such as Interface Builder, Instruments, and iOS Simulator, along with an extensive SDK for building iOS apps.
Employability	-
Entrepreneurship	-
Skill Development	-
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-

Unit II	Objective-C
Local	-
Regional	-
National	
Global	Objective C can be used globally with its syntax and syntactic rules
Employability	-
Entrepreneurship	-
Skill Development	-
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	Encapsulating Data
Local	In programming, "local" usually refers to variables, data, or methods that are confined to a specific scope, such as within a function or a block. Local variables are only accessible within the block or function where they are declared.
Regional	-
National	-
Global	
Employability	
Entrepreneurship	-
Skill	-

Development	
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	Developing iOS Applications
Local	-
Regional	-
National	"National" might represent initiatives or policies related to iOS app development adopted or regulated at the national level. For example, it could include national-level educational programs or government-supported initiatives promoting digital skills and app development.
Global	
Employability	-
Entrepreneurship	-
Skill Development	-
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
SDG	SDG 4, SDG 8, SDG 9
NEP 2020	Promoting universal access to education, holistic development, multidisciplinary approach, skill development, critical thinking,

	creativity, ICT integration, research and development, global competencies, and professional ethics.
POE/4th IR	Skill Development / Coding / Programming Software

K.R. MANGALAM UNIVERSITY

MOBILE APPLICATION DEVELOPMENT USING IOS LAB

Department:	**Department of Computer Science and Engineering**		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Mobile Application Development using iOS Lab	**ENSP459**	0-0-2	1
Type of Course:	Minor (Department Elective IV)		
Pre-requisite(s), if any: Basics of Android			

Lab Experiments

Defined Course Outcomes

COs	
CO 1	Create iPhone apps using Objective-C and Apple's new programming language, use industry tools and frameworks such as Cocoa, Xcode, UIKit, Git.
CO 2	Understand and know how to use properly UIKit, asynchronous code, Core Image, NSURL Session and JSON Map Kit and Core Location, Auto Layout, Source Control, Core Data, Animation, and the app submission process.
CO 3	Read and write programs based on Objective-C, also have a strong grasp of Objective-C objects
CO 4	Organize their code professionally using objects and blocks, prototype several entry- level apps and try to publish on App store.

Ex. No.	**Experiment Title**	**Mapped CO/COs**

1	Case Study of Objective-C language.	CO2
2	Case study of Windows and MAC systems	CO2
3	Case Study of XCode based on MAC Systems	CO2
4	Design an App for UISwitch based on Objective-C language	CO1
5	Design an App for UISlider based on Objective-C language	CO1
6	Design an App for UIStepper based on Objective-C language	CO1
7	Write a program for creating Story Boards	CO1
8	Design an App for UIAnimation based on Objective-C language	CO1
9	Create a Simple Calculator using Objective-C Language	CO1
10	Design an App for UIProgress Bar based on Objective-C language	CO1
11	Design an App for UIDatePicker Bar based on Objective-C language	CO1
12	Write an Objective-C program to print factorial of a given number	CO3
13	Write an Objective-C program to print Fibonacci series	CO3
14	Write an Objective-C program that displays the Phrase "Hello World"	CO3
15	Write an Objective-C program for displaying the value of variables	CO3
16	Write an Objective-C program for displaying the sum and subtraction of two variables	CO3
17	Write an Objective-C program for displaying the multiplication and division of the two variables	CO3

18	Write an Objective-C program that demonstrate control structure of Objective-C language	CO3
19	Create a Button using Objective-C	CO3
20	Write an Objective-C program to print the value of a variable inside a text, place it in parentheses, and insert a backslash just prior to the opening parenthesis.	CO3
21	Write an Objective-C program to print Floyd's Triangle.	CO3
22	Write an Objective-C program to print palindrome of a number.	CO3
23	Write an Objective-C program to print pyramid.	CO3
24	Write an Objective-C program to find greatest number in between three numbers	CO3
25	Write an Objective-C program to check whether a number is even or odd.	CO3
	Mini Project 1: Make an interactive project based on iOS App using Objective-C Language	CO4
	Mini Project 2: Upload your iOS App in Apple AppStore and Publish it	CO4

K.R. MANGALAM UNIVERSITY

DEVOPS & AUTOMATION

Department:	**Department of Computer Science and Engineering**		
Course Name: **DevOps & Automation**	**Course Code**	**L-T-P**	**Credits**
	ENSP411	4-0-0	4
Type of Course:	Minor (Department Elective IV)		
Pre-requisite(s), if any: Nil			
Brief Syllabus: Throughout the subject, students will engage in hands-on exercises and projects to gain practical experience with various DevOps tools and practices. By the end of the course, students will be well-equipped to embrace the DevOps culture and apply automation techniques to enhance software development, delivery, and operations processes.			
UNIT WISE DETAILS			
Unit Number: 1	**Title: Introduction to DevOps**		**No. of hours: 10**
Content Summary: Overview of DevOps: Definition, objectives, and benefits. DevOps Principles: Collaboration, automation, continuous integration, continuous delivery, and continuous deployment. DevOps Tools: Introduction to popular DevOps tools like Git, Jenkins, Docker, Kubernetes, and Ansible. Version Control with Git: Branching, merging, and collaborative development using Git. Continuous Integration (CI): Setting up CI pipelines with Jenkins for automated building and testing. Continuous Delivery and Deployment: Implementing CD pipelines for deploying			

applications to various environments.

Unit Number: 2	Title: Containerization and Orchestration	No. of hours: 10

Content Summary:

Introduction to Containers: Docker and containerization concepts.

Container Management: Working with Docker containers, images, and registries.

Docker Compose: Managing multi-container applications.

Introduction to Kubernetes: Container orchestration and Kubernetes architecture.

Deploying Applications with Kubernetes: Deploying, scaling, and managing applications on Kubernetes.

Unit Number: 3	Title: Configuration Management and Infrastructure as Code (IaC)	No. of hours: 10

Content Summary:

Introduction to Configuration Management: Need for configuration management tools.

Managing Infrastructure with Ansible: Ansible architecture and playbooks for automated configuration management.

Infrastructure as Code (IaC) Concepts: Managing infrastructure using code, benefits of IaC.

IaC with Terraform: Infrastructure provisioning using Terraform and cloud service providers (e.g., AWS, Azure).

Unit Number: 4	Title: Monitoring, Logging, and Security in DevOps	No. of hours: 10

Content Summary:

Application Monitoring: Monitoring tools and techniques for tracking application performance and health.

Log Management: Centralized log collection, analysis, and visualization.

Security in DevOps: Implementing security best practices in CI/CD pipelines and containerized environments.

DevOps Culture and Collaboration: Encouraging collaboration between development and operations teams.
***SELF-LEARNING COMPONENTS:** https://elearn.nptel.ac.in/shop/iit-workshops/completed/cicd-devops-automation-and-devsecops-automation/ **Please Note:** 1)Students are supposed to learn the components on self-basis 2) At least 5-10 % syllabus will be asked in end term exams from self-learning components.
Reference Books: 1. Jez Humble and David Farley, "Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation," Pearson Education, Inc., 2011. 2. Nigel Poulton, "The Kubernetes Book," Independently published, 2018. 3. Sam Newman, "Building Microservices: Designing Fine-Grained Systems," O'Reilly Media, Inc., 2015. 4. Eberhard Wolff, "Microservices Patterns: With examples in Java," Manning Publications, 2018. 5. Yevgeniy Brikman, "Terraform: Up & Running: Writing Infrastructure as Code," O'Reilly Media, Inc., 2017.

Define Course Outcomes (CO)

COs	Statements
CO1	**Understand** the principles and benefits of DevOps, and its role in enhancing collaboration and efficiency between development and operations teams.
CO2	**Acquire** hands-on experience with popular DevOps tools such as Git, Jenkins, Docker, Kubernetes, and Ansible for implementing continuous integration, continuous delivery, and automated deployment processes.

CO3	**Demonstrate** proficiency in containerization and orchestration techniques using Docker and Kubernetes for efficient and scalable application deployment and management.
CO4	**Implement** configuration management and Infrastructure as Code (IaC) using Ansible and Terraform to automate the provisioning and management of infrastructure resources.
CO5	**Develop** skills in monitoring, logging, and security practices in the context of DevOps, ensuring application performance, resilience, and adherence to security best practices.

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels© 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	-	-	P1
C02	C2	-	P2
CO3	C3	L3	P3
CO4	C4	-	-
CO5	C5	L5	P5

CO-PO Mapping

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	2	2	-	3	-	3	-	-	-	3
CO2	3	3	-	3	3	-	-	3	-	-	-	3
CO3	3	3	2	2	3	2	2	3	-	-	-	3
CO4	-	3	2	3	3	-	2	3	2	-	-	3
CO5	-	3	2	3	-	-	3	3	-	-	-	3

Justification for mapping must be relevant.

1=weakly mapped
2= moderately mapped
3=strongly mapped

CO-PSO Mapping

PO	PO1	PO2	PO3	PSO4
CO1	3	-	1	-
CO2	2	1	-	1
CO3	3	-	-	-
CO4	1	1	1	1
CO5	-	2	-	-

Relevance of the Syllabus to various indicators

Unit I	Introduction
Local	Addresses local understanding of the Internet and its impact on society
Regional	Addresses regional internet connectivity and network infrastructure requirements
National	Contributes to national digital literacy and internet connectivity strategies
Global	Aligns with global trends in internet technologies and network

	protocols
Employability	Develops skills in using internet-based services and understanding network protocols
Entrepreneurship	-
Skill Development	Develops basic knowledge and skills in internet technologies and network protocols
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit II	
Local	Addresses local understanding of the Internet and its impact on society
Regional	-
National	Contributes to national digital literacy and internet connectivity strategies
Global	Aligns with global trends in internet technologies and network protocols
Employability	Develops skills in using internet-based services and understanding network protocols
Entrepreneurship	-
Skill Development	Develops basic knowledge and skills in internet technologies and network protocols
Professional Ethics	-
Gender	-

Human Values	-
Environment & Sustainability	-
Unit III	
Local	Addresses local network security needs and practices
Regional	-
National	Contributes to national network security strategies and protocols
Global	Aligns with global trends in network security techniques and protocols
Employability	Develops skills in network programming and network security techniques
Entrepreneurship	-
Skill Development	Develops knowledge and skills in client-server programming and network security
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	
Local	Addresses local understanding and implementation of internet-based services
Regional	-
National	Contributes to national digital communication strategies and multimedia applications
Global	Aligns with global trends in internet telephony, multimedia

	applications, and SEO
Employability	Develops skills in internet telephony, multimedia applications, and SEO
Entrepreneurship	-
Skill Development	Develops knowledge and skills in internet telephony, multimedia applications, and SEO
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
SDG	SDG 4
NEP 2020	-
POE/4th IR	Aligns with the concepts of internet telephony, multimedia applications, and SEO

K.R. MANGALAM UNIVERSITY

DEVOPS & AUTOMATION LAB

Department:	**Department of Computer Science and Engineering**		
Course Name: **DevOps & Automation** Lab	**Course Code**	**L-T-P**	**Credits**
	ENSP461	0-0-2	1
Type of Course:	Minor (Department Elective IV)		
Pre-requisite(s), if any:			

Proposed Lab Experiments

Defined Course Outcomes

COs	**Course Outcomes (COs)**
CO 1	**Gain** hands-on experience in setting up version control using Git and performing collaborative software development with branching and merging techniques.
CO 2	**Acquire** practical knowledge in implementing continuous integration and continuous deployment (CI/CD) pipelines using Jenkins, automating the build, test, and deployment processes.
CO 3	**Develop** proficiency in containerization with Docker, including managing Docker containers and images, and deploying applications on Kubernetes for efficient and scalable orchestration.
CO 4	**Demonstrate** skills in infrastructure automation and configuration management using Ansible and Terraform to provision and manage cloud resources and application configurations.
CO 5	**Understand** and **apply** monitoring, logging, and security practices in DevOps, ensuring application performance, resilience, and adherence to security best practices throughout the software development lifecycle.

Ex. No.	Experiment Title	Mapped CO(s)
1	Setting up version control with Git	CO1
2	Implementing a basic Jenkins CI/CD pipeline	CO2
3	Automating application deployment with Jenkins	CO2
4	Containerizing an application using Docker	CO3
5	Managing Docker containers and images	CO3
6	Deploying applications with Kubernetes	CO3
7	Implementing Kubernetes deployment strategies	CO3
8	Continuous deployment with Kubernetes	CO3
9	Configuring infrastructure with Ansible	CO4
10	Automating application configuration with Ansible	CO4
11	Implementing Infrastructure as Code (IaC) with Terraform	CO4
12	Creating scalable and resilient infrastructure with Terraform	CO4
13	Monitoring application performance with Prometheus	CO5
14	Logging and centralized log management	CO5
15	Implementing security measures in CI/CD pipelines	CO5
16	Implementing feature flags for controlled feature rollout	CO5
17	Load testing and performance optimization	CO5
18	Automating application tests with Selenium	CO2, CO5
19	Integrating automated testing in CI/CD pipelines	CO2, CO5

20	Blue-green deployment for zero-downtime updates	CO3, CO5
21	Canary deployment for testing new features	CO3, CO5
22	Implementing GitOps for application deployments	CO3, CO5
23	Managing secrets and sensitive data securely	CO5
24	Disaster recovery planning and testing	CO5
25	Creating a DevOps project integrating multiple tools	CO1, CO2, CO3, CO4, CO5

1. **Setting up version control with Git**: Exercise: Initialize a Git repository, create branches, perform commits, and push changes to a remote repository. Project: Collaboratively work on a project using branching and merging techniques in Git.
2. **Implementing a basic Jenkins CI/CD pipeline**: Exercise: Set up a simple Jenkins pipeline to build and test a sample application from version control. Project: Develop a complete CI/CD pipeline that includes code building, automated testing, and deployment to a staging environment.
3. **Automating application deployment with Jenkins**: Exercise: Configure Jenkins to automatically deploy the application to a test server upon successful build. Project: Implement a full-fledged CD pipeline with Jenkins, including deployment to production after successful testing.
4. **Containerizing an application using Docker**: Exercise: Dockerize a basic application and run it in a container. Project: Containerize a multi-service application with Docker Compose for easier deployment.
5. **Managing Docker containers and images**: Exercise: Explore Docker commands to manage containers and images, such as starting, stopping, and cleaning up. Project: Implement a container registry and manage images for different application versions.
6. **Deploying applications with Kubernetes**: Exercise: Set up a Kubernetes cluster and deploy a basic application using YAML manifests. Project: Deploy

a microservices-based application with Kubernetes, configuring services and network policies.

7. **Implementing Kubernetes deployment strategies**: Exercise: Implement rolling updates and rollbacks in Kubernetes. Project: Use Kubernetes deployment strategies like blue-green and canary deployments for a real-world application.

8. **Continuous deployment with Kubernetes**: Exercise: Set up a Jenkins pipeline for continuous deployment to Kubernetes. Project: Create an end-to-end automated CD pipeline with Jenkins and Kubernetes.

9. **Configuring infrastructure with Ansible**: Exercise: Use Ansible to provision and configure virtual machines. Project: Create a playbook to configure a complete development environment for an application.

10. **Automating application configuration with Ansible**: Exercise: Create Ansible playbooks to automate application-specific configurations. Project: Implement dynamic inventory and use Ansible roles for better code organization.

11. **Implementing Infrastructure as Code (IaC) with Terraform**: Exercise: Set up a basic Terraform configuration to create cloud resources. Project: Use Terraform to define infrastructure for a scalable and fault-tolerant application.

12. **Creating scalable and resilient infrastructure with Terraform**: Exercise: Implement auto-scaling and load balancing in Terraform. Project: Design a Terraform template for a highly available architecture using multiple availability zones.

13. **Monitoring application performance with Prometheus**: Exercise: Set up Prometheus for monitoring application metrics. Project: Create custom Prometheus metrics and use Grafana for visualization and alerting.

14. **Logging and centralized log management**: Exercise: Configure centralized log collection using tools like Fluentd or Logstash. Project: Set up ELK (Elasticsearch, Logstash, and Kibana) stack for efficient log analysis.

15. **Implementing security measures in CI/CD pipelines**: Exercise: Use Jenkins plugins to implement security checks in CI/CD pipelines. Project: Implement security scanning tools like SonarQube and integrate them into the pipeline.

16. **Implementing feature flags for controlled feature rollout**: Exercise: Add feature flags to a sample application to enable/disable specific features. Project: Implement a feature flag service for a real-world application and manage feature rollout.

17. **Load testing and performance optimization**: Exercise: Use load testing tools to evaluate application performance under heavy traffic. Project: Analyze performance bottlenecks and optimize the application for scalability.

18. **Automating application tests with Selenium**: Exercise: Use Selenium WebDriver for automating browser-based tests. Project: Develop an automated testing suite covering multiple application features.

19. **Integrating automated testing in CI/CD pipelines**: Exercise: Integrate automated tests into the Jenkins CI/CD pipeline. Project: Implement a complete testing strategy, including unit, integration, and end-to-end tests.

20. **Blue-green deployment for zero-downtime updates**: Exercise: Perform blue-green deployment for a sample application update. Project: Set up a blue-green deployment strategy for a production application.

21. **Canary deployment for testing new features**: Exercise: Implement canary deployment for a specific application feature. Project: Use canary deployment to gradually release new features to a subset of users.

22. **Implementing GitOps for application deployments**: Exercise: Use GitOps principles to manage Kubernetes manifests with Git. Project: Implement a GitOps workflow for application deployment and configuration management.

23. **Managing secrets and sensitive data securely**: Exercise: Utilize Kubernetes secrets or HashiCorp Vault to manage sensitive data. Project: Set up a secure secret management system for a production environment.

24. **Disaster recovery planning and testing**: Exercise: Design a disaster recovery plan for a sample application. Project: Test the disaster recovery plan and validate its effectiveness.

25. **Creating a DevOps project integrating multiple tools**: Exercise: Choose and integrate various DevOps tools into a sample project. Project: Create an end-to-end DevOps project showcasing the integration of tools and best practices.

.NET FRAMEWORK

Department:	**Department of Computer Science and Engineering**		
Course Name: **.NET Framework**	**Course Code**	**L-T-P**	**Credits**
	ENSP413	4-0-0	4
Type of Course:	Minor (Department Elective IV)		
Pre-requisite(s), if any:			
Brief Syllabus: The ".NET Framework" syllabus covers introduction and components of .NET, programming languages, Visual Studio, OOP, exception handling, memory management, Windows Forms/WPF, ASP.NET, web services, .NET Core, Entity Framework, and WCF. Emphasis on practical application and development skills for building robust and secure applications.			
UNIT WISE DETAILS			
Unit Number: 1	**Title: Introduction to .NET Framework**	**No. of hours: 8**	
Content Summary: Overview of .NET Framework ,Introduction to the .NET platform, Evolution and history of .NET Framework, Key components and architecture of .NET Framework, Common Language Runtime (CLR) and Just-In-Time (JIT) compilation, Common Intermediate Language (CIL) and Intermediate Language (IL), Programming Languages in .NET (C# as the primary language for .NET development & Visual Basic .NET) ,Introduction to Visual Studio IDE, Installation and configuration of .NET Framework and Visual Studio, NuGet package manager and third-party libraries			
Unit Number: 2	**Title: .NET Framework Fundamentals**	**No. of hours: 8**	

Object-Oriented Programming (OOP) in .NET, Classes, objects, and inheritance, Exception Handling and Debugging, Debugging techniques and tools in Visual Studio, Logging and error reporting in .NET applications, Memory Management and Garbage Collection, Automatic memory management in .NET, Garbage collection concepts and algorithms, Finalizers and the Dispose pattern, Performance considerations and best practices

Unit Number: 3	**Title: Building Applications with .NET Framework**	**No. of hours: 12**

Windows Forms and WPF Applications, Introduction to Windows Forms and Windows Presentation Foundation (WPF), Designing user interfaces using WinForms/WPF controls, Event-driven programming and event handling, Data binding and data access in WinForms/WPF applications, ASP.NET Web Development, Data access and validation in ASP.NET applications, Web Services and RESTful APIs, Creating and consuming web services in .NET, Authentication and security considerations in web services.

Unit Number: 4	**Title: Advanced Topics in .NET Framework**	**No. of hours: 12**

Content Summary:

.NET Core and Cross-Platform Development, Introduction to .NET Core and its advantages, Building cross-platform applications with .NET Core, Deploying and hosting .NET Core applications, Entity Framework and Database Connectivity, Overview of Entity Framework and Object-Relational Mapping (ORM), Creating and manipulating databases with Entity Framework, Querying data using LINQ (Language Integrated Query), Handling database migrations and versioning, Windows Communication Foundation (WCF), Introduction to WCF and service-oriented architecture (SOA), Creating and consuming WCF services, Message exchange patterns and bindings in WCF, Security and reliability in WCF applications

*Self-Learning Components:

1. Online Tutorials and Documentation: Direct students to the official Microsoft documentation for .NET Framework, which provides comprehensive guides and resources. Microsoft .NET Documentation
2. Hands-on Coding Exercises: Assign coding exercises from platforms like LeetCode or HackerRank that focus on implementing concepts of .NET

Framework. LeetCode HackerRank

3. Project-Based Learning: Encourage students to work on small projects using different aspects of the .NET Framework. Provide examples of project ideas and resources like GitHub repositories for inspiration. GitHub

Please Note:

1)Students are supposed to learn the components on self-basis

2) At least 5-10 % syllabus will be asked in end term exams from self-learning components.

Reference/Text Books:

1. "Mastering C# and .NET Framework" by Jayantha Dhanapala
2. "Pro C# and .NET Framework" by Andrew Troelsen
3. ".NET Framework Programming with C#" by G. Shankar
4. ".NET Programming: Concepts and Practice" by Atul Kumar

Define Course Outcomes (CO)

COs	Statements
CO1	**Understanding** the fundamental concepts and components of the .NET Framework.
CO 2	**Applying** knowledge to design and develop applications using Windows Forms, WPF, and ASP.NET.
CO 3	**Analyzing** performance considerations and troubleshooting errors in the .NET Framework.

CO 4	**Integrating** advanced topics like .NET Core, Entity Framework, and WCF for cross-platform development and service creation.
CO 5	**Assessing** security, reliability, scalability, and performance of applications developed using the .NET Framework.

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels© 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C1	A1	P1
C02	C3	A2	P2
CO3	C2	A3	P3
CO4	-	-	-
CO5	C5	-	P5

CO-PO Mapping

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12

CO1	3	3	-	-	2	-	-	-	-	1	-	3
CO2	1	3	3		1	-	-	-	-	2	-	3
CO3	1		3	2	1	-	-	-	-	1	-	3
CO4	-	-	3	1	2	-	-	-	-	-	-	3
CO5	-	2	3	-	2	-	-	-	-	1	-	3

CO-PSO Mapping

PO	PSO1	PSO2	POS3	PSO4
CO1	3	3	-	-
CO2	1	3	1	
CO3	2	1	-	1
CO4	-	1	-	1
CO5	1	2	-	-

Relevance of the Syllabus to various indicators

Unit I	**Introduction to .NET Framework**
Local	-
Regional	-
National	Understanding the evolution and history of the .NET Framework provides context specific to the national software development landscape.
Global	-
Employability	Understanding the .NET Framework provides valuable skills and knowledge sought by employers in the software development industry.
Entrepreneurship	Understanding the .NET platform can support the development of innovative software products and services.
Skill Development	Studying the .NET Framework helps develop technical skills in application development and programming.

Professional Ethics	Familiarity with the .NET Framework enables professionals to adhere to ethical practices while developing software applications.
Gender	-
Human Values	-
Environment & Sustainability	-
Unit II	**NET Framework Fundamentals**
Local	Understanding OOP in .NET can have local relevance in terms of specific programming practices and patterns adopted within the local software development community.
Regional	-
National	Understanding classes, objects, and inheritance in the .NET Framework is relevant at all levels of software development.
Global	Aligns with global trends in internet technologies and network protocols
Employability	Knowledge of classes, objects, and inheritance enhances employability in the software development field.
Entrepreneurship	Knowledge of OOP in .NET supports the development of innovative software products and services by entrepreneurs.
Skill Development	Learning about classes, objects, and inheritance improves technical skills required
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	**Building Applications with .NET Framework**

Local	Understanding Windows Forms and WPF applications is relevant at a local level as it involves designing user interfaces and developing desktop applications specific to the local context.
Regional	Understanding Windows Forms and WPF applications is relevant at a regional level as these frameworks are commonly used in software development within a specific region.
National	Introduction to Windows Forms and WPF is important at a national level as these frameworks serve as the foundation for developing various types of applications used nationwide.
Global	Introduction to Windows Forms and WPF is important globally as these frameworks are fundamental to developing user interfaces and applications used on a global scale.
Employability	Understanding the basics of Windows Forms and WPF is crucial for employability in software development roles. Proficiency in these frameworks demonstrates competence and versatility, making individuals more desirable to potential employers.
Entrepreneurship	Knowledge of Windows Forms and WPF applications enables entrepreneurs to create innovative software products and services, driving business growth and success.
Skill Development	Understanding Windows Forms and WPF applications enhances technical skills in software development, enabling individuals to design and develop user-friendly and visually appealing applications.
Professional Ethics	Familiarity with Windows Forms and WPF applications ensures adherence to ethical standards in software development, including data privacy, accessibility, and industry best practices for usability and security.
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	**Advanced Topics in .NET Framework**

Local	the content on .NET Core, Entity Framework, and Windows Communication Foundation (WCF) has local relevance as it addresses the specific development and database connectivity needs within the local context.
Regional	-
National	The content on .NET Core, Entity Framework, and Windows Communication Foundation (WCF) is nationally relevant as it addresses specific development and database connectivity needs within the country, considering national requirements and technologies.
Global	addresses development and database connectivity needs on a global scale, considering international requirements and technologies.
Employability	Content equips individuals with the skills and knowledge necessary for software development roles. Proficiency in these technologies enhances employability prospects and opens up opportunities in the job market.
Entrepreneurship	Understanding these technologies enables entrepreneurs to create and scale their own software ventures.
Skill Development	Learning and applying these technologies contribute to the development of practical skills that are in demand in the industry, improving professional capabilities and career prospects
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
SDG	SDG 9, SDG 4, and SDG 8,
NEP 2020	-
POE/4th IR	the content on .NET Core, Entity Framework, and Windows

	Communication Foundation (WCF) addresses professional ethics and aligns with the demands and innovations of the Fourth Industrial Revolution (4IR).

.NET FRAMEWORK LAB

Department:	**Department of Computer Science and Engineering**		
Course Name: **.NET Framework Lab**	**Course Code**	**L-T-P**	**Credits**
	ENSP463	0-0-2	1
Type of Course:	Minor (Department Elective IV)		
Pre-requisite(s), if any: Nil			

Proposed Lab Experiments

Defined Course Outcomes

COs	
CO 1	**Gain** a thorough understanding of the core concepts and components of the .NET Framework.
CO 2	**Apply** .NET Framework knowledge to design and develop applications, solving programming problems effectively.
CO 3	**Analyze** and troubleshoot .NET applications, using debugging techniques and optimizing performance.
CO 4	**Integrate** advanced .NET topics like .NET Core, Entity Framework, and WCF to create cross-platform applications, work with databases, and build services.

Ex. No	Experiment Title	Mapped CO/COs
1	Installing and setting up the .NET Framework, Visual Studio IDE, and NuGet package manager	CO1
2	Creating a basic console application in C# or Visual Basic.NET and running it in Visual Studio.	CO1
3	Write a program to display "Hello World" using C#.	CO2
4	Create a Windows Forms application to design a simple calculator.	CO2
5	Develop a console application to perform basic arithmetic operations	CO2
6	Create a class hierarchy to represent different types of vehicles.	CO2
7	Implement inheritance and polymorphism concepts in a C# program.	CO2
8	Design a Windows Forms application to manage student records.	CO3
9	Create a WPF application to build a simple photo gallery.	CO3
10	Develop a web application to display and manage a list of books using ASP.NET..	CO3
11	Implement form validation and data access in an ASP.NET application.	CO3
12	Build a RESTful API using ASP.NET Web API to perform CRUD operations on a database.	CO3
13	Create a client application to consume a web service and display the retrieved data.	CO2
14	Implement a cross-platform application using .NET Core.	CO3
15	Develop a database-driven application using Entity	CO3

	Framework for data manipulation.	
16	Design and implement a WCF service to provide secure communication between client and server.	CO4
17	Connect a .NET application to a database using ADO.NET and retrieve data.	CO3
18	Use LINQ (Language Integrated Query) to perform data querying and manipulation operations.	CO3
19	Deploy a .NET application to a web server or a cloud platform.	CO4
20	Configure and manage the hosting environment for a .NET application.	CO4
21	Use debugging techniques and tools in Visual Studio to identify and fix bugs in a program.	CO2
22	Create a program to demonstrate the automatic memory management feature in .NET.	CO4
23	Implement a program to analyze and optimize memory usage in a .NET application.	CO2
24	Develop a WCF service to perform CRUD operations on a database.	CO4
25	Design a client application to consume the WCF service and display the retrieved data.	CO4

NEW AGE PROGRAMMING LANGUAGES

Department:	**Department of Computer Science and Engineering**		
Course Name: **New-Age programming languages**	**Course Code**	**L-T-P**	**Credits**
	ENSP415	4-0-0	4
Type of Course:	Minor (Department Elective IV)		
Pre-requisite(s), if any: Nil			
Brief Syllabus: New-Age programming languages (GO, F#, Clojure, Kotlin) provides an introduction to the concepts and applications of modern programming languages. It explore the features and benefits of GO, F#, Clojure, and Kotlin, and develop practical skills in programming using these languages. The course will cover language syntax, data types, control structures, functional programming concepts, concurrency, and integration with other technologies.			
UNIT WISE DETAILS			
Unit Number: 1	**Title: GO programming Language**	**No. of hours: 10**	
Content Summary: Overview of GO, F#, Clojure, and Kotlin, Comparison with traditional programming languages, Installation and setup of development environment, Introduction to GO syntax and data types, Control structures, Functions and packages in GO, Arrays, slices, and maps in GO, Structs and custom data types, Pointers and memory management, Concurrency and parallelism in GO, Error Handling, Concurrent Programming in GO, Advanced GO Concepts- Function closures and anonymous functions, Reflection and type introspection, Testing and benchmarking in GO, Writing concurrent and parallel programs.			
Unit Number: 2	**Title: F# Programming Language**	**No. of hours: 10**	

Content Summary:

Introduction to F# syntax and functional programming concepts, Data Types, Variables, Operators, Decision Making, Loops, Functions, Strings, Options, Immutable data types and pattern matching, Higher-order functions and currying, Asynchronous and parallel programming in F#, Object-Oriented Programming with F#, Database access with F#, Querying and manipulating data using F#, Integration with relational and NoSQL databases

Unit Number: 3	Title: Introduction to Clojure Programming	No. of hours: 10

Content Summary:

Overview of Clojure and its features, Setting up the development environment, Basic syntax and data structures in Clojure, Functional Programming in Clojure, Immutable data and pure functions, Higher-order functions and recursion, Collections and sequence operations in Clojure, Destructuring and pattern matching, Macros and metaprogramming in Clojure, Concurrency models in Clojure, Asynchronous programming with core.async, Parallel programming with reducers and pmap, Interacting with Java libraries and APIs, Java interoperability in Clojure, Working with Java collections and objects, Web Development with Clojure, Building web applications using Clojure and Ring, Database access and persistence in Clojure, Error Handling and Testing: Exception handling and error management in Clojure, Testing strategies and frameworks in Clojure, Data Manipulation and Transformation: Data manipulation with Clojure's sequence functions, Data transformation with transducers, Data-driven development with data literals and data readers

Unit Number: 4	Title: Introduction to Kotlin Programming	No. of hours: 10

Content Summary:

Overview of Kotlin and its advantages, Setting up the development environment, Basic syntax and data types in Kotlin, Conditional statements and loops, Function declarations and parameters, Lambda expressions and higher-order functions, Object-Oriented Programming in Kotlin: Classes, objects, and inheritance, Properties and access modifiers, Interfaces and abstract classes, Understanding nullable and non-nullable types, Safe calls and the Elvis operator, Type inference and smart casting, Collections and Functional Programming: Working with lists, sets, and maps in Kotlin, Collection operations and transformations, Introduction to functional programming concepts in Kotlin, Creating extension functions in Kotlin, Using DSLs for domain-specific problems, Builder pattern and DSL implementation.

***Self-Learning Components:**

2. Web programming with GO
3. F# for Data Science and Machine Learning:
4. Metaprogramming and DSLs in Clojure:
5. Android App Development with Kotlin:

References:

1. Building Modern Web Applications with Go (Golang) by Udemy
2. https://www.jetbrains.com/academy/
3. https://www.classcentral.com/subject/f-sharp
4. https://www.classcentral.com/subject/clojure

Please Note:

1)Students are supposed to learn the components on self-basis

2) At least 5-10 % syllabus will be asked in end term exams from self-learning components.

Reference Books:

1. The Go Programming Language, Alan A. A. Donovan and Brian W. Kernighan, Addison-Wesley Professional.
2. An Introduction to Programming in Go, Caleb Doxsey, CreateSpace Independent Publishing.
3. Real-World Functional Programming: With Examples in F# and C#, Tomas Petricek and Jon Skeet, Manning.
4. Programming F# 3.0: A Comprehensive Guide for Writing Simple Code to Solve Complex Problems, Chris Smith, O'Reilly Media.
5. Getting Clojure: Build Your Functional Skills One Idea at a Time, Russ Olsen, O'Reilly.
6. The Joy of Clojure, Michael Fogus and Chris Houser, Manning Publication.
7. Atomic Kotlin, Bruce Eckel and Svetlana Isakova, Mindview LLC.
8. Kotlin in Action, Dmitry Jemerov and Svetlana Isakova, Manning Publication.

Online References:

1. https://gobyexample.com/ [
2. https://golang.org/doc/
3. https://www.youtube.com/playlist?list=PLlxmoA0rQ-LwgK1JsnMsakYNACYGa1cjR
4. https://kotlinlang.org/docs/home.html
5. https://docs.microsoft.com/en-us/dotnet/fsharp/
6. https://www.udemy.com/course/learning-functional-programming-with-f/

7. https://clojure.org/guides/getting_started

Course Outcomes (CO)

COs	Statements
CO1	**Understand** the fundamental principles and paradigms of modern programming languages, including functional programming, object-oriented programming, and concurrent programming.
CO2	**Develop** proficiency in using the syntax, data structures, and control flow constructs of each language (GO, F#, Clojure, and Kotlin) to solve programming problems.
CO3	**Explore** the unique features and strengths of each language, such as Go's focus on concurrency, F#'s functional programming capabilities, Clojure's emphasis on immutability and simplicity, and Kotlin's interoperability with existing Java code.
CO4	**Apply** the languages' respective development tools, such as Go's gofmt and go vet, F#'s F# Interactive (FSI), Clojure's Leiningen or Boot, and Kotlin's integrated development environment (IDE) support, to improve code quality and productivity.
CO5	**Design and implement** projects that integrate multiple programming languages, using appropriate inter-language communication mechanisms and libraries (e.g., Go and Kotlin interacting via REST APIs, F# and Clojure communicating via message queue

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels(C) 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C2	A1	-
C02	C3	A2	P2
CO3	C2	A3	-
CO4	C3	A4	P3
CO5	C6	-	P4

CO-PO Mapping

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	2	-	-	2	-	-	-	-	2	1	3
CO2	2	2	-	-	2	-	-	-	-	2	2	3
CO3	2	2	-	3	-	-	-	-	-		-	3
CO4	-	-	-	-	3	-	-	-	-	3	1	2
CO5	-	-	-	-		2	-	-	-	2	-	3

1=weakly mapped
2= moderately mapped
3=strongly mapped

CO-PSO Mapping

PO	PSO1	PSO2	PSO3	PSO4
CO1	3	-	-	3
CO2	3	2	-	-
CO3	-	2	-	3
CO4	-	2	-	3
CO5	-	2	2	2

Relevance of the Syllabus to various indicators

Unit I	Introduction to New-Age Programming Languages and GO programming Language
Local	-
Regional	-
National	Provides essential knowledge and skills related to modern programming languages like GO, F#, Clojure, and Kotlin, which are widely used in national software development projects and initiatives.
Global	Relevant in the global software development community, as these languages and concepts have international adoption and usage.
Employability	Highly valued in the job market, as these languages are used in various industries and offer opportunities for software development roles.
Entrepreneurship	Equips students with knowledge of modern programming languages and advanced concepts, enabling them to explore entrepreneurial opportunities in software development and innovation using these languages.
Skill Development	Enhances students' technical skills in software development, making them more competent in the field of computer science.
Professional Ethics	While not directly related to professional ethics, the syllabus indirectly promotes ethical practices by emphasizing the importance of error handling, writing efficient and concurrent programs, and following best practices in software development.
Gender	-
Human Values	Indirectly supports human values by fostering the development of software solutions that are efficient, maintainable, and user-friendly, aligning with values such as accessibility, usability, and user-centric design.
Environment & Sustainability	-

Unit II	F# Programming Language
Local	-
Regional	-
National	Provides essential knowledge and skills which are relevant at the national level for software development, data analysis, and database management projects.
Global	As a globally recognized programming language used in various industries and research domains.
Employability	Highly valued in the job market, as these skills are sought after by companies and organizations involved in software development, data analysis, and database management.
Entrepreneurship	Equips with knowledge that allows to explore entrepreneurial opportunities in software development, data-centric applications, and database-driven solutions.
Skill Development	Enhances technical skills in programming, functional programming paradigms, and data manipulation, making them more competent in the field of computer science.
Professional Ethics	Indirectly promotes ethical practices by emphasizing good programming practices, data integrity, and security considerations
Gender	-
Human Values	Indirectly supports human values by fostering the development of software solutions that are efficient, maintainable, and user-friendly, aligning with values such as accessibility, usability, and user-centric design.
Environment & Sustainability	-
Unit III	Introduction to Clojure Programming
Local	-
Regional	-

National	Provides essential knowledge and skills which are relevant at the national level for software development, web application development, and data-driven solutions.
Global	As a globally recognized programming language used in various industries and research domains.
Employability	Language is highly valued in the job market. The skills are sought after by companies and organizations involved in software development, web application development, and data analysis.
Entrepreneurship	Explores entrepreneurial opportunities in software development, web application startups, and data-centric solutions
Skill Development	Enhances technical skills, making more competent in the field of computer science
Professional Ethics	Indirectly promotes ethical practices by emphasizing good programming practices, data integrity, and security considerations.
Gender	-
Human Values	Indirectly supports human values by fostering the development of software solutions that are efficient, maintainable, and user-friendly, aligning with values such as accessibility, usability, and user-centric design. .
Environment & Sustainability	-
Unit IV	Introduction to Kotlin Programming
Local	-
Regional	-
National	Provides essential knowledge relevant at the national level for software development, mobile app development, and general-purpose programming.
Global	Language has global applicability.

Employability	Highly valued in the job market. These skills are sought after by companies and organizations involved in software development, mobile app development, and DSL-based solutions.
Entrepreneurship	Explore entrepreneurial opportunities in software development, mobile app startups, and domain-specific language development
Skill Development	Make more competent in the field.
Professional Ethics	Indirectly promotes ethical practices by emphasizing good programming practices, code readability, and modularity.
Gender	-
Human Values	Indirectly by fostering the development of software solutions efficient, maintainable, and user-friendly.
Environment & Sustainability	-
SDG	SDG 4, SDG 8, SDG 9
NEP 2020	Quality education, equity, critical thinking, digital literacy, skill development.
POE/4th IR	Technological advancements, digital transformation, and future-ready skills.

NEW AGE PROGRAMMING LANGUAGES LAB

Department:	**Department of Computer Science and Engineering**		
Course Name:	**Course Code**	**L-T-P**	**Credits**
New Age Programming languages Lab	**ENSP465**	0-0-2	1
Type of Course:	Minor (Department Elective IV)		
Pre-requisite(s), if any: Nil			

Course Outcomes (CO)

COs	Statements
CO1	**Understand** the fundamental principles and paradigms of modern programming languages.
CO2	**Develop** proficiency in using the syntax, data structures, and control flow constructs of each language.
CO3	**Explore** the unique features and strengths of each language, such as Go's focus on concurrency, F#'s functional programming capabilities, Clojure's emphasis on immutability and simplicity, and Kotlin's interoperability with existing Java code.
CO4	**Apply** the languages' respective development tools and best practices.
CO5	**Design and implement** projects that utilize the strengths of each language to tackle complex problems or tasks.

Proposed Lab Experiments

Ex. No	Experiment Title	Mapped CO/COs
	Practical on GO Programming Language	
1	Write a program that takes user input and performs basic calculations (e.g., addition, subtraction, multiplication) using different data types like integers and floats. Use control structures like if statements and loops to handle different scenarios and validate user input.	CO2
2	Create a package that contains multiple functions to perform common tasks, such as string manipulation or mathematical operations. Use these functions in a separate program to demonstrate their functionality and reusability.	CO1
3	Implement a program that stores a collection of elements using arrays. Perform operations like adding, removing, or updating elements	CO2
4	Define a struct Person with the following members: name, age, job and salary. Create methods associated with the struct to read data in structure and print data.	CO4
5	Develop a program that utilizes pointers to modify and manipulate data in memory. Explore concepts like referencing, dereferencing, and memory allocation/deallocation.	CO2
6	Write a program that demonstrates the use of Go routines and channels to achieve concurrent execution of tasks.	CO3
7	Create a program that handles various error scenarios and provides appropriate error messages or responses. Write unit tests for critical functions and verify their correctness using Go's testing package.	CO5
8	**Mini Project**: Task Manager Application in Go Create a task manager application using the Go programming language. The application should allow users to manage their	CO5

		tasks by adding, updating, and deleting tasks. The tasks should have attributes such as title, description, due date, and status (e.g., "in progress", "completed").	
Practicals on F# Programming Language			
9	a.	WAP to read marks of 4subjects and calculate the Percentage of student and find the result according to given conditions 60>=1st Division 60<&& 50>= 2nd Division 50<&& 40>=3rd Division 40<=fail.	CO2
	b.	WAP to accept an integer and check whether it is prime or not.	
10	a.	Write a function that takes a string as input and returns the reverse of the string. Also check if a given string is a palindrome	CO2
	b.	Create a function that takes a string as input and performs the following transformations: i. If the string contains only alphabetic characters, convert it to uppercase. ii. If the string contains only numeric characters, convert it to an integer and double its value. iii. If the string contains a mix of alphabetic and numeric characters, return it as is.	
	c.	Design a function that validates an email address based on specific rules, such as the presence of an '@' symbol and a valid domain name. Use pattern matching to check if the input string matches the expected email format.	
12		Implement a program that performs various operations on lists using higher-order functions (define a list of integers or strings). Write pure functions that demonstrate the map, filter, reduce/fold operations.	CO1
13		Implement a program that performs multiple I/O-bound or computationally intensive tasks concurrently using F#'s asynchronous workflows and parallel programming constructs.	CO3

14	Create a program that demonstrates the object-oriented programming (OOP) capabilities of F#. Define classes, objects, and inheritance hierarchies using F#'s OOP syntax.	CO3
15	Create a program that demonstrates the following tasks: i. Establish a connection to both the relational and NoSQL databases using appropriate database drivers or libraries. ii. Perform basic CRUD operations (Create, Read, Update, Delete) on the databases.	CO4
16	**Mini Project**: Employee Management System Create an Employee Management System using the F# programming language and a relational database. The system should allow users to perform CRUD (Create, Read, Update, Delete) operations on employee records stored in the database. It should provide functionality to add new employees, retrieve employee information, update employee details, and delete employee records.	CO5
Practicals on Clojure Programming Language		
17	Write a program that demonstrates the basic syntax and data structures in Clojure, such as lists, vectors, maps, and sets.	CO1
18	Write functions that manipulate and transform sequences using operations such as map, filter, reduce, and take.	CO2
19	Implement a program that showcases asynchronous programming using the core.async library.	CO3
20	Write code that calls Java methods, creates Java objects, and works with Java collections and objects from Clojure.	CO4
21	Develop a web application using Clojure and the Ring library. Set up routes, handle HTTP requests and responses, and render dynamic content.	CO5
22	Write functions that interact with the database, perform CRUD operations, and handle transactions.	CO5
23	Implement error handling mechanisms, such as exception handling and error management, in Clojure.	CO4

24	**Mini Project**: Blogging Platform with Clojure Create a Blogging Platform using the Clojure programming language. The platform should allow users to create and publish blog posts, manage user accounts, and provide functionality for reading and commenting on blog posts. It should utilize a relational database for data storage and retrieval.		CO5
Practicals on Kotlin Programming Language			
25	19	WAP for print following o/p Hello Kotlin!!!	CO2
	20	WAP to take employee's basic salary, dept_code and experience. Calculate bonus according to following criteria i. dept_code = 101 && exp <= 2 bonus = 3% ii. dept_code = 102 && exp <= 4 bonus = 5% iii. dept_code = 103 && exp <= 7 bonus = 8%	
	21	WAP to accept an integer and display average of digit.	
26	Write a program in Kotlin that demonstrates various aspects of function declarations, parameters, and higher-order functions. a. Implement a function that takes two integer parameters and returns their sum. b. Create a function that has default parameter values for an optional third parameter, which is a string representing a greeting. If no greeting is provided, the function should use a default greeting. c. Explore named parameters by creating a function that takes multiple parameters and demonstrate how to call the function by specifying the parameter names explicitly. d. Implement a variable-length argument function that takes a variable number of integers and calculates their average. e. Utilize a higher-order function by creating a function that accepts a lambda expression as a parameter. The lambda should take an integer parameter and return the square of that integer.		CO2
27	WAP to create a class Student with data members' rollno, student name, course and percentage and member functions to accept and display the details of student.		CO1

	a. Implement properties, methods, and constructors in classes. b. Explore access modifiers and visibility scopes in Kotlin.	
28	Implement a program that demonstrates the declaration and usage of nullable and non-nullable variables. Utilize safe calls (?.) and the Elvis operator (?:) to handle nullable values and provide alternative values or perform fallback actions.	CO3
29	WAP to implement various collections like lists, sets, and maps in Kotlin and perform common operations on them. Use collection functions and transformations such as map, filter, and reduce to manipulate data.	CO2
30	Implement a DSL for a domain-specific problem, showcasing Kotlin's expressive syntax and extension functions.	CO5
31	Implement a program that demonstrates the creation and usage of extension functions in Kotlin (Choose a specific class or data type, such as String). For example, you can create an extension function that counts the number of vowels in a string or reverses the string.	CO3
32	**Mini Project**: Quiz App Build a quiz application that presents users with multiple-choice questions on various topics. Users can select their answers, and the app provides instant feedback on correctness. Keep track of the user's score and display the result at the end of the quiz. Include features like a timer, score calculation, and a database of questions.	CO5

SUMMER INTERNSHIP/MINOR PROJECT

Department:	**Department of Computer Science and Engineering**		
Course Name: Minor Project-III	**Course Code**	**L-T-P**	**Credits**
	ENSI451	---	2
Type of Course:	Project		
Pre-requisite(s), if any: NA			

- Students expected to develop a basic project that demonstrates the application of learnings from studied subjects.
- Students are required to submit a hard copy of project file as per the template. File needs to be submitted in spiral bind.
- Project will be evaluated on the scale of 100 with following evaluation criteria.
    - Project idea & features (10)
    - Literature review (10)
    - Tools & Techniques employed (10)
    - Methodology (10)
    - Presentation of Results and its usefulness (20)
    - Implementation and its understandability (10)
    - Meetings & comments by guide (20)
    - Research paper (10)

File format for Minor project

1.	Abstract	Page No.
2.	Introduction (description of broad topic)	
3.	Motivation	
4.	Literature Review	
5.	Gap Analysis	
6.	Problem Statement	
7.	Objectives	
8.	Tools/platform used	
9.	Methodology	
10.	Experimental Setup	
11.	Evaluation Metrics	
12.	Results And Discussion	
13.	Conclusion & Future Work	
14.	References	
15.	Annexure I: Responsibility Chart	
16.	Annexure II: Screenshots of all the MS-Team Meetings with links (online)/ handwritten comments(offline) from guide	
17.	Annexure III Complete implementation code	
18.	Annexure IV Research Paper (Published/Submitted)	