K.R. MANGALAM UNIVERSITY

K.R. MANGALAM UNIVERSITY

THE COMPLETE WORLD OF EDUCATION

SCHOOL OF ENGINEERING AND TECHNOLOGY

MCA (Master's in Computer Applications)

Specialization in AI & ML

Programme Code: 94

POSTGRADUATE COURSE

Scheme of Study and Syllabi

(with effect from 2026-27)

Session: 2026-28

Approved in the 40th Meeting of Academic Council

Table of Contents

S.N.	Content	Page No.
1	Preamble	03
2	Categories of Courses	05
3	University Vision & Mission	07
4	About the School	08
5	School Vision & Mission	11
6	About the Program	12
7	Program Educational Objectives (PEO)	14
8	Program Objectives (PO)	14
9	Program Specific Objectives (PSO)	16
10	Career Avenues	17
11	Duration & Eligibility Criteria	20
12	Student's Structured Learning Experience in the Programme	21
13	Scheme of Studies	33
14	Syllabus	37

Preamble

Welcome to the School of Engineering & Technology (SOET) at K.R. Mangalam University — a vibrant ecosystem of innovation, technology, creativity and experiential learning dedicated to shaping the next generation of engineers, innovators and technology leaders.

Established in 2013, SOET has emerged as a leading center for industry-focused education, project-based learning and skill development. The School is committed to delivering a transformative learning experience that combines strong academic foundations with practical implementation, innovation and professional excellence. With state-of-the-art infrastructure, advanced laboratories, modern computing facilities and distinguished faculty members, SOET provides an environment where students are empowered to learn, build, innovate and grow.

The School offers a wide range of undergraduate, postgraduate and doctoral programs including B.Tech, BCA, B.Sc, M.Tech and MCA across emerging technology domains. Specialized B.Tech programs in Artificial Intelligence & Machine Learning, Data Science, Cyber Security, Full Stack Development, UI/UX Development and Robotics & AI are carefully designed to align with evolving industry requirements and future technology trends.

At SOET, learning extends far beyond traditional classrooms. The curriculum emphasizes project-based learning, coding culture, real-world problem solving, interdisciplinary learning and hands-on implementation from the very first year. Students actively participate in coding competitions, hackathons, bootcamps, innovation challenges, technical clubs, workshops and industry projects that strengthen technical competency, creativity and professional confidence. Signature initiatives such as *Hack-KRMU*, *MindBenders*, *Robo-Rush* and participation in the *Smart India Hackathon* encourage students to develop innovative solutions for real-world challenges.

The School maintains strong industry integration through collaborations with globally recognized organizations such as IBM, Microsoft, NASSCOM, EC-Council, Xebia and ImaginXP. Industry experts contribute to curriculum design, bootcamp programs, certifications, workshops, internships and mentoring activities, ensuring that students gain practical exposure to modern technologies, tools and industry workflows.

SOET also promotes a strong self-learning and certification culture aligned with NEP 2020 principles. Students are encouraged to pursue MOOCs, global certifications and industry learning pathways through platforms such as Google, AWS, Microsoft, Cisco, Coursera and NPTEL to strengthen lifelong learning and employability skills. Dedicated Centres of Excellence in AI & Machine Learning and Robotics & Automation further support advanced learning, research and innovation activities through cutting-edge infrastructure and practical experimentation facilities.

With its strong focus on employability, innovation, entrepreneurship and practical learning, SOET continues to nurture future-ready professionals capable of contributing effectively to the rapidly evolving global technology ecosystem. We welcome you to a community where excellence, innovation and continuous learning form the foundation of every student's journey.

School of Engineering & Technology

K.R Mangalam University

Categories of Courses

Major: The major would provide the opportunity for a student to pursue in-depth study of a particular subject or discipline.

Ability Enhancement Course (AEC): In alignment with the vision and recommendations of the National Education Policy (NEP) 2020, the Ability Enhancement Courses (AEC) are designed to develop essential communication, analytical, quantitative and professional competencies required for holistic student development, employability and lifelong learning. The courses emphasize practical skill development, critical thinking, problem-solving ability, communication proficiency and workplace readiness through experiential and application-oriented learning approaches.

The AEC curriculum comprising *Foundations of Everyday Communication*, *Professional & Workplace Communication*, *Advanced Communication and Interview Mastery* and *Arithmetic & Reasoning* focuses on strengthening students' verbal and written communication skills, interpersonal effectiveness, aptitude, logical reasoning, interview preparedness and professional confidence. These courses support NEP 2020 objectives by promoting multidisciplinary learning, skill enhancement, experiential education and competency-based development necessary for academic progression, career readiness and active participation in the modern knowledge economy.

The courses are delivered through interactive learning methods including presentations, group discussions, role plays, workplace simulations, aptitude exercises, mock interviews, problem-solving activities and self-learning components to ensure practical exposure and continuous professional growth.

Skills Enhancement Courses (SEC): These courses are aimed at imparting practical skills, hands-on training, soft skills, etc., to enhance theemployability of students.

Discipline Specific Electives/Specialization (DSE/SPC): The purpose of offering discipline-specific

electives is to provide students with the flexibility to specialize in emerging and high-demand domains such as Full Stack Development, Cloud Computing, AI & ML, and Cyber Security. These electives are designed to equip students with advanced knowledge and skills in their chosen fields, ensuring they are well-prepared for specialized roles and industry demands in these cutting-edge areas.

Industry project/Research Project: Students choosing a 2-Year PG degree(MCA) are required to take up research projects under the guidance of a faculty member. The students are expected to complete the Research Project in the final semester. The research outcomes of their project work may be published in peer-reviewed journals or may be presented in conferences /seminars or may be patented.

Value Added Courses (VAC): The purpose of offering Value Added Courses (VAC) is to supplement the regular academic curriculum with industry-oriented knowledge, practical skills, and interdisciplinary competencies that enhance students' overall professional development and employability. These courses are designed to provide learners with additional expertise in emerging and high-demand domains, enabling them to remain competitive in the rapidly evolving technological and professional landscape.

Value Added Courses offer students the flexibility to explore specialized areas such as Full Stack Development, Cloud Computing, Artificial Intelligence & Machine Learning, Cyber Security, Data Science, Internet of Things (IoT), DevOps, Mobile Application Development, Business Analytics, and other contemporary technologies. The courses focus on hands-on learning, problem-solving, innovation, and real-world applications to prepare students for industry-specific roles and future career opportunities.

University Vision & Mission

Vision

K. R. Mangalam University aspires to become an internationally recognizedinstitution of higher learning through excellence in inter-disciplinary education, research and innovation, preparing socially responsible life-longlearners 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.

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 excellentinfrastructure, state of the art Labs, and a team of qualified and research- oriented faculty members.

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

Our strength lies in our highly qualified, research oriented, and committed teaching faculty. We believe in empowering minds through expert guidance, ensuring that our students receive a world-class education that prepares them for the challenges of the ever-evolving technological landscape.

The School of Engineering & Technology is committed to providing a cutting-edge curriculum by integrating the best practices from top global universitiesand leveraging the rich knowledge resources of the Open-Source Society University. The curriculum focuses on problem-solving, design, development, interdisciplinary learning, skill development, research opportunities and application of various emerging technologies with focus on innovative teaching learning methodologies.

We take pride in offering an industry-integrated curriculum that goes beyondtraditional education. Collaborations and training led by industry experts, along with partnerships with renowned organizations such as IBM, Samatrix,Xebia, E.C Council, ImaginXP etc ensure that our students gain practical insights and skills that align with real-world industry demands.

With elective options across various domains, including AI, Cloud Computing, Cyber Security, and Full Stack Development, we empower students to customize their learning experience. Our goal is to provide the flexibility needed for each student to shape their academic and professional future.

We prioritize career growth by offering comprehensive training, placements, international internships, and preparation for further studies. Our commitment to nurturing globally competitive professionals is reflected inthe diverse pathways we pave for our students.

SOET aims at transforming the students into competitive engineers with adequate analytical skills, making them more acceptable to potential employers in the country. At our school, we emphasize learning through doing. Whether it's project-based learning, field projects, research projects, internships, or engaging in competitive coding, our students actively shape their futures by applying theoretical knowledge to practical scenarios. We provide opportunities for industrial projects, R&D projects, and start-upprojects in the final year, ensuring that our students engage in real-world innovation.

We are dedicated to fostering a culture of innovation and entrepreneurship, recognizing these as essential pillars for the success of our students in the rapidly evolving world of technology. We inspire innovation and entrepreneurship through our dynamic Entrepreneurship and Incubation

Center, engaging contests like 'MindBenders' ,'Hack-KRMU,' participation in ‘Smart India Hackathon’, International Conference ‘MRIE’ empowering students to become forward-thinking leaders in the ever-evolving realm of technology.

We pride ourselves on providing state-of-the-art computing facilities and infrastructure. Our modern labs and computing resources are equipped to support the diverse needs of our students, enabling them to engage in advanced research, simulations, and hands-on projects.

K.R. Mangalam University has marked its presence in Delhi NCR as a value- based university, successfully imparting quality education in all domains. Ouralumni are working across all sectors of technology, from MNCs to PSUs.

School Vision & Mission

Vision

To excel in scientific and technical education through integrated teaching,research, and innovation.

Mission

- **Creating** a unique and innovative learning experience to enhance quality in the domain of Engineering & Technology.
- **Promoting** Curricular, co-curricular and extracurricular activities that support overall personality development and lifelong learning, emphasizing character building and ethical behavior.
- **Focusing** on employability through research, innovation and entrepreneurial mindset development.
- **Enhancing** collaborations with National and International organizations and institutions to develop cross-cultural understanding to adapt and thrive in the 21st century.

About the Program

The **Master of Computer Applications (MCA) in Artificial Intelligence (AI) & Machine Learning (ML)** is a specialized postgraduate program designed to equip students with advanced knowledge and skills in AI, ML, and data-driven decision-making. Typically spanning two to three years, the program integrates both theoretical foundations and hands-on practical experience, preparing students for cutting-edge roles in AI research, development, and applications across various industries.

The curriculum covers a broad spectrum of subjects, including machine learning algorithms, deep learning, neural networks, natural language processing (NLP), computer vision, data mining, and big data analytics. Additionally, students gain expertise in key mathematical concepts such as linear algebra, probability, and optimization techniques, which form the backbone of AI and ML models. The program also emphasizes programming skills in languages like Python, R, and frameworks such as TensorFlow and PyTorch.

Beyond core AI and ML topics, the MCA in AI & ML often includes courses in business analytics, software engineering, cloud computing, and communication skills to ensure a well-rounded professional development. Students engage in hands-on projects, internships, and research-driven practical labs that simulate real-world AI applications, fostering critical thinking and problem-solving abilities.

A defining feature of this program is its focus on **emerging AI technologies**, including reinforcement learning, generative AI, edge AI, and AI ethics. This ensures that graduates remain at the forefront of technological advancements and are well-prepared to tackle industry challenges. The program also encourages research and innovation, often culminating in a significant **capstone project or thesis**, allowing students to explore and contribute to state-of-the-art AI developments.

Overall, the MCA in AI & ML is structured to produce highly skilled professionals capable of excelling

in various domains such as AI research, data science, robotics, automation, and intelligent system development, making them valuable assets in the evolving tech landscape.

Programme Outcomes (POs)

Programme Outcomes are statements that describe what the students are expected to know and would be able to do upon the graduation. These relate to the skills, knowledge, and behavior that students acquire through the Programme.

Programme Specific Outcomes (PSOs)

Programme Specific Outcomes define what the students should be able to do at the time of graduation and they are Programme specific. There are two to four PSOs for a Programme.

Programme Educational Objectives (PEOs)

Programme Educational Objectives of a degree Programme are the statements that describe the expected achievements of graduates in their career, and what the graduates are expected to perform and achieve during the first few years after graduation.

Credit

Credit refers to a unit that measures the amount of academic work required for a course. It typically reflects the number of instructional hours per week.

Program Educational Objectives (PEO)

PEO1: Achieve success in industry, government, academia, research, entrepreneurship, or consulting roles related to computer applications.

PEO2: Apply advanced computer application principles and methodologies to solve real-world problems, integrating theoretical knowledge with practical skills.

PEO3: Uphold professional ethics and social responsibilities, contributing to societal well-being and sustainability through the effective use oftechnology.

PEO4: Exhibit leadership and teamwork skills to effectively manage and collaborate in diverse, multidisciplinary projects within the field of computer applications.

Program Outcome (PO)

PO1: Engineering Knowledge: Apply knowledge of mathematics, natural science, computing, engineering fundamentals and an engineering specialization to develop to the solution of complex engineering problems.

PO2: Problem Analysis: Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions with consideration for sustainable development.

PO3: Design/Development of Solutions: Design creative solutions for complex engineering problems and design/develop systems/components/processes to meet identified needs with consideration for the public health and safety, whole-life cost, net zero carbon, culture, society and environment as required.

PO4: Conduct Investigations of Complex Problems: Conduct investigations of complex engineering problems using research-based knowledge including design of experiments, modelling, analysis &

interpretation of data to provide valid conclusions.

PO5: Engineering Tool Usage: Create, select and apply appropriate techniques, resources and modern engineering & IT tools, including prediction and modelling recognizing their limitations to solve complex engineering problems.

PO6: The Engineer and The World: Analyze and evaluate societal and environmental aspects while solving complex engineering problems for its impact on sustainability with reference to economy, health, safety, legal framework, culture and environment.

PO7: Ethics: Apply ethical principles and commit to professional ethics, human values, diversity and inclusion; adhere to national & international laws.

PO8: Individual and Collaborative Team work: Function effectively as an individual, and as a member or leader in diverse/multi-disciplinary teams.

PO9: Communication: Communicate effectively and inclusively within the engineering community and society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations considering cultural, language, and learning differences

PO10: Project Management and Finance: Apply knowledge and understanding of engineering management principles and economic decision-making and apply these to one's own work, as a member and leader in a team, and to manage projects and in multidisciplinary environments.

PO11: Life-Long Learning: Recognize the need for, and have the preparation and ability for i) independent and life-long learning ii) adaptability to new and emerging technologies and iii) critical thinking in the broadest context of technological change.

Program Specific Outcomes (PSOs)

PSO1: Developing and deploying AI-driven applications, models, and intelligent systems to solve real-world problems efficiently.

PSO2: Designing and implementing machine learning algorithms, deep learning models, and data-driven solutions for predictive analytics and decision-making.

PSO3: Ensuring adherence to ethical AI principles, regulatory frameworks, and industry best practices in AI and ML development.

PSO4: Applying advanced data science techniques, including natural language processing, computer vision, and reinforcement learning, to build intelligent automation systems.

PSO5: Evaluating and integrating emerging AI technologies such as generative AI, edge computing, and explainable AI into modern industry applications.

Career Avenues

A Master's in Computer Applications (MCA) with a specialization in **Artificial Intelligence & Machine Learning (AI & ML)** opens up diverse career opportunities in the rapidly expanding field of AI-driven technologies. Graduates can explore roles that involve designing intelligent systems, analyzing data, developing machine learning models, and implementing AI solutions across industries. The key career avenues available to MCA in AI & ML graduates are as follows:

1. Machine Learning Engineer

Develops and deploys machine learning models, fine-tuning algorithms, and optimizing AI-driven applications for predictive analytics, automation, and decision-making.

2. Data Scientist

Extracts meaningful insights from complex datasets using statistical analysis, AI models, and data visualization techniques to drive business strategies.

3. Artificial Intelligence Engineer

Designs and implements AI-based solutions, including neural networks, natural language processing (NLP), and reinforcement learning models for various applications.

4. Deep Learning Specialist

Focuses on designing and training deep learning models for applications such as image recognition, speech processing, and autonomous systems.

5. Natural Language Processing (NLP) Engineer

Develops AI models for language-based applications such as chatbots, virtual assistants, speech recognition, and sentiment analysis.

6. Computer Vision Engineer

Works on AI-driven image and video analysis applications, including facial recognition, medical

imaging, and autonomous vehicle vision systems.

7. Big Data Engineer

Handles large-scale data processing, storage, and analytics using AI-driven frameworks such as Hadoop, Spark, and cloud-based platforms.

8. AI Research Scientist

Engages in cutting-edge AI research, developing innovative models and contributing to advancements in AI methodologies and applications.

9. Robotics Engineer

Applies AI and ML algorithms to automate robotics processes, enabling autonomous behavior and intelligent decision-making in robotic systems.

10. AI Ethics and Policy Specialist

Ensures responsible AI development by analyzing ethical considerations, biases in AI models, and compliance with AI regulations and frameworks.

11. Business Intelligence Analyst

Uses AI-driven analytics to provide insights into business performance, helping organizations make data-driven decisions.

12. AI Consultant

Advises businesses on integrating AI and ML solutions into their operations, optimizing processes, and improving efficiency.

13. Data Engineer

Designs and manages data pipelines, ensuring efficient storage, transformation, and retrieval of data for AI and ML applications.

14. AI Product Manager

Oversees the development and deployment of AI-based products, ensuring they align with market needs and business objectives.

15. Cloud AI Engineer

Implements AI solutions on cloud platforms like AWS, Azure, and Google Cloud, optimizing scalability and performance.

16. Reinforcement Learning Engineer

Specializes in AI systems that learn through trial and error, optimizing decision-making in applications like robotics, game development, and autonomous agents.

17. Edge AI Developer

Works on AI models designed for edge computing, enabling real-time decision-making on IoT devices and smart systems.

18. AI Startup Founder/Entrepreneur

Builds AI-driven startups, leveraging AI and ML innovations to develop products and services across various industries.

19. Academia and Research

Pursues AI research in universities and research institutions, contributing to the advancement of AI and ML methodologies.

Graduates with an MCA in AI & ML are well-equipped to work in various industries, including **healthcare, finance, e-commerce, automotive, cybersecurity, and entertainment**, making them highly sought-after professionals in the AI-driven digital era.

Duration

2Years (4 Semesters) - Full-Time Program

Eligibility Criteria

The candidate must have a bachelor's degree in a relevant field such as Computer Science, Information Technology, or Computer Applications from a recognized university or institution, with a minimum aggregate of 50% marks. Additionally, the candidate must have studied Mathematics as a subject in 12th grade and secured at least 50% marks in it. The reservation and relaxation for SC/ST/OBC/PWD and other categories shall be as per the applicable rules of the central or state government.

Student's Structured Learning Experience in the Programme

University Education Objective

Focus on Employability and Entrepreneurship through Holistic Education using Bloom's Taxonomy. By targeting all levels of Bloom's Taxonomy—remembering, understanding, applying, analysing, evaluating, and creating—students are equipped with the knowledge, skills, and attitudes necessary for the workforce and entrepreneurial success. At KRMU we emphasize on learners critical thinking, problem-solving, and innovation, ensuring application of theoretical knowledge in practical settings. This approach nurtures adaptability, creativity, and ethical decision-making, enabling graduates to excel in diverse professional environments and to innovate in entrepreneurial endeavor's, contributing to economic growth and societal well-being.

Importance of Structured Learning Experiences:

A structured learning experience (SLE) is crucial for effective education as it provides a clear and organized framework for acquiring knowledge and skills. By following a well-defined curriculum, teaching-learning methods and assessment strategies, learners can build on prior knowledge systematically, ensuring that foundational concepts are understood before moving on to more complex topics. This approach not only enhances comprehension but also fosters critical thinking by allowing learners to connect ideas and apply them in various contexts. Moreover, a structured learning experience helps in setting clear goals and benchmarks, enabling both educators and students to track progress and make necessary adjustments. Ultimately, it creates a conducive environment for sustained intellectual growth, encouraging learners to achieve their full potential.

At K.R. Mangalam University SLE is designed as rigorous activities that are integrated into the curriculum and provide students with opportunities for learning in two parts:

Inside the Classroom:

Our educational approach within the classroom is designed to foster **cognitive development** and enhance **student-centric learning**. We prioritize active engagement and deep understanding by employing a variety of methods, tools, and techniques. These include **problem-based learning**, **case studies**, **interactive discussions**, and **technology-enhanced learning platforms**. Our faculty focuses on developing critical thinking, analytical reasoning, and problem-solving abilities, ensuring students achieve well-defined **cognitive outcomes**. Additionally, we integrate the use of **modern teaching tools**, such as Learning Management Systems (LMS), virtual labs, and multimedia resources, to enhance the learning experience and accommodate diverse learning styles. This comprehensive approach not only promotes academic excellence but also nurtures independent learning and lifelong intellectual curiosity.

Outside the Classroom:

Beyond the classroom, our focus shifts to developing students' **people skills** and **psychomotor skills** through hands-on experiences in **industry, community, and laboratory settings**. We encourage participation in internships, industrial visits, community engagement projects, and research opportunities, which allow students to apply theoretical knowledge to real-world challenges. These activities build essential interpersonal skills such as **teamwork, leadership, communication**, and **professional networking**. Simultaneously, students engage in **lab-based learning** and technical workshops that refine their psychomotor abilities, including precision, technical expertise, and problem-solving under practical conditions. Through these outside-the-classroom experiences, students gain a holistic skill set that prepares them to excel in both professional and societal contexts, aligning their education with real-world expectations and industry needs.

Educational Planning and Execution

The Master of Computer Application (MCA) at K.R. Mangalam University is designed to foster a holistic educational experience, integrating both theoretical knowledge and practical skills. The program offers students a structured path from entry to exit, ensuring they develop technical expertise, problem-solving skills, and professional competencies.

Entry Phase

Upon entering the MCA program, students are introduced to the foundational concepts of Artificial Intelligence, Machine Learning and programming. This phase is designed to strengthen their understanding of core scientific and technical principles. Courses such as AI & ML, Problem solving and advanced programming concept, Advanced DBMS and Data Structure and Algorithm provide a strong foundation. Students also engage in hands-on laboratory sessions to complement theoretical learning, which helps them connect classroom knowledge with real-world applications.

Orientation Program: The university conducts a **one-day orientation program** for first-year students to familiarize them with the university's environment and key aspects. During the program, students are introduced to the university's highlights, important procedures, key functionaries, and the code of conduct. This orientation serves to ensure that students are well-informed and prepared for a smooth transition into university life.

In the first year, students are exposed to critical problem-solving approaches, basic programming, and ethics in engineering, laying the groundwork for their technical and professional growth.

Induction program: The School organizes a **5-day induction program** for first-year students, aimed at providing them with a comprehensive understanding of the school's various aspects. During the program, students are introduced to learning resources, facilities, and opportunities available to them, along with the rules and regulations governing academic and campus life. The induction also includes faculty

introductions, guidelines on academic conduct, and detailed information about examination and evaluation methods, ensuring students are well-prepared for their academic journey.

Core Learning

As students advance through the program, they delve deeper into core computer science subjects such as Data Structures, Algorithms, Object-Oriented Programming (C++), Operating Systems, and Database Management Systems. This phase emphasizes both theoretical concepts and their practical application through lab work. The learning is enhanced through exposure to industry-standard tools and techniques, including programming languages like Java and Python, and systems for data management and networking.

The structured academic schedule, with a well-distributed credit system over eight semesters, ensures students acquire deep technical knowledge and skills in software development, systems design, and computing technologies. The Summer Internship Programs and Minor Projects in the curriculum allow students to apply their learning in real-life projects, facilitating experiential learning.

Summer Internships: School offers 2-credit summer internships spanning 6 weeks, where students are encouraged to pursue internships in startups, industries, or premier institutions such as IITs, NITs, and IIITs. In addition, students have the opportunity to earn global certifications during this period. The School also organizes in-house summer schools in collaboration with industry partners, providing further avenues for students to gain hands-on experience and enhance their professional skills. These initiatives are designed to offer students practical exposure, helping them develop industry-relevant expertise.

Skill Development

Throughout the program, there is a significant emphasis on developing practical skills and ensuring students are industry-ready. Courses on Artificial Intelligence, Machine Learning, Cloud Computing, and Cybersecurity provide students with cutting-edge knowledge in emerging fields.

Capstone and Exit Phase

In the final semesters, students undertake discipline-specific electives and capstone projects. These projects integrate the knowledge and skills they have acquired over the course of their studies. Electives such as Natural Language Processing, Generative AI, and Blockchain Technologies offer students the flexibility to specialize in areas of their interest.

The final Industrial Project or R&D Project in the eighth semester is a full-time engagement where students work on live industry problems, research projects, or start-up ideas. This project phase, combined with career readiness boot camps and placement preparation activities, ensures that students are equipped to enter the workforce with both technical competence and professional acumen.

Co-Curricular and Extra-Curricular Activities

Students are encouraged to participate in various clubs, societies, and extra-curricular activities. Engagement in activities such as hackathons, coding competitions, and leadership roles in clubs fosters teamwork, leadership, and creativity. These activities complement academic learning, contributing to the students' holistic development.

Ethics and Professional Values

The program places a strong emphasis on ethics and professionalism. Students are taught to incorporate ethical considerations in technological development and decision-making processes. This prepares them to not only be skilled engineers but also responsible professionals who contribute positively to society.

Career Counselling and Entrepreneurship

The university offers comprehensive **career counselling services**, providing students with expert guidance on **job placements, internships**, and **skill development** to help them effectively navigate their career paths. In addition, the university's **incubation center** plays a pivotal role in nurturing **entrepreneurial and leadership skills**, empowering students to explore innovative ideas and launch

their own ventures. These initiatives are designed to equip students with the tools and resources necessary for professional success and entrepreneurial growth.

Course Registration

Every student has to register at the beginning of each semester for the courses offered in the given semester. Major courses are registered centrally for the students. However, for other multidisciplinary courses (DSE, OE) the students have to register by themselves through ERP.

Student Support Services

Mentor-Mentee: At K.R. Mangalam University, the **Mentor-Mentee Program** plays a crucial role in fostering academic and personal growth. Each student is assigned a faculty mentor who serves as a guide throughout their academic journey. This program ensures continuous interaction, where mentors assist students with academic planning, help in resolving personal issues, and provide career guidance. The mentor-mentee relationship transcends the classroom and often involves personal development, professional growth, and overall well-being. The program aims to nurture a supportive environment that enhances the learning experience and helps students reach their full potential.

Counselling and Wellness Services: The university places a strong emphasis on the mental and emotional well-being of its students through its Counselling and Wellness Services. A dedicated team of trained counselors provides personalized sessions, workshops, and wellness programs to address the mental health needs of the student community. These services focus on holistic well-being, including stress management, emotional resilience, and coping strategies. Regular wellness programs, meditation sessions, and mental health awareness campaigns are conducted to promote a balanced lifestyle and ensure that students can focus on their studies while maintaining their emotional health.

Evaluation of Learning:

At K.R. Mangalam University, assessment and evaluation are integral components of the teaching-learning process, designed to ensure continuous academic progress and holistic development of students. The university follows a Learning Outcome-Based Framework (LOCF), where assessments are aligned with the specific learning outcomes of each program. A variety of assessment methods, including assignments, presentations, quizzes, practical examinations, and project work, are used to gauge students' understanding. The examination system is 100% automated, ensuring timely and transparent evaluation processes. Results are processed efficiently, typically within 13 days, and complaints related to evaluation are minimal, reflecting the university's commitment to maintaining a high standard of academic integrity. This robust system of continuous assessment and feedback fosters a culture of academic excellence and skill development among students.

Capstone Project-Based Learning & Self-Learning Ecosystem

The School of Engineering & Technology (SOET), K.R. Mangalam University strongly believes in experiential, project-based and self-driven learning as essential components of modern engineering education aligned with the vision of NEP 2020 and contemporary industry expectations. To strengthen practical understanding, innovation, problem-solving ability and lifelong learning skills, every course incorporates a **Capstone Project Assignment** as an integral academic component.

The Capstone Project is designed as the **5th assignment** within each course, where students are expected to apply the cumulative knowledge, concepts, tools and techniques learned across all units of the subject to solve a practical mini-project or real-world problem. Unlike traditional assignments focused only on theoretical understanding, the Capstone Project emphasizes implementation, experimentation, creativity, analysis, design thinking and solution development.

The Capstone Assignment encourages students to work in a highly self-learning and exploratory mode.

Students are expected to independently explore additional technologies, frameworks, documentation, research materials, online resources and practical implementation approaches beyond the prescribed syllabus. Faculty members act as mentors and facilitators, guiding students towards industry-oriented thinking, innovation and structured problem-solving rather than limiting learning only to classroom delivery.

As part of this initiative, students are expected to invest approximately **45–50 hours of additional self-learning and implementation effort** in completing the Capstone Project Assignment. This approach helps students strengthen:

- practical implementation skills,
- coding and development capabilities,
- analytical and design thinking,
- documentation and presentation skills,
- research orientation,
- teamwork and collaboration,
- independent learning habits and
- professional confidence.

To further promote lifelong learning and continuous skill development, faculty members actively encourage students to undertake additional MOOCs, industry certifications, open-source learning activities, technical workshops, coding platforms and free online courses relevant to the course domain. Students are motivated to explore modern industry trends, emerging technologies and interdisciplinary applications beyond the prescribed curriculum to become self-sufficient learners and future-ready

professionals.

The Capstone Project framework also promotes:

- experiential learning,
- innovation culture,
- portfolio development,
- industry readiness,
- practical problem-solving and
- outcome-based education practices.

Through this structured self-learning ecosystem, students gradually develop the ability to independently learn, adapt and innovate in rapidly evolving technological environments, thereby nurturing the qualities of lifelong learners, innovators and responsible technology professionals.

Capstone Project Evaluation Rubrics (recommended) (20 Marks)

Evaluation Component	**Description**	**Marks**
Problem Understanding & Solution Design (Viva-Based Evaluation)	Understanding of the problem statement, clarity of objectives, architecture/design thinking, justification of approach, technical decision-making and conceptual understanding evaluated through viva/personal interaction	5
Technical Implementation & Development Contribution (GitHub-Based Evaluation)	Quality of implementation, modularity, coding practices, repository management, version control practices, frequency	5

	and quality of GitHub commits, documentation and overall development contribution	
Outcomes, Use-Case Relevance & Presentation	Practical outcomes achieved, demonstration of working solution, real-world applicability, innovation, use-case relevance, presentation quality and ability to communicate project outcomes effectively	5
Self-Learning & Additional Exploration	Evidence of independent learning, exploration of additional tools/technologies, online learning resources, certifications, research or implementation beyond classroom discussions	5
Total		**20**

Flexibility in Evaluation Rubrics

While the above rubric framework provides a standard guideline for Capstone Project evaluation across programs and courses, individual faculty members may suitably customize, refine or redefine the evaluation rubrics based on:

- the nature of the course,
- project complexity,
- domain-specific requirements,
- implementation expectations,
- industry relevance and
- learning outcomes of the subject.

Faculty members are encouraged to adopt flexible and outcome-oriented evaluation practices that appropriately assess innovation, practical implementation, technical depth, research orientation, creativity, teamwork and self-learning demonstrated by students during the Capstone Project work.

Evaluation Scheme:

Evaluation Components	**Weightage**
Internal Assessments: **Continuous Assessment (40 Marks)** **(Minimum 5 components to be used and to be evenly spaced)** Project/ Quizzes/Test/ Assignments and Essays/ Presentations/ Class Participation/ Case Studies/Lab Work/ Reflective Journals	**40 Marks**
Mid Term Exam	**20 Marks**
External Assessments: End term Examination	**40 Marks**
Total	**100 Marks**

Feedback and Continuous Improvement Mechanisms:

K.R. Mangalam University is deeply committed to academic excellence through a robust **feedback and continuous improvement system**. This system is designed to gather comprehensive input from a diverse range of stakeholders, including **students, faculty, alumni, employers, and academic peers**. Feedback is systematically collected and thoroughly analyzed to identify areas for enhancement in **curricula, teaching methodologies, and academic processes**. Based on the insights gained, actionable measures are formulated and communicated to the appropriate bodies for timely implementation.

This structured feedback mechanism ensures that the university's programs remain aligned with **industry**

trends and societal needs, providing students with a cutting-edge education that prepares them for real-world challenges. Moreover, the university demonstrates its commitment to continuous improvement through **regular curriculum updates** and the integration of **innovative teaching strategies**, fostering an environment where both faculty and students can grow and excel. By maintaining this cycle of feedback and improvement, K.R. Mangalam University ensures the continuous advancement of its academic offerings and the overall learning experience.

Academic Integrity and Ethics:

K.R. Mangalam University upholds the highest standards of academic integrity and ethics as a core value of its educational philosophy. The university implements a zero-tolerance policy towards academic misconduct, including plagiarism & other unethical practices. To ensure transparency and honesty in academic work, plagiarism detection software like Drill bit is used to maintain the originality of student submissions and research outputs. Students and faculty are regularly sensitized on the importance of ethical behavior through workshops, seminars, and classroom discussions. The university also integrates ethics and professional values into its curriculum across various disciplines.

Scheme of Studies

Program Name	Master of Computer Applications (AI & ML)
Total Credits	81
Total Semesters	4

Credit Distribution Summary

Program Name	I	II	III	IV	Total Credits
MCA	23	24	20	14	81

SEMESTER I

SN	Course Code	Category	Course Title	L	T	P	Credit
1	CSE601	Major	Problem Solving and Advanced Programming Concepts	3	0	2	4
2	CSE602	Major	Data Structures and Algorithms	3	0	2	4
3	CSE603	Major	Advanced Database Management Systems	3	0	2	4
4	AML601	DSE/SPC	Data Engineering, Visualization & Feature Engineering	3	0	2	4
5	SECSE001	SEC	Data Analysis with Power BI & KNIME	1	0	2	2
6	SECSE002	SEC	Linux Systems & Shell Scripting Lab	0	0	4	2
7		AEC	Professional & Workplace Communication	**3**	**0**	**0**	**3**
			TOTAL	**16**	**0**	**14**	**23**

SEMESTER II

SN	Course Code	Category	Course Title	L	T	P	Credit
1	CSE605	Major	Modern Java Programming & Application Development	3	0	2	4
2	CSE607	Major	Advanced Operating Systems Engineering	3	0	2	4
3	CSE608	Major	Advanced Computer Networks	3	0	2	4
4	AML602	DSE/SPC	Machine Learning: Foundations to Advanced Models	3	0	2	4
5	SEAML001	SEC	AI Fluency & Digital Intelligence	2	0	2	3
6	PRAML051	Proj	Minor Project	0	0	4	2
7		AEC	Advanced Communication and Interview Mastery	3	0	0	3
	TOTAL			**16**	**0**	**14**	**24**

SEMESTER III

SN	Course Code	Category	Course Title	L	T	P	Credit
1	AML701	DSE/SPC	Deep Learning, NLP & Generative AI	3	0	2	4
2	AML702	DSE/SPC	AI Engineering, MLOps & Intelligent Systems Development	3	0	2	4
3		VAC	Value Added Course-I	2	0	0	2
4	CSE013	Major	Professional Problem Solving & Career Readiness-I	2	0	4	4
5	SIAML051	INT	Summer Internship	0	0	4	2
6	PRAML052	Proj	Capstone Project-I	0	0	8	4
	TOTAL			**10**	**0**	**20**	**20**

SEMESTER IV

SN	Course Code	Category	Course Title	L	T	P	Credit
1	INAML051	Proj/INT	Capstone Project II /Industry Internship	0	0	24	12
2		VAC	Value Added Course-II	2	0	0	2
			TOTAL	2	0	24	14

Total Credits: 81

Syllabus

Semester: 1

PROBLEM SOLVING AND ADVANCED PROGRAMMING CONCEPTS

Program Name	**MCA AI ML**	**Semester**	**I**
Course Name:	**Course Code**	**L-T-P**	**Credits**
Problem Solving and Advanced Programming Concepts	CSE601	3-0-2	4
Type of Course:	Major		
Pre-requisite(s): Intermediate-level programming in Python, with a solid understanding of data structures (lists, dictionaries, sets), control structures, functions, recursion, and basic algorithmic problem-solving.			

Course Description:

This course introduces modern Python programming through a practical, industry-oriented, and problem-solving approach designed for MCA students from diverse academic backgrounds. The course focuses on computational thinking, structured programming, modular development, data handling, automation, analytical programming, and real-world application development using Python.

Students will learn programming fundamentals, collections, functions, object-oriented programming, file handling, APIs, databases, data visualization, and automation techniques through hands-on coding and mini applications. The course emphasizes practical implementation skills required for software development, backend programming, scripting, analytics, automation, and technical interviews.

The laboratory component focuses on building small real-world utilities, automation scripts, data-

processing systems, analytical dashboards, and API-driven applications using modern Python libraries and tools.

The Course Outcomes (COs).

On completion of the course the participants will be:

CO	Course Outcome	Bloom's Level
CO1	Apply Python fundamentals for problem solving.	L3 – Apply
CO2	Develop modular and data-driven applications.	L3 – Apply
CO3	Build Python applications using APIs and databases.	L3 – Apply
CO4	Analyze datasets using Python tools.	L4 – Analyze

CO–PO Articulation Matrix

COs \ POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	2	1	1	2	-	-	1	1	-	2
CO2	3	2	3	2	3	1	1	2	2	1	2
CO3	3	3	3	2	3	1	1	2	2	2	2
CO4	2	3	2	3	3	1	1	1	2	1	3

CO–PO Articulation Matrix

COs / PSOs	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	2	2	1	1
CO2	3	3	2	2	3
CO3	3	3	3	2	3

COs / PSOs	PSO1	PSO2	PSO3	PSO4	PSO5
CO4	2	3	3	3	3

Course Outline:

Unit Number: 1	Title: Python Fundamentals & Computational Thinking	No. of hours: 12

Content:

- **Introduction to Computational Thinking:** problem decomposition, logical reasoning and algorithmic thinking
- **Python Environment & Development Tools:** Python installation, VS Code, Jupyter Notebook and Google Colab
- **Python Basics:** variables, data types, identifiers, operators and type conversion
- **Input/Output Operations:** formatted printing, user interaction and dynamic output generation
- **Control Flow Statements:** conditional statements, loops, nested structures, break and continue
- **Functions & Modular Programming:** function creation, parameters, return values and reusable code
- **Debugging & Error Handling Basics:** syntax errors, runtime errors and debugging practices
- **Problem Solving Applications:** grading systems, billing systems, attendance systems and menu-driven programs

Unit Number: 2	Title: Collections, Data Processing & Analytical Programming	No. of hours: 11

Content:

- **Python Collections:** lists, tuples, sets, dictionaries and nested collections
- **String Processing & Text Manipulation:** searching, splitting, joining, formatting and validations
- **List Comprehensions & Data Transformation Techniques**
- **Collection-Based Problem Solving:** frequency analysis, filtering, grouping and summarization
- **Introduction to NumPy:** arrays, indexing, slicing and mathematical operations
- **Introduction to Pandas:** Series, DataFrames, filtering, sorting and grouped analysis

- **Data Cleaning & Preprocessing:** handling missing values and duplicate removal
- **Applications in Student Analytics & Business Reporting**

Unit Number: 3	Title: File Handling, APIs, Databases & Web-Oriented Python	No. of hours: 11

Content:

- **Object-Oriented Programming Basics:** classes, objects, constructors and methods
- **File Handling:** text files, CSV handling and JSON processing
- **Exception Handling:** try-except blocks, validation handling and defensive programming
- **Modules & Packages:** reusable utilities and modular code organization
- **Working with APIs:** request-response cycle, JSON responses and public API consumption
- **SQLite Database Operations:** CRUD operations and Python database connectivity
- **Introduction to Flask:** routes, templates and simple web application basics
- **Automation Applications:** report generation, file organization and workflow automation

Unit Number: 4	Title: Data Visualization & Industry-Oriented Applications	No. of hours: 11

Content:

- **Advanced Pandas Operations:** grouping, aggregation and merging datasets
- **Data Visualization:** Matplotlib, Seaborn, line charts, bar charts, histograms and scatter plots
- **Dashboard-Oriented Reporting:** analytical summaries and visual reporting techniques
- **Functional Programming Basics:** lambda expressions, map(), filter() and reduce()
- **Web Data Handling:** processing JSON and API-based datasets
- **Analytical Applications:** sales analysis, customer feedback analysis and productivity tracking
- **Performance Optimization & Code Refactoring Basics**

Laboratory Experiments – CO Mapping Table

S. No.	Experiment Title	Mapped CO(s)	Bloom's Level
1	Expense & Attendance Management System	CO1	L3 – Apply
2	Student Data Processing & Reporting System	CO2	L3 – Apply
3	NumPy & Pandas Analytics Toolkit	CO2	L3 – Apply
4	API & File Automation System	CO3	L3 – Apply
5	SQLite & Flask Mini Application	CO3	L3 – Apply
6	Sales Analytics & Visualization Dashboard	CO4	L4 – Analyze
7	JSON & API Data Analysis System	CO4	L4 – Analyze
8	Capstone: Smart Productivity & Analytics Platform	CO1, CO2, CO3, CO4	L4 – Analyze

Unit 1 Lab Assignment: Expense & Attendance Management System	CO1
Problem Statement	Develop a Python application for managing expenses, attendance and formatted reports.
Sub-Problems	1. Accept and validate user inputs 2. Generate attendance summaries and reports 3. Create grading and billing systems 4. Implement menu-driven interaction 5. Handle invalid inputs using validation techniques 6. Generate daily and monthly summaries
Unit 2 Lab Assignment: Student Data Processing & Reporting System	CO2
Problem Statement	Develop a Python application for processing and analyzing student and business datasets.

Sub-Problems	1. Process records using lists and dictionaries 2. Perform filtering, sorting and grouping operations 3. Analyze frequencies and categorized data 4. Use NumPy arrays for numerical processing 5. Use Pandas DataFrames for reporting and analysis 6. Generate summarized analytical outputs
Unit 3 Lab Assignment: API & File Automation System	CO3
Problem Statement	Build a Python application that automates reports and processes online and offline data sources.
Sub-Problems	1. Read and write CSV and JSON files 2. Consume public APIs and process responses 3. Store records using SQLite databases 4. Create simple Flask routes and templates 5. Generate automated reports and summaries 6. Organize reusable modules and packages
Unit 4 Lab Assignment: Sales Analytics & Visualization Dashboard	CO4
Problem Statement	Develop a data analytics dashboard for generating visual insights from real-world datasets.
Sub-Problems	1. Clean and preprocess datasets 2. Analyze grouped and filtered data 3. Create charts and graphical reports 4. Compare trends and categories 5. Generate dashboard-style analytical summaries 6. Process JSON and API-based datasets

Capstone Lab Assignment: Smart Productivity & Analytics Platform	CO1, CO2, CO3, CO4
Problem Statement	Design and develop a Python-based productivity and analytics platform that combines automation, reporting, visualization and API-based data processing.
Sub-Problems	1. Develop task and activity tracking modules 2. Store records using files and SQLite databases 3. Integrate API-based information retrieval 4. Generate visual analytical reports 5. Automate reporting workflows 6. Process structured datasets using Pandas 7. Build reusable modular utilities 8. Present dashboard-oriented outputs and summaries

Text Books

- Python Crash Course, Eric Matthes, 3rd Edition, No Starch Press, 2023. ISBN: 9781718502708
- Python for Data Analysis, Wes McKinney, 3rd Edition, O'Reilly Media, 2022. ISBN: 9781098104030
- Automate the Boring Stuff with Python, Al Sweigart, 2nd Edition, No Starch Press, 2019. ISBN: 9781593279929

Recommended Reference Books

- Python Data Science Handbook, Jake VanderPlas, 2nd Edition, O'Reilly Media, 2022., ISBN: 9781098121228
- Fluent Python, Luciano Ramalho, 2nd Edition, O'Reilly Media, 2022. ISBN: 9781492056355

- Flask Web Development, Miguel Grinberg, 2nd Edition, O'Reilly Media, 2018. ISBN: 9781491991732

Recommended Online Resources

- Python Official Documentation — https://docs.python.org/3/
- Pandas Documentation — https://pandas.pydata.org/docs/
- NumPy Documentation — https://numpy.org/doc/
- Flask Documentation — https://flask.palletsprojects.com/
- Kaggle Learn Python — https://www.kaggle.com/learn/python
- freeCodeCamp Python Course — https://www.freecodecamp.org/learn/scientific-computing-with-python/
- Real Python Tutorials — https://realpython.com/

DATA STRUCTURES AND ALGORITHMS

Program Name	**MCA AI ML**	**Semester**	**I**
Course Name:	**Course Code**	**L-T-P**	**Credits**
Data Structures and Algorithms	CSE602	3-0-2	4
Type of Course:	Major		
Pre-requisite(s): Intermediate-level programming in Python, with a solid understanding of data structures (lists, dictionaries, sets), control structures, functions, recursion, and basic algorithmic problem-solving.			

Course Description:

This course introduces students to fundamental data structures, algorithmic problem solving, and computational optimization techniques essential for software development, competitive programming, technical interviews, and modern computing systems. The course focuses on designing efficient solutions using appropriate data structures and algorithmic paradigms for real-world computational problems.

Students will learn arrays, linked lists, stacks, queues, trees, graphs, hashing, recursion, sorting, searching, dynamic programming, greedy algorithms, and complexity analysis through implementation-oriented learning and coding-based problem solving. The course emphasizes practical applications, runtime benchmarking, memory optimization, debugging, and problem-solving patterns frequently used in technical interviews and industry assessments.

The laboratory component focuses on implementation, visualization, optimization, benchmarking, and real-world applications using C++/Python and STL-based programming practices.

The Course Outcomes (COs).

On completion of the course the participants will be:

CO	Course Outcome	Bloom's Level
CO1	Apply linear data structures for problem solving.	L3 – Apply
CO2	Implement trees, hashing and graph structures.	L3 – Apply
CO3	Design algorithms using standard optimization techniques.	L3 – Apply
CO4	Analyze runtime, memory usage and scalability of algorithms.	L4 – Analyze

CO–PO Articulation Matrix

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

CO–PSO Articulation Matrix

COs \ PSOs	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	3	2	1	2
CO2	3	3	3	2	2
CO3	3	3	3	3	3
CO4	2	3	3	3	2

Course Outline:

Unit Number: 1	Title: Foundations of Data Structures & Linear Data Structures	No. of hours: 12

Content:

- **Introduction to Data Structures & Algorithms:** abstract data types, algorithmic thinking and role of DSA in software engineering
- **Complexity Analysis:** time complexity, space complexity, Big-O, Big-Ω and Big-Θ notations
- **Arrays & Dynamic Arrays:** traversal, insertion, deletion, searching and memory representation
- **Strings & String Operations:** pattern matching basics, string manipulation and text processing applications
- **Recursion & Recursive Problem Solving:** recursion trees, backtracking basics and recursive implementation patterns
- **Linked Lists:** singly, doubly and circular linked lists with operations and applications
- **Stacks & Queues:** stack operations, queue types, circular queue, deque and priority queue basics
- **Applications of Linear Data Structures:** browser history, undo-redo systems, scheduling and expression evaluation

Unit Number: 2	Title: Trees, Hashing & Graph Data Structures	No. of hours: 11

Content:

- **Tree Fundamentals:** tree terminology, binary trees and tree traversal techniques
- **Binary Search Trees (BST):** insertion, deletion, searching and balanced search concepts
- **Heap Data Structure:** min-heap, max-heap, heap operations and heap sort
- **Hashing Techniques:** hash functions, collision handling and hash table implementation
- **Trie & Prefix-Based Searching Basics**
- **Graph Representation:** adjacency matrix and adjacency list
- **Graph Traversal Algorithms:** BFS and DFS traversal techniques

- **Shortest Path & Spanning Tree Concepts:** Dijkstra's algorithm, Prim's and Kruskal's algorithms
- **Applications of Trees & Graphs:** file systems, social networks, recommendation systems and routing applications

Unit Number: 3	Title: Sorting, Searching & Algorithm Design Techniques	No. of hours: 11

Content:

- **Searching Algorithms:** linear search, binary search and search optimization techniques
- **Sorting Algorithms:** bubble sort, insertion sort, selection sort, merge sort, quick sort and heap sort
- **Divide-and-Conquer Technique:** recursive decomposition and recurrence analysis
- **Greedy Algorithms:** activity selection, fractional knapsack and Huffman coding
- **Dynamic Programming Concepts:** memorization, tabulation and optimization strategies
- **Dynamic Programming Applications:** Fibonacci, knapsack problem and longest common subsequence
- **Backtracking Techniques:** N-Queens and subset sum problems
- **Competitive Programming Patterns:** sliding window, two pointers and hashing-based optimization
- **Algorithm Benchmarking & Runtime Comparison**

Unit Number: 4	Title: Advanced Problem Solving & Real-World Applications	No. of hours: 11

Content:

- **Advanced Graph Algorithms:** shortest path optimization and topological sorting
- **String Matching Algorithms:** Rabin-Karp and KMP algorithms
- **Algorithm Engineering:** profiling, benchmarking and memory optimization
- **Scalability & Performance Analysis:** runtime comparison on large datasets
- **Parallel & Distributed Algorithm Basics**
- **Data Structures in Modern Systems:** databases, AI systems, search engines and cloud applications
- **Problem Solving for Technical Interviews & Coding Assessments**

Laboratory Experiments – CO Mapping Table

S. No.	Experiment Title	Mapped CO(s)	Bloom's Level
1	Complexity & Runtime Benchmarking System	CO1	L3 – Apply
2	Linear Data Structure Operations Toolkit	CO1	L3 – Apply
3	Stack, Queue & Expression Evaluation System	CO1	L3 – Apply
4	Tree Traversal & Heap Operations Simulator	CO2	L3 – Apply
5	Hashing & Graph Traversal System	CO2	L4 – Analyze
6	Sorting & Searching Performance Comparator	CO3	L3 – Apply
7	Dynamic Programming & Greedy Optimization System	CO3	L4 – Analyze
8	Capstone: Smart Delivery Route Optimization System	CO1, CO2, CO3, CO4	L4 – Analyze

Unit 1 Lab Assignment: Linear Data Structures	CO1
Problem Statement	Develop a Python application for managing expenses, attendance and formatted reports.
Sub-Problems	1. Implement array operations and dynamic array resizing 2. Perform string manipulation and searching operations 3. Implement singly and doubly linked lists 4. Implement stack and queue operations 5. Simulate browser history or undo-redo systems using stacks 6. Compare runtime performance of linear data structure operations
Unit 2 Lab Assignment: Trees, Hashing & Graphs	CO2
Problem Statement	Develop programs for representing and analyzing hierarchical and network-based data structures.

Sub-Problems	1. Implement binary tree and BST operations 2. Perform tree traversal techniques 3. Implement heap operations and heap sort 4. Implement hashing with collision handling 5. Represent graphs using adjacency structures 6. Implement BFS and DFS graph traversal algorithms	
Unit 3 Lab Assignment: Sorting & Optimization Algorithms		CO3
Problem Statement	Develop optimized solutions using sorting, searching and algorithm design techniques.	
Sub-Problems	1. Implement and compare sorting algorithms 2. Implement binary search and search optimization techniques 3. Solve activity selection and fractional knapsack problems 4. Implement dynamic programming approaches for optimization problems 5. Solve N-Queens and subset sum problems using backtracking 6. Benchmark runtime and memory efficiency of algorithms	
Unit 4 Lab Assignment: Advanced Algorithms & Scalability Analysis		CO4
Problem Statement	Analyze advanced algorithms and evaluate scalability for real-world computational problems.	
Sub-Problems	1. Implement shortest path and topological sorting algorithms 2. Implement Rabin-Karp and KMP string matching algorithms 3. Analyze runtime and memory usage on large datasets 4. Compare exact and optimized algorithmic solutions 5. Visualize scalability trends using graphs and charts	
Capstone Lab Assignment: Smart Delivery Route Optimization System		CO1, CO2, CO3, CO4

Problem Statement	Design and develop a delivery route optimization system that computes efficient paths and scheduling strategies using graph and optimization algorithms.
Sub-Problems	1. Represent delivery locations using graph structures 2. Implement shortest path algorithms for route optimization 3. Analyze delivery scheduling using greedy approaches 4. Compare runtime performance for different graph sizes 5. Visualize optimized routes and performance metrics 6. Generate benchmarking and scalability analysis reports

Text Books

- Data Structures and Algorithms Made Easy, Narasimha Karumanchi, 5th Edition, CareerMonk Publications, 2023. ISBN: 9788193245279
- Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest & Clifford Stein, 4th Edition, MIT Press, 2022. ISBN: 9780262046305
- Problem Solving with Algorithms and Data Structures Using Python, Bradley N. Miller & David L. Ranum, 2nd Edition, Franklin, Beedle & Associates, 2019. ISBN: 9781590282571

Recommended Reference Books

- The Algorithm Design Manual Steven S. Skiena, 3rd Edition, Springer, 2020 ISBN: 783030542559
- Algorithms Robert Sedgewick & KevinWayne, 4th Edition, Addison-Wesley Professional, 2011. ISBN: 9780321573513
- Grokking Algorithms Aditya Bhargava, 2nd Edition, Manning Publications, 2024. ISBN: 9781633438538

Recommended Online Resources

- GeeksforGeeks DSA — https://www.geeksforgeeks.org/data-structures/
- VisuAlgo — https://visualgo.net/en
- LeetCode — https://leetcode.com/
- HackerRank Algorithms — https://www.hackerrank.com/domains/algorithms
- USACO Guide — https://usaco.guide/
- MIT OpenCourseWare Algorithms— https://ocw.mit.edu/courses/6-006-introduction-to-algorithms-fall-2011/

ADVANCED DATABASE MANAGEMENT SYSTEM

Program Name	**MCA AI ML**	**Semester**	**I**
Course Name:	**Course Code**	**L-T-P**	**Credits**
Advanced Database Management System	CSE603	3-0-2	4
Type of Course:	Major		
Pre-requisite(s): Prior knowledge of relational databases, SQL, ER modeling, normalization, and basic transactions.			

Course Description:

This course equips students with advanced database concepts required for modern data-driven applications. Emphasis is placed on distributed and NoSQL systems, transaction processing, query optimization, and database security. It also explores current technologies including time-series and graph databases, preparing students for research, development, and high-performance database design.

The Course Outcomes (COs).

On completion of the course the participants will be:

CO	**Course Outcome**	**Bloom's Level**
CO1	Designing and implementing distributed and parallel database architectures for scalable and reliable systems.	L3 – Apply
CO2	Applying advanced transaction processing, concurrency control, and recovery techniques in distributed environments.	L3 – Apply
CO3	Evaluating modern database models including NoSQL, time-series, and graph databases for diverse use cases.	L4 – Analyze

CO4	Assessing and implementing query optimization, access control, and data security mechanisms in large-scale databases.	L4 – Analyze

CO–PO Articulation Matrix

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

CO–PSO Articulation Matrix

COs \ PSOs	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	3	2	1	2
CO2	3	3	3	2	2
CO3	3	3	2	2	3
CO4	3	3	3	3	2

Course Outline:

Unit Number: 1	Title: Distributed and Parallel Databases	No. of hours: 12

Content:

- **Distributed DBMS Architecture:** homogeneous and heterogeneous distributed databases

- **Data Distribution Techniques:** fragmentation, replication, and allocation strategies
- **Distributed Query Processing:** distributed query execution and optimization techniques
- **Parallel Databases:** parallel query execution and intra-query parallelism
- **Consistency in Distributed Systems:** CAP theorem, strong consistency, eventual consistency, and causal consistency
- **Fault Tolerance & Availability:** distributed recovery and high-availability mechanisms

Unit Number: 2	**Title: Advanced Transaction and Concurrency Control**	**No. of hours: 11**

Content:

- **Concurrency Control Techniques:** two-phase locking (2PL), timestamp ordering, and MVCC
- **Deadlock Management:** deadlock detection, prevention, and avoidance techniques
- **Advanced Transaction Models:** nested transactions, long-duration transactions, and compensating transactions
- **Distributed Transactions:** two-phase commit (2PC) and three-phase commit (3PC) protocols
- **Database Recovery Mechanisms:** ARIES recovery algorithm, logging, checkpoints, and rollback recovery
- **Transaction Reliability:** consistency and fault tolerance in distributed environments

Unit Number: 3	**Title: Emerging Database Models**	**No. of hours: 11**

Content:

- **NoSQL Database Models:** document databases (MongoDB), column-family databases (Cassandra), key-value databases (Redis), and graph databases (Neo4j)
- **Time-Series Databases:** architecture and applications of InfluxDB and TimescaleDB
- **Spatial and Temporal Databases:** managing location-aware and time-dependent data
- **Semantic Data Models:** RDF, OWL, and SPARQL fundamentals
- **Modern Data Management Systems:** data lakes and federated database systems

• **Performance & Scalability:** comparison of relational and NoSQL systems for big data applications		
Unit Number: 4	**Title: Query Optimization and Database Security**	**No. of hours: 11**
Content: • **Query Optimization Techniques:** cost-based optimization, join reordering, and bushy tree optimization • **Advanced Query Processing:** materialized views and query rewriting techniques • **Database Access Control Models:** DAC, MAC, RBAC, and ABAC mechanisms • **Database Security Threats:** SQL injection and inference attacks • **Data Protection Techniques:** transparent data encryption (TDE), row-level encryption, and column-level encryption • **Auditing & Compliance:** auditing, monitoring, GDPR, and HIPAA compliance mechanisms		

Laboratory Experiments – CO Mapping Table

S. N	Experiment Title	Mapped CO(s)	Bloom's Level
1	Distributed Inventory & Replication Management System	CO1	L3 – Apply
2	Distributed Transaction & Concurrency Control Simulator	CO2	L3 – Apply
3	NoSQL and Time-Series Data Analytics Platform	CO3	L3 – Apply
4	Secure Query Optimization & Database Security Framework	CO4	L4 – Analyze
5	Capstone: Intelligent Distributed Multi-Model Database System	CO1, CO2, CO3, CO4	L4 – Analyze

Unit 1 Lab Assignment: Distributed Inventory & Replication Management System	CO1

Problem Statement	Design and implement a distributed database application that simulates inventory management across multiple warehouse locations using concepts of fragmentation, replication, distributed query processing, and parallel execution.
Sub-Problems	1. Design distributed schemas for multiple warehouse locations 2. Implement horizontal and vertical data fragmentation 3. Simulate data replication and allocation strategies 4. Execute distributed SQL queries across multiple nodes 5. Compare homogeneous and heterogeneous database configurations 6. Implement parallel query execution for inventory analytics 7. Analyze consistency models under distributed updates 8. Generate reports for stock synchronization across branches
Unit 2 Lab Assignment: Distributed Transaction & Concurrency Control Simulator	CO2
Problem Statement	Develop a transaction processing simulator that demonstrates advanced concurrency control, distributed transactions, deadlock handling, and recovery mechanisms in a multi-user database environment.
Sub-Problems	1. Implement two-phase locking (2PL) mechanisms 2. Simulate timestamp ordering and MVCC protocols 3. Design deadlock detection and prevention algorithms 4. Implement nested and long-duration transactions 5. Simulate distributed transactions using Two-Phase Commit (2PC) 6. Demonstrate Three-Phase Commit (3PC) protocol execution 7. Implement logging, checkpoints, and rollback recovery 8. Simulate ARIES-based recovery operations after system failure

Unit 3 Lab Assignment: NoSQL and Time-Series Data Analytics Platform		CO3
Problem Statement	Build a multi-model database application using NoSQL, graph, and time-series databases to manage and analyze large-scale real-world datasets.	
Sub-Problems	1. Store and retrieve documents using MongoDB 2. Implement key-value caching using Redis 3. Design column-family data models using Cassandra 4. Build graph relationships and queries using Neo4j 5. Store sensor or financial data using InfluxDB or TimescaleDB 6. Perform temporal and spatial data analysis 7. Execute basic RDF and SPARQL queries for semantic data 8. Compare relational and NoSQL database performance for analytics workloads	
Unit 4 Lab Assignment: Secure Query Optimization & Database Security Framework		CO4
Problem Statement	Create a secure database framework that demonstrates query optimization techniques, access control mechanisms, and protection against database security threats.	
Sub-Problems	1. Analyze execution plans using cost-based optimization 2. Implement join optimization and query rewriting techniques 3. Create and manage materialized views for performance improvement 4. Design role-based and attribute-based access control systems 5. Implement secure authentication and authorization modules 6. Detect and prevent SQL injection attacks 7. Apply encryption techniques for sensitive database attributes 8. Generate auditing and compliance reports for secure transactions	
Capstone Lab Assignment: Intelligent Distributed Multi-Model Database		CO1, CO2, CO3, CO4

System	
Problem Statement	Design and develop an enterprise-level intelligent distributed database platform integrating distributed databases, transaction management, NoSQL systems, query optimization, and security mechanisms for large-scale real-world applications.
Sub-Problems	1. Design a distributed multi-location database architecture 2. Implement fragmentation, replication, and distributed storage 3. Develop transaction processing with concurrency control mechanisms 4. Integrate relational and NoSQL databases within one system 5. Store and analyze time-series and graph-based datasets 6. Implement distributed query processing and optimization techniques 7. Apply secure authentication, authorization, and encryption mechanisms 8. Generate analytical dashboards and reporting modules 9. Implement recovery, logging, and fault-tolerance strategies 10. Deploy the system with scalability and high-availability considerations

Text Books

- Abraham Silberschatz, Henry F. Korth and S. Sudarshan, *Database System Concepts*, 7th Edition, McGraw-Hill Education, 2021.
- Hector Garcia-Molina, Jeffrey D. Ullman and Jennifer Widom, *Database Systems: The Complete Book*, 2nd Edition, Pearson Education, 2013.

Recommended Reference Books

- Raghu Ramakrishnan and Johannes Gehrke, *Database Management Systems*, 3rd Edition, McGraw-Hill Education, 2003.
- M. Tamer Özsu and Patrick Valduriez, *Principles of Distributed Database Systems*, 4th Edition, Springer, 2020. ISBN: 9783030262006

- Pramod J. Sadalage and Martin Fowler, *NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence*, Addison-Wesley Professional, 2013. ISBN: 9780321826626
- Eric Redmond and Jim R. Wilson, *Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement*, 2nd Edition, Pragmatic Bookshelf, 2018. ISBN: 9781680502534
- Ian Robinson, Jim Webber and Emil Eifrem, *Graph Databases*, 2nd Edition, O'Reilly Media, 2015. ISBN: 9781491930892

Recommended Online Resources

- Oracle Database Documentation
- MongoDB Documentation
- Apache Cassandra Documentation
- Neo4j Graph Database Documentation
- PostgreSQL Documentation
- MySQL Reference Manual
- Microsoft SQL Server Documentation
- Redis Documentation
- InfluxDB Documentation
- GeeksforGeeks DBMS and NoSQL Tutorials

DATA ENGINEERING, VISUALIZATION & FEATURE ENGINEERING

Program Name	**MCA AI ML**	**Semester**	**I**
Course Name:	**Course Code**	**L-T-P**	**Credits**
Data Engineering, Visualization & Feature Engineering	AML601	3-0-2	4
Type of Course:	DSE/SPC		
Pre-requisite(s): Basic knowledge of databases, Python programming, statistics, and data analysis concepts.			

Course Description:

This course provides comprehensive knowledge of modern data engineering pipelines, data preprocessing, visualization techniques, and feature engineering methods used in machine learning and analytics systems. The course focuses on data collection, transformation, storage, cleaning, feature extraction, visualization dashboards, and scalable data processing frameworks. Students gain practical exposure to ETL pipelines, big data tools, visualization platforms, and feature optimization strategies required for intelligent data-driven applications. The course emphasizes real-world analytics workflows used in industry, business intelligence, and AI systems.

The Course Outcomes (COs).

On completion of the course the participants will be:

COs	**Statements**	**Bloom's Level**
CO1	Explain data engineering concepts, ETL pipelines, and scalable data processing architectures for modern analytics systems.	L2 – Understand
CO2	Apply data preprocessing, transformation, and feature engineering techniques for	L3 – Apply

	machine learning and analytical applications.	
CO3	Analyze datasets using visualization tools, statistical methods, and interactive dashboards to derive business insights.	L4 – Analyze
CO4	Evaluate data engineering frameworks, visualization strategies, and feature selection methods for scalable intelligent systems.	L5 – Evaluate

CO–PO Articulation Matrix

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

CO–PSO Articulation Matrix

COs \ PSOs	**PSO1**	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	3	2	1	1	1
CO2	3	3	2	2	3
CO3	3	3	3	2	2
CO4	3	3	3	3	3
CO5	3	2	1	1	1

Course Outline:

Unit Number: 1	Title: Data Engineering Fundamentals & ETL Pipelines	No. of hours: 12

Content:

- **Introduction to Data Engineering:** data lifecycle, structured and unstructured data
- **Data Sources & Collection:** APIs, databases, logs, streaming data, and web data
- **ETL & ELT Pipelines:** extraction, transformation, and loading techniques
- **Data Warehousing Concepts:** star schema, snowflake schema, and OLAP systems
- **Big Data Ecosystem:** Hadoop, Spark, and distributed data processing basics
- **Data Storage Technologies:** relational, NoSQL, and cloud-based storage systems
- **Workflow Automation:** scheduling and orchestration tools for data pipelines

Unit Number: 2	Title: Data Preprocessing & Feature Engineering	No. of hours: 11

Content:

- **Data Cleaning Techniques:** handling missing values, duplicates, and noisy data
- **Data Transformation Methods:** normalization, standardization, encoding, and scaling
- **Feature Engineering Fundamentals:** feature extraction and feature transformation
- **Feature Selection Techniques:** filter, wrapper, and embedded methods
- **Dimensionality Reduction:** PCA and feature reduction techniques
- **Handling Imbalanced Data:** sampling and balancing methods
- **Data Preparation for ML Models:** train-test splitting and preprocessing pipelines

Unit Number: 3	Title: Data Visualization & Business Intelligence	No. of hours: 11

Content:

- **Data Visualization Fundamentals:** principles of effective data storytelling
- **Visualization Libraries & Tools:** Matplotlib, Seaborn, Plotly, and Tableau basics
- **Dashboard Design:** KPIs, charts, filters, and interactive reports
- **Business Intelligence Systems:** analytics dashboards and reporting platforms
- **Exploratory Data Analysis (EDA):** statistical summaries and trend analysis
- **Geospatial & Time-Series Visualization:** maps and temporal analytics dashboards
- **Visual Analytics:** interpreting patterns and business insights from visual data

Unit Number: 4	Title: Scalable Data Systems & Advanced Analytics	No. of hours: 11

Content:

- **Distributed Data Processing:** Spark transformations and distributed analytics
- **Cloud Data Engineering:** cloud storage and analytics platforms
- **Real-Time Data Analytics:** stream processing and event-driven systems
- **Data Governance & Security:** privacy, compliance, and secure data pipelines
- **Advanced Feature Engineering:** automated feature engineering and feature stores
- **Performance Optimization:** scalable analytics and query optimization techniques
- **Emerging Trends:** AI-driven analytics, AutoML, and intelligent data systems

Laboratory Experiments – CO Mapping Table

S. N	Experiment Title	Mapped CO(s)	Bloom's Level
1	ETL Pipeline & Data Integration System	CO1	L2 – Understand
2	Data Cleaning & Feature Engineering Framework	CO2	L3 – Apply
3	Interactive Dashboard & Visualization Analytics Platform	CO3	L4 – Analyze

4	Distributed Data Analytics & Optimization System	CO4	L5 – Evaluate
5	Capstone: Intelligent End-to-End Data Engineering & Analytics Platform	CO1, CO2, CO3, CO4	L5 – Evaluate

Unit 1 Lab Assignment: ETL Pipeline & Data Integration System		CO1
Problem Statement	Design and implement ETL pipelines for collecting, transforming, and integrating structured and unstructured datasets from multiple data sources.	
Sub-Problems	• Extract data from databases and APIs • Transform raw datasets into structured formats • Implement automated ETL workflows • Load datasets into data warehouses • Analyze data consistency and quality • Schedule pipeline automation tasks • Generate ETL execution reports	
Unit 2 Lab Assignment: Data Cleaning & Feature Engineering Framework		CO2
Problem Statement	Develop preprocessing and feature engineering systems for preparing datasets suitable for machine learning and analytics applications.	
Sub-Problems	• Handle missing and inconsistent data • Apply normalization and encoding techniques • Implement feature extraction methods • Perform feature selection and dimensionality reduction • Handle imbalanced datasets	

	• Build reusable preprocessing pipelines • Generate feature analysis reports	
Unit 3 Lab Assignment: Interactive Dashboard & Visualization Analytics Platform		CO3
Problem Statement	Create interactive visualization dashboards and analytical reports for business intelligence and exploratory data analysis.	
Sub-Problems	• Design charts and visualization reports • Create KPI dashboards and filters • Implement time-series and geospatial visualizations • Perform exploratory data analysis (EDA) • Build interactive business reports using BI tools • Analyze trends and business insights • Generate visual storytelling reports	
Unit 4 Lab Assignment: Distributed Data Analytics & Optimization System		CO4
Problem Statement	Design scalable distributed analytics systems integrating cloud data engineering, stream processing, and advanced feature optimization techniques.	
Sub-Problems	• Configure distributed data processing workflows • Implement Spark-based analytics operations • Analyze streaming datasets in real time • Optimize scalable data processing pipelines • Apply advanced feature engineering strategies • Implement secure and compliant data systems • Generate performance and scalability reports	

Capstone Lab Assignment: Intelligent End-to-End Data Engineering & Analytics Platform	CO1,CO2,CO3,CO4
Problem Statement	Design and develop a scalable intelligent data engineering and analytics platform integrating ETL pipelines, feature engineering, visualization dashboards, distributed analytics, and cloud-based data processing systems.
Sub-Problems	• Design enterprise data engineering architecture • Implement automated ETL and preprocessing workflows • Develop feature engineering and optimization modules • Create interactive analytics dashboards • Integrate distributed and cloud-based analytics systems • Implement real-time data processing pipelines • Apply data governance and security mechanisms • Generate business intelligence and predictive analytics reports • Optimize scalability and analytics performance • Present a complete intelligent analytics platform solution

Textbooks

- Fundamentals of Data Engineering, Joe Reis and Matt Housley, 1st Edition, O'Reilly Media, 2022. ISBN: 9781098108304.
- Joe Reis and Matt Housley, *Fundamentals of Data Engineering*, O'Reilly Media, 2022.Edition, O'Reilly Media, 2020. ISBN: 978-1492051725
- Feature Engineering for Machine Learning, Alice Zheng and Amanda Casari, 1st Edition,

O'Reilly Media, 2018. ISBN: 9781491953242.

Recommended Reference Book

- Wes McKinney, *Python for Data Analysis*, 3rd Edition, O'Reilly Media, 2022. ISBN: 9781098104030
- Cole Nussbaumer Knaflic, *Storytelling with Data*, Wiley, 2015. ISBN: 9781119002253
- Alberto Cairo, *The Truthful Art: Data, Charts, and Maps for Communication*, New Riders, 2016. ISBN: 9780321934079
- Max Kuhn and Kjell Johnson, *Feature Engineering and Selection*, CRC Press, 2019. ISBN: 9781138079229
- Bill Chambers and Matei Zaharia, *Spark: The Definitive Guide*, O'Reilly Media, 2018. ISBN: 9781491912218

Recommended Online Resources

- Apache Spark Documentation
- Hadoop Documentation
- Tableau Learning Resources
- Microsoft Power BI Documentation
- Kaggle Data Engineering Tutorials
- Google Cloud Data Engineering Learning Path
- AWS Data Analytics Documentation
- GeeksforGeeks Data Engineering Tutorials
- TutorialsPoint Big Data and Visualization Tutorials
- Python Pandas & Matplotlib Documentation

Data Analysis with Power BI & KNIME

Program Name	MCA AI ML	Semester	I
Course Name:	**Course Code**	**L-T-P**	**Credits**
Data Analysis with Power BI & KNIME	SECSE001	1-0-2	2
Type of Course:	SEC		
Pre-requisite(s): Basic understanding of statistics, spreadsheets, and introductory data analysis concepts.			

Course Description:

This course provides a hands-on introduction to data analysis and visualization using Power BI and KNIME. Students will learn how to extract, transform, and visualize data, apply analytical models, and generate business insights using both graphical and low-code platforms. The course emphasizes real-world data preparation, dashboard creation, and predictive analysis in a user-friendly environment.

The Course Outcomes (COs).

On completion of the course the participants will be:

CO	Course Outcome	Bloom's Level
CO1	Importing, cleaning, and preparing data using Power BI and KNIME workflows.	L3 – Apply
CO2	Building dynamic dashboards and visualizations to summarize insights.	L3 – Apply
CO3	Applying statistical and machine learning techniques using low-code tools.	L3 – Apply
CO4	Evaluating data-driven decisions through reports, KPIs, and business scenarios.	L4 – Analyze

CO–PO Articulation Matrix

COs \ POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	2	2	2	3	1	1	1	2	1	2
CO2	3	2	3	2	3	1	1	2	3	1	2
CO3	3	3	3	3	3	1	1	2	2	2	3
CO4	2	3	3	3	2	2	1	2	3	2	3

CO–PSO Articulation Matrix

COs \ PSOs	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	3	2	1	2
CO2	3	3	3	2	2
CO3	3	3	2	2	3
CO4	3	3	3	3	2

Course Outline:

Unit Number: 1	**Title: Introduction to Data Analysis and BI Tools**	**No. of hours: 8**
Content: • Introduction to data analysis and business intelligence		

- Types of data: structured, semi-structured, and unstructured
- Data collection, preprocessing, and transformation techniques
- Overview of Business Intelligence (BI) platforms
- Introduction to Power BI interface and components
- Introduction to KNIME Analytics Platform and workflow concepts
- Data import from Excel, CSV, databases, and web sources
- Data quality issues and cleaning techniques

Unit Number: 2	**Title: Power BI for Interactive Dashboards**	**No. of hours: 6**

Content:

- Data loading and transformation using Power Query
- Data modelling and relationships in Power BI
- Creating charts, tables, maps, cards, and slicers
- Designing interactive dashboards and reports
- Introduction to DAX (Data Analysis Expressions)
- Creating calculated columns, measures, and KPIs
- Dashboard publishing and sharing using Power BI Service

Unit Number: 3	**Title: KNIME for Visual Workflow-based Analysis**	**No. of hours: 8**

Content:

- Introduction to KNIME nodes and workflows
- Data filtering, sorting, grouping, and aggregation
- Joining and merging datasets using KNIME
- Exploratory Data Analysis (EDA) techniques
- Data visualization using KNIME analytical nodes
- Workflow automation and reusable analytical pipelines
- Statistical analysis using built-in KNIME components

Unit Number: 4	Title: Basic Predictive Analytics with Power BI & KNIME	No. of hours: 8

Content:

- Introduction to predictive analytics and business forecasting
- Trend analysis and forecasting in Power BI
- Classification techniques using Decision Trees in KNIME
- Clustering using k-Means algorithm
- Model evaluation metrics: accuracy, precision, recall
- Generating analytical and predictive reports
- Business decision-making using data-driven insights

Laboratory Experiments – CO Mapping Table

S. N	Experiment Title	Mapped CO(s)	Bloom's Level
1	Data Cleaning and Preparation using Power BI & KNIME	CO1	L3 – Apply
2	Interactive Dashboard and KPI Development using Power BI	CO2	L3 – Apply
3	Exploratory Data Analysis and Workflow Automation using KNIME	CO3	L3 – Apply
4	Predictive Analytics and Business Insight Generation	CO4	L4 – Analyze
5	Capstone: Integrated Business Intelligence and Predictive Analytics System	CO1, CO2, CO3, CO4	L4 – Analyze

Unit 1 Lab Assignment: Data Cleaning and Preparation using Power BI & KNIME	CO1

Problem Statement	Develop a data preparation workflow using Power BI and KNIME to import, clean, transform, and organize raw datasets for analytical processing.
Sub-Problems	• Import Excel, CSV, and database datasets • Handle missing and inconsistent data values • Perform data transformation and formatting operations • Apply filtering, sorting, and grouping techniques • Merge and join multiple datasets • Create reusable preprocessing workflows • Generate cleaned datasets for reporting and analytics
Unit 2 Lab Assignment: Interactive Dashboard and KPI Development using Power BI	CO2
Problem Statement	Design and develop interactive dashboards and KPI-based business reports using Power BI visualization tools.
Sub-Problems	• Create charts, graphs, maps, and slicers • Develop calculated columns and measures using DAX • Design KPI cards and summary indicators • Build drill-through and filter-enabled dashboards • Implement dynamic reporting features • Generate business performance dashboards • Publish reports to Power BI Service
Unit 3 Lab Assignment: Exploratory Data Analysis and Workflow Automation using KNIME	CO3
Problem Statement	Build workflow-driven analytical systems using KNIME to perform exploratory data

	analysis and automate analytical tasks.	
Sub-Problems	• Create workflows using KNIME nodes • Perform filtering, aggregation, and data manipulation • Conduct exploratory data analysis using visualization nodes • Automate repetitive analytical operations • Apply statistical analysis techniques • Generate workflow-based analytical summaries • Export processed analytical reports	
Unit 4 Lab Assignment: Predictive Analytics and Business Insight Generation		CO4
Problem Statement	Implement predictive analytics models using Power BI and KNIME to generate data-driven business insights and forecasting reports.	
Sub-Problems	• Perform trend analysis and forecasting in Power BI • Build classification models using Decision Trees • Implement clustering using k-Means algorithm • Evaluate model performance using accuracy metrics • Generate prediction-based business reports • Analyze business scenarios using predictive analytics • Create automated insight generation dashboards	
Capstone Lab Assignment: Integrated Business Intelligence and Predictive Analytics System		CO1, CO2, CO3, CO4
Problem Statement	Design and develop a complete business intelligence and predictive analytics platform integrating Power BI dashboards, KNIME workflows, data preprocessing, and machine learning models for real-world decision-making applications.	

Sub-Problems	• Import and integrate multi-source datasets • Perform data cleaning and transformation operations • Develop interactive dashboards and KPI reports • Create workflow-based automated analytics using KNIME • Implement predictive analytics models for forecasting and classification • Generate business insights and analytical summaries • Publish and present analytical reports for decision-making • Design reusable workflows and reporting modules • Evaluate analytical outcomes using real-world datasets • Build an end-to-end data-driven decision support system

Text Books

- Microsoft Power BI Data Analyst Certification Guide, Daniil Maslyuk, 2nd Edition, Packt Publishing, 2024.
- KNIME Beginner's Luck, Rosaria Silipo, 2nd Edition, KNIME Press, 2022.
- C. Ott and F. Villwock, *KNIME Beginner's Luck: A Guide to KNIME Data Analytics Platform*, KNIME Press, 2021.
- Adam Aspin, *Pro Power BI Desktop*, 2nd Edition, Apress, 2019.

Recommended Reference Books

1. Brett Powell, *Mastering Microsoft Power BI*, 2nd Edition, Packt Publishing, 2021. ISBN: 9781801811484
2. Greg Deckler, *DAX Cookbook*, Packt Publishing, 2020. ISBN: 9781839217074

3. John Foreman, *Data Smart: Using Data Science to Transform Information into Insight*, Wiley, 2013. ISBN: 9781118661468
4. Michael Berthold, Nicolas Cebron, Fabian Dill, Thomas Gabriel, Tobias Kötter, Thorsten Meinl and Peter Ohl, *KNIME: The Konstanz Information Miner*, Springer, 2009. ISBN: 9783540782480
5. Foster Provost and Tom Fawcett, *Data Science for Business*, O'Reilly Media, 2013. ISBN: 9781449361327

Recommended Online Resources

- Microsoft Power BI Documentation
- Microsoft Learn – Power BI Learning Paths
- KNIME Analytics Platform Documentation
- KNIME Hub Workflows and Examples
- Kaggle Datasets and Analytics Resources
- GeeksforGeeks Data Analytics Tutorials
- Coursera and edX Business Intelligence Courses
- SQLBI DAX Learning Resources
- Tableau and Data Visualization Community Blogs

LINUX SYSTEM & SHELL SCRIPTING LAB

Program Name	**MCA AI ML**	**Semester**	**I**
Course Name:	**Course Code**	**L-T-P**	**Credits**
Linux System & Shell Scripting Lab	SECSE002	0-0-4	2
Type of Course:	SEC		
Pre-requisite(s): Basic understanding of operating systems, file handling concepts, and introductory programming knowledge.			

Course Description:

This laboratory course introduces students to Linux operating system fundamentals and shell scripting through practical, hands-on activities. The course emphasizes command-line operations, file management, shell programming, automation, process handling, and text-processing utilities used in real-world software development and system administration environments.

Students will learn Linux commands, bash scripting, file permissions, process management, cron jobs, automation utilities, and administrative scripting through guided laboratory exercises and mini-projects. The course focuses on developing practical scripting skills required for DevOps, backend development, cloud environments, cybersecurity operations, and Linux-based automation tasks.

The laboratory component emphasizes real-world scripting, command-line productivity, automation workflows, and system-level problem solving using Linux utilities and shell scripting tools.

The Course Outcomes (COs).

On completion of the course the participants will be:

CO	Course Outcome	Bloom's Level
CO1	Apply Linux commands and file system operations for administrative tasks.	L3 – Apply
CO2	Develop shell scripts using control structures, variables, and functions.	L3 – Apply
CO3	Automate system and text-processing tasks using Linux utilities and shell scripting.	L3 – Apply
CO4	Analyze and optimize shell scripts for system automation and monitoring.	L4 – Analyze

CO–PO Articulation Matrix

COs \ POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	2	2	1	2	–	–	1	1	–	2
CO2	3	3	3	2	2	1	1	2	2	1	2
CO3	3	3	3	2	3	1	1	2	2	2	2
CO4	2	3	3	3	3	1	1	1	2	1	3

CO–PSO Articulation Matrix

COs / PSOs	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	2	2	1	1
CO2	3	3	2	2	2
CO3	3	3	3	2	3
CO4	2	3	3	3	3

Laboratory Experiments – CO Mapping Table

S. No.	Experiment Title	Mapped CO(s)	Bloom's Level
1	Linux File System & Command Practice	CO1	L3 – Apply
2	User, Permission & Process Management	CO1	L3 – Apply
3	Basic Shell Script Development	CO2	L3 – Apply
4	Conditional & Loop-Based Automation Scripts	CO2	L3 – Apply
5	Text Processing Utilities Implementation	CO3	L3 – Apply
6	Cron Jobs & Backup Automation Scripts	CO3	L3 – Apply
7	System Monitoring & Log Analysis Scripts	CO4	L4 – Analyze
8	Capstone: Linux Automation & Administration Toolkit	CO1, CO2, CO3, CO4	L4 – Analyze

Lab Assignment 1: Linux File System & Command Practice	CO1
Problem Statement	Perform Linux command-line operations for managing files, directories, users, and permissions.
Sub-Problems	• Create and organize directory structures • Perform file copy, move, rename, and delete operations • Configure file permissions and ownership • Use pipes and redirection commands • Create user accounts and groups • Generate command-line activity reports
Lab Assignment 2: Shell Script Development	CO2
Problem	Develop shell scripts for automating common Linux tasks and implementing logical

Statement	operations.	
Sub-Problems	• Create scripts using variables and inputs • Implement conditional statements • Develop loop-based automation scripts • Use functions and modular scripting • Process command-line arguments • Debug shell scripts using tracing techniques	
Lab Assignment 3: Text Processing & Automation System		CO3
Problem Statement	Build shell scripts that automate text processing and system administration tasks.	
Sub-Problems	• Process text files using grep, sed, and awk • Generate sorted and filtered reports • Schedule automated tasks using cron • Create backup automation scripts • Monitor running processes and services • Generate log analysis summaries	
Lab Assignment 4: Linux Monitoring & Automation Toolkit		CO4
Problem Statement	Develop a Linux-based automation toolkit for monitoring and managing system resources.	
Sub-Problems	• Develop menu-driven shell applications • Monitor CPU, memory, and disk usage • Implement network connectivity scripts	

	• Generate automated system reports • Handle script errors and exceptions • Optimize scripts for performance and reusability	
Capstone Lab Assignment: Linux System Administration & Automation Toolkit		CO1, CO2, CO3, CO4
Problem Statement	Design and develop a Linux-based administration toolkit that automates system management and monitoring operations.	
Sub-Problems	• Automate file backup and restoration • Monitor system performance metrics • Generate scheduled administrative reports • Analyze logs and detect system anomalies • Create reusable modular shell utilities • Implement process and service monitoring • Automate cleanup and maintenance tasks • Build a menu-driven administrative dashboard	

Text Books

- Linux Command Line and Shell Scripting Bible, Richard Blum & Christine Bresnahan, 4th Edition, Wiley, 2020.ISBN: 9781119700918
- Classic Shell Scripting, Arnold Robbins & Nelson Beebe, O'Reilly Media, 2020. ISBN: 9780596005950
- The Linux Command Line, William Shotts, 2nd Edition, No Starch Press, 2019. ISBN: 9781593279523

Recommended Reference Books

- UNIX and Linux System Administration Handbook, Evi Nemeth et al., 5th Edition, Pearson, 2017 ISBN: 9780134277559
- Bash Guide for Beginners, Machtelt Garrels, GNU Documentation.
- Linux Pocket Guide, Daniel Barrett, 3rd Edition, O'Reilly Media, 2016. ISBN: 9781491927571

Recommended Online Resources

- Linux Documentation Project
- GNU Bash Documentation
- Ubuntu Linux Tutorials
- Linux Journey
- OverTheWire Bandit Linux Practice
- Shell Scripting Tutorial by TutorialsPoint

PROFESSIONAL & WORKPLACE COMMUNICATION

Program Name	MCA AI ML	Semester	I
Course Name:	**Course Code**	**L-T-P**	**Credits**
Professional & Workplace Communication		3-0-0	3
Type of Course:	AEC		

Course Description:

This course focuses on developing practical social, professional, and workplace communication skills through hands-on activities, presentations, digital content creation, collaborative discussions, and real-world communication scenarios. The course emphasizes communication as an employability and professional growth skill rather than a theoretical subject.

Students will learn interpersonal communication, professional writing, digital communication, presentation delivery, storytelling, persuasion, networking, workplace interaction, and interview communication through application-based and portfolio-driven learning. The course integrates speaking, writing, listening, collaboration, and digital communication practices using real-world projects and communication simulations.

By the end of the course, students will develop a professional communication portfolio including presentations, digital profiles, written communication samples, professional content, and recorded speaking activities that demonstrate workplace readiness and communication confidence.

The Course Outcomes (COs).

On completion of the course the participants will be:

CO	Course Outcome	Bloom's Level
CO1	Apply interpersonal communication and collaboration skills in professional situations.	L3 – Apply
CO2	Create professional digital and written communication content.	L3 – Apply
CO3	Deliver presentations, pitches, and persuasive talks effectively.	L3 – Apply
CO4	Demonstrate professional communication skills in workplace interactions and interviews.	L4 – Analyze

CO–PO Articulation Matrix

COs \ POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	1	2	1	1	1	2	2	3	3	2	2
CO2	1	2	2	1	3	1	2	2	3	2	2
CO3	1	2	2	1	2	2	2	3	3	2	2
CO4	1	3	2	2	2	2	3	3	3	3	3

CO–PSO Articulation Matrix

COs \ PSOs	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	2	3	2	1	1
CO2	2	3	2	1	2

CO3	2	3	3	2	2
CO4	2	3	3	2	2

Module-wise Syllabus

Module 1: The Communicator's Toolkit
This module focuses on foundational interpersonal communication skills required for social, academic, and workplace interactions. Students learn active listening, emotional awareness, conflict handling, feedback mechanisms, and culturally sensitive communication practices through practical activities and collaborative discussions. **Topics Covered** • Communication psychology and communication barriers • Active listening and empathetic communication • Non-verbal communication and body language • Emotional awareness and emotional intelligence • Difficult conversations and conflict resolution • Cultural communication and diversity awareness • Giving and receiving constructive feedback **Practical Activities** • Listening and observation exercises • Communication role-plays • Conflict resolution scenarios

- Cross-cultural interaction activities
- Peer feedback circles
- Emotional intelligence reflection tasks

Signature Projects

- Listening pattern analysis report
- Conflict resolution case study
- Cultural communication interview
- Structured peer feedback challenge

Module 2: Digital Communication Mastery

This module develops professional writing and digital communication skills essential for modern workplaces. Students create professional emails, reports, LinkedIn profiles, blogs, portfolios, and business communication documents using practical applications and digital platforms.

Topics Covered

- Professional email writing and etiquette
- Business writing frameworks and formal communication
- Report writing and professional documentation
- LinkedIn profile optimization and networking
- Personal branding and professional identity
- Digital content creation and storytelling
- Professional blogging and online publishing
- Portfolio design and presentation

Practical Activities

- Email drafting and response simulations

- Resume and cover letter creation
- LinkedIn profile development
- Blog writing and publishing
- Business proposal writing
- Digital portfolio development

Signature Projects

- LinkedIn profile makeover
- Professional blog series
- Email crisis management simulation
- Business proposal competition
- Digital portfolio creation

Module 3: The Art of Influence

This module helps students develop public speaking, storytelling, presentation, and persuasive communication abilities. Students learn how to engage audiences, structure presentations, deliver impactful talks, and communicate confidently in formal and informal settings.

Topics Covered

- Storytelling techniques and narrative structures
- TED-style speaking and presentation strategies
- Slide design and visual communication
- Voice modulation and speaking presence
- Impromptu speaking techniques
- Persuasive communication strategies
- Audience engagement methods
- Question and answer handling techniques

Practical Activities

- Presentation analysis and redesign
- Impromptu speaking rounds
- Storytelling workshops
- Pitch presentations
- Podcast recording activities
- Public speaking practice sessions

Signature Projects

- TEDx-style recorded talk
- Startup pitch presentation
- Data storytelling project
- Podcast episode creation
- Impromptu speaking competition

Module 4: Professional Excellence

This module focuses on workplace communication, networking, interviews, leadership communication, and professional interaction skills required for career growth and organizational success.

Topics Covered

- Professional presence and workplace etiquette
- Networking and relationship building
- Meeting communication and coordination
- Interview communication and response structuring
- Team communication and collaboration
- Leadership communication techniques
- Global workplace communication practices

- Career conversations and professional growth discussions

Practical Activities

- Networking simulations
- Mock interviews
- Group discussion practice
- Meeting management exercises
- Team collaboration challenges
- Elevator pitch activities

Signature Projects

- Professional video resume
- Mock interview marathon
- Networking event simulation
- Team challenge project
- Professional brand audit

Practical Activities – CO Mapping Table

S. N	Activity Title	Mapped CO(s)	Bloom's Level
1	Interpersonal Communication & Listening Activities	CO1	L3 – Apply
2	Digital Writing & Professional Content Creation	CO2	L3 – Apply
3	Presentation, Storytelling & Public Speaking Tasks	CO3	L3 – Apply
4	Interview, Networking & Workplace Communication Practice	CO4	L4 – Analyze
5	Professional Communication Portfolio Project	CO1, CO2, CO3, CO4	L4 – Analyze

Major Portfolio Components

- LinkedIn Profile
- Professional Resume & Cover Letters
- Business Writing Samples
- Blog Posts & Professional Content
- Presentation Decks
- TED-style Recorded Talk
- Podcast Episode
- Elevator Pitch Video
- Professional Video Resume
- Interview Performance Recordings

Assessment & Evaluation Framework

Component	**Weightage**
Weekly Activities & Deliverables	25%
Module Signature Projects	30%
Mid-Term Practical Evaluation	15%
Final Professional Portfolio	20%
Participation & Peer Learning	10%

Text Books

- Dale Carnegie, *How to Win Friends and Influence People*, Simon & Schuster, 2022.
- Carmine Gallo, *Talk Like TED*, St. Martin's Press, 2014.

Recommended Reference Books

- Ann Handley, *Everybody Writes*, Wiley, 2014.

- Chris Voss and Tahl Raz, *Never Split the Difference*, HarperBusiness, 2016.
- William Zinsser, *On Writing Well*, Harper Perennial, 2006.

Recommended Online Resources

- TED Talks
- LinkedIn Learning
- Coursera Business Communication Courses
- Harvard Business Review Communication Resources
- Toastmasters International

K.R. MANGALAM UNIVERSITY

Semester: II

MODERN JAVA PROGRAMMING AND APPLICATION DEVELOPMENT

Program Name	**MCA AI ML**	**Semester**	**II**
Course Name:	**Course Code**	**L-T-P**	**Credits**
Modern Java Programming and Application Development	CSE605	3-0-2	4
Type of Course:	Major		
Pre-requisite(s): Basic understanding of programming concepts such as variables, data types, operators, control structures, functions, arrays, and problem-solving techniques. Familiarity with object-oriented programming fundamentals and prior exposure to any programming language such as C, C++, or Python will be beneficial.			

Course Description:

This course introduces students to programming fundamentals and problem-solving techniques using Java with emphasis on modern Java 17 features and object-oriented application development. The course is designed for first-semester MCA learners to build computational thinking, logical reasoning, structured programming abilities, and software development practices through hands-on implementation.

Students learn core Java syntax, object-oriented programming, exception handling, collections, file handling, multithreading, functional programming concepts, and introductory database connectivity using modern Java practices. The course also incorporates Java 17 enhancements including switch expressions, text blocks, records, sealed classes, pattern matching fundamentals, and modular programming concepts. The course emphasizes practical implementation through mini applications, real-world programming problems, console-based systems, and foundational enterprise development practices. By the end of the course, students will be able to design, develop, debug, and maintain Java applications using structured and object-oriented programming principles.

The Course Outcomes (COs).

On completion of the course the participants will be:

CO	Course Outcome	Bloom's Level
CO1	Apply Java programming fundamentals to solve computational and logical problems.	L3 – Apply
CO2	Develop modular and reusable Java programs using object-oriented programming and collections.	L3 – Apply
CO3	Build robust Java applications using exception handling, file handling, multithreading, and database connectivity.	L3 – Apply
CO4	Analyze and develop modern Java applications using functional programming concepts and Java 17 features.	L4 – Analyze

CO–PO Articulation Matrix

COs \ POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	2	2	1	2	1	1	1	1	–	2
CO2	3	3	3	2	3	1	1	2	2	1	2
CO3	3	3	3	3	3	1	1	2	2	2	2
CO4	3	3	3	3	3	1	1	2	2	2	3

CO–PSO Articulation Matrix

COs \ PSOs	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	2	1	–	1

CO2	3	3	2	1	2
CO3	3	3	2	2	3
CO4	2	3	3	2	2

Course Outline:

Unit Number: 1	**Title: Java Fundamentals & Problem-Solving Basics**	**No. of hours: 12**

Content:

- **Introduction to Java ecosystem:** history of Java, JVM, JDK and JRE architecture, Java compilation and execution process, installation and IDE setup, introduction to IntelliJ IDEA/Eclipse and JShell
- **Core Java syntax & data types:** variables, identifiers, keywords, primitive and non-primitive data types, type conversion and casting, operators and expressions, user input using Scanner class, formatted output
- **Control flow statements:** conditional statements, loops, nested loops, switch expressions in Java 17, break, continue and enhanced switch syntax
- **Arrays & basic problem solving**: one-dimensional and two-dimensional arrays, traversal techniques, searching and sorting basics, pattern-based problems and logical programming
- **Introduction to computational thinking:** decomposition, abstraction, algorithmic thinking, dry run techniques and structured problem-solving
- **Java 17 introductory features:** text blocks, enhanced switch expressions and basic introduction to records

Unit Number: 2	**Title: Object-Oriented Programming, Methods & Collections**	**No. of hours: 11**

Content:

- **Methods & modular programming:** method definition, parameters and arguments, return types, method overloading, recursion and scope of variables
- **Object-oriented programming concepts:** classes and objects, constructors, encapsulation, inheritance,

polymorphism, abstraction and interfaces

- **Advanced OOP concepts:** abstract classes, dynamic method dispatch, super keyword, final keyword and package creation
- **String handling & wrapper classes:** String, StringBuilder, StringBuffer, wrapper classes and utility methods
- **Collections framework:** ArrayList, LinkedList, HashSet, TreeSet, HashMap and iteration techniques using iterators and enhanced for-loop
- **Java coding patterns:** duplicate detection, frequency counting, data grouping and collection-based processing
- **Introduction to Java 17 records and sealed classes:** immutable data models, controlled inheritance and modern class design

Unit Number: 3	Title: Exception Handling, File Handling & Database Connectivity	No. of hours: 12

Content:

- **Exception handling:** try-catch-finally blocks, multiple catch statements, throw and throws keywords, custom exceptions and best practices for robust programming
- **File handling & streams:** file reading and writing, buffered streams, character streams, serialization basics and object persistence
- **Java I/O and NIO basics:** Paths, Files class and basic file operations using Java NIO
- **Multithreading fundamentals:** thread lifecycle, thread creation using Thread class and Runnable interface, synchronization basics and thread communication concepts
- **Database connectivity using JDBC:** introduction to databases, JDBC architecture, drivers, database connections, PreparedStatement and ResultSet handling
- **Debugging & code quality:** debugging techniques, logical tracing, exception debugging and clean coding practices

• **Modular application development:** package structuring, reusable utility classes and project organization		
Unit Number: 4	**Title: Functional Programming, Streams & Modern Java Applications**	**No. of hours: 8**

Content:

- **Functional programming concepts:** lambda expressions, functional interfaces and method references
- **Stream API:** stream creation, filtering, mapping, sorting, reduction operations and collectors
- **Optional class & modern Java practices:** handling null values, immutable collections and clean functional coding
- **Java 17 modern features:** pattern matching for instanceof, sealed classes overview, records in enterprise applications and switch expression enhancements
- **Date and time API:** LocalDate, LocalTime, LocalDateTime and formatting utilities
- **Introduction to modular programming:** Java modules, module-info.java and modular application concepts
- **Basic GUI introduction:** overview of JavaFX/Swing components and event-driven programming fundamentals

Laboratory Experiments – CO Mapping Table

S. N	Experiment Title	Mapped CO(s)	Bloom's Level
1	Smart Utility & Expense Calculator System	CO1	L3 – Apply
2	Student & Inventory Management System using OOP	CO2	L3 – Apply
3	Banking & Employee Record Management System	CO3	L3 – Apply
4	Functional Data Processing & Stream Analytics System	CO4	L4 – Analyze
5	Capstone: Integrated Personal Management System	CO1, CO2, CO3, CO4	L4 – Analyze

Unit 1 Lab Assignment: Smart Utility & Expense Calculator System	CO1
Problem Statement	Design a console-based Java application that helps users perform daily utility calculations and expense analysis using core Java programming concepts.
Sub-Problems	• Accept user inputs using Scanner class • Perform arithmetic and logical operations • Implement conditional and menu-driven systems • Generate formatted reports and summaries • Create number-based and pattern-based utilities • Implement array traversal and calculations • Develop mini calculator and expense tracking modules
Unit 2 Lab Assignment: Student & Inventory Management System using OOP	CO2
Problem Statement	Develop an object-oriented Java application to manage student and inventory records using classes, objects and collections.
Sub-Problems	• Create classes with attributes and methods • Implement constructors and encapsulation • Apply inheritance and polymorphism • Store and process records using ArrayList and HashMap • Search, sort and filter records • Generate summaries and reports • Implement reusable utility methods and packages
Unit 3 Lab Assignment: Banking & Employee Record Management System	CO3
Problem Statement	Build a Java application that manages banking or employee records using exception

	handling, file handling, multithreading and JDBC connectivity.
Sub-Problems	• Implement account or employee management features • Store records using files and serialization • Handle runtime errors using exception handling • Connect Java applications with databases using JDBC • Execute CRUD operations using PreparedStatement • Implement multithreading for parallel processing tasks • Organize applications into reusable modules
Unit 4 Lab Assignment: Functional Data Processing & Stream Analytics System	CO4
Problem Statement	Create a modern Java application that processes and analyzes datasets using Stream API and functional programming concepts.
Sub-Problems	• Process collections using Stream API • Apply filtering, mapping and reduction operations • Use lambda expressions and method references • Implement Optional class for null-safe operations • Generate analytical summaries and reports • Develop modular utility-based systems • Apply Java 17 modern programming features
Capstone Lab Assignment: Integrated Personal Management System	CO1, CO2, CO3, CO4
Problem Statement	Design an integrated Java-based management system that combines object-oriented programming, database connectivity, file handling and modern Java features to solve

	real-world problems.
Sub-Problems	• Develop user authentication and record management modules • Implement object-oriented system design • Store and retrieve records using JDBC and files • Apply exception handling and validation mechanisms • Use collections and Stream API for analytics • Generate reports and summaries • Create reusable modules and utility classes • Implement modern Java 17 features in application design

Text Book

- Cay S. Horstmann, "Core Java Volume I – Fundamentals," 13th Edition, Pearson, 2025.
- Herbert Schildt, "Java: The Complete Reference," 13th Edition, McGraw Hill, 2022.

Recommended Reference Books

- Joshua Bloch, "Effective Java," 3rd Edition, Addison-Wesley, 2018. ISBN: 9780134685991
- Raoul-Gabriel Urma, Mario Fusco and Alan Mycroft, "Modern Java in Action," 2nd Edition, Manning Publications, 2022. ISBN: 9781617293566
- Kathy Sierra and Bert Bates, "Head First Java," 3rd Edition, O'Reilly Media, 2022. ISBN: 9781491910771

Recommended Online Resources

- Oracle Java Documentation
- OpenJDK Documentation

MACHINE LEARNING: FOUNDATIONS TO ADVANCED MODELS

Program Name	**MCA AI ML**	**Semester**	**II**
Course Name:	**Course Code**	**L-T-P**	**Credits**
Machine Learning: Foundations to Advanced Models	AML602	3-0-2	4
Type of Course:	DSE/SPC		
Pre-requisite(s): Basic knowledge of Python programming, statistics, linear algebra, probability, and data analysis concepts.			

Course Description:

This course provides a comprehensive understanding of machine learning concepts ranging from foundational algorithms to advanced predictive models. The course focuses on supervised and unsupervised learning, model optimization, deep learning fundamentals, feature engineering, and real-world AI applications. Students gain practical exposure to machine learning pipelines, model training, evaluation, deployment, and scalable intelligent systems using modern ML frameworks and tools. The course emphasizes industry-oriented applications in business analytics, healthcare, finance, recommendation systems, and intelligent automation.

The Course Outcomes (COs).

On completion of the course the participants will be:

COs	**Statements**	**Bloom's Level**
CO1	Explain fundamental machine learning concepts, workflows, and mathematical foundations for intelligent systems.	L2 – Understand
CO2	Apply supervised and unsupervised learning algorithms for classification, prediction, and clustering problems.	L3 – Apply

CO3	Analyze machine learning models using feature engineering, optimization, and performance evaluation techniques.	L4 – Analyze
CO4	Evaluate advanced machine learning and deep learning models for scalable real-world AI applications.	L5 – Evaluate

CO–PO Articulation Matrix

COs \ POs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**
CO1	3	2	1	1	2	–	–	–	1	1	2
CO2	3	3	3	2	3	–	–	1	1	1	2
CO3	3	3	2	3	3	1	–	1	2	2	2
CO4	3	3	3	3	3	1	1	1	2	2	3

CO–PSO Articulation Matrix

COs / PSOs	**PSO1**	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	3	2	1	1	1
CO2	3	3	2	2	3
CO3	3	3	3	2	2
CO4	3	3	3	3	3

Course Outline:

Unit Number: 1	Title: Foundations of Machine Learning	No. of hours: 12

Content:

- **Introduction to Machine Learning:** AI, ML, and deep learning ecosystems
- **Machine Learning Workflow:** data collection, preprocessing, training, and testing
- **Types of Learning:** supervised, unsupervised, semi-supervised, and reinforcement learning
- **Mathematical Foundations:** probability, statistics, linear algebra, and optimization basics
- **Data Preprocessing:** handling missing data, normalization, encoding, and scaling
- **Python ML Ecosystem:** NumPy, Pandas, Scikit-learn, and Jupyter Notebook
- **Bias-Variance Tradeoff:** underfitting and overfitting concepts

Unit Number: 2	Title: Supervised & Unsupervised Learning Models	No. of hours: 11

Content:

- **Regression Models:** linear regression, polynomial regression, and logistic regression
- **Classification Algorithms:** decision trees, random forests, naïve Bayes, and SVM
- **Clustering Techniques:** k-means, hierarchical clustering, and DBSCAN
- **Association Rule Mining:** Apriori algorithm and market basket analysis
- **Dimensionality Reduction:** PCA and feature reduction methods
- **Model Training & Testing:** train-test split and cross-validation techniques
- **Ensemble Learning:** bagging, boosting, and stacking approaches

Unit Number: 3	Title: Feature Engineering, Model Optimization & Evaluation	No. of hours: 12

Content:

- **Feature Engineering:** feature extraction, transformation, and selection methods
- **Handling Imbalanced Data:** oversampling, under sampling, and SMOTE techniques

- **Hyperparameter Optimization:** grid search and random search methods
- **Performance Metrics:** accuracy, precision, recall, F1-score, ROC-AUC
- **Model Evaluation Techniques:** confusion matrix and validation strategies
- **Pipeline Automation:** preprocessing and ML workflow pipelines
- **Explainable AI (XAI):** interpretability and fairness in machine learning

Unit Number: 4	**Title: Advanced Machine Learning & Deep Learning Applications**	**No. of hours: 10**

Content:

- **Neural Network Fundamentals:** perceptron, activation functions, and backpropagation
- **Deep Learning Models:** CNNs, RNNs, and transformers overview
- **Natural Language Processing Basics:** text preprocessing and embeddings
- **Recommendation Systems:** collaborative and content-based filtering
- **Computer Vision Applications:** image classification and object detection basics
- **Model Deployment Concepts:** APIs, cloud deployment, and edge AI basics
- **Emerging Trends:** AutoML, generative AI, and intelligent autonomous systems

Laboratory Experiments – CO Mapping Table

S. N	Experiment Title	Mapped CO(s)	Bloom's Level
1	Machine Learning Workflow & Data Preprocessing Lab	CO1	L2 – Understand
2	Supervised & Unsupervised Model Development	CO2	L3 – Apply
3	Feature Engineering & Model Optimization Framework	CO3	L4 – Analyze
4	Advanced ML & Deep Learning Application System	CO4	L5 – Evaluate
5	Capstone: Intelligent End-to-End Machine Learning Platform	CO1, CO2, CO3, CO4	L5 – Evaluate

Unit 1 Lab Assignment: Machine Learning Workflow & Data Preprocessing Lab		CO1
Problem Statement	Design machine learning workflows integrating data preprocessing, transformation, and exploratory analysis for intelligent analytics applications.	
Sub-Problems	• Import and preprocess datasets • Handle missing and noisy data • Apply normalization and encoding techniques • Perform exploratory data analysis (EDA) • Create ML-ready datasets • Visualize data distributions and trends • Generate preprocessing and analysis reports	
Unit 2 Lab Assignment: Supervised & Unsupervised Model Development		CO2
Problem Statement	Develop supervised and unsupervised learning models for prediction, classification, and clustering applications using machine learning frameworks.	
Sub-Problems	• Implement regression models • Build classification algorithms • Perform clustering and segmentation analysis • Apply dimensionality reduction techniques • Train and test ML models • Compare model performance metrics • Generate prediction and clustering reports	
Unit 3 Lab Assignment: Feature Engineering & Model Optimization Framework		CO3
Problem Statement	Develop feature engineering and optimization systems for improving machine learning	

	model accuracy and interpretability.	
Sub-Problems	• Perform feature extraction and transformation • Apply feature selection techniques • Handle imbalanced datasets using sampling methods • Optimize models using hyperparameter tuning • Build ML preprocessing pipelines • Analyze model explainability and fairness • Generate optimization and evaluation reports	
Unit 4 Lab Assignment: Advanced ML & Deep Learning Application System		CO4
Problem Statement	Design advanced AI applications integrating deep learning, NLP, computer vision, and recommendation systems for intelligent automation.	
Sub-Problems	• Implement CNN and RNN architectures • Perform text preprocessing and sentiment analysis • Develop recommendation systems • Apply image classification techniques • Deploy ML models using APIs or cloud services • Generate AI application performance reports	
Capstone Lab Assignment: Intelligent End-to-End Machine Learning Platform		CO1, CO2, CO3, CO4
Problem Statement	Design and develop a scalable intelligent machine learning platform integrating data preprocessing, feature engineering, predictive analytics, deep learning, visualization, and deployment pipelines for real-world applications.	
Sub-Problems	• Design enterprise ML architecture • Implement automated preprocessing pipelines	

	• Develop supervised and unsupervised learning systems • Integrate feature engineering and optimization modules • Create interactive analytics dashboards and reports • Implement deep learning and AI applications • Deploy ML models using cloud or edge platforms • Apply explainable AI and fairness mechanisms • Optimize scalability and model performance • Present a complete intelligent machine learning solution

Text Book

- Aurélien Géron, *Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow*, 3rd Edition, O'Reilly Media, 2022.
- Tom M. Mitchell, *Machine Learning*, McGraw-Hill Education, 2017.

Recommended Reference Books

- Christopher M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006. ISBN: 9780387310732
- Ian Goodfellow, Yoshua Bengio and Aaron Courville, *Deep Learning*, MIT Press, 2016. ISBN: 9780262035613
- Sebastian Raschka and Vahid Mirjalili, *Python Machine Learning*, 3rd Edition, Packt Publishing, 2019. ISBN: 9781789955750
- Trevor Hastie, Robert Tibshirani and Jerome Friedman, *The Elements of Statistical Learning*, 2nd Edition, Springer, 2009. ISBN: 9780387848570

- Andreas C. Müller and Sarah Guido, *Introduction to Machine Learning with Python*, O'Reilly Media, 2016. ISBN: 9781449369415

Recommended Online Resources

- Scikit-learn Documentation
- TensorFlow Documentation
- PyTorch Documentation
- Kaggle Machine Learning Tutorials
- Google Machine Learning Crash Course
- AWS Machine Learning Learning Path
- Microsoft Learn – AI & ML Fundamentals
- Coursera Machine Learning Specializations
- GeeksforGeeks Machine Learning Tutorials
- TutorialsPoint Artificial Intelligence & ML Tutorials

ADVANCED OPERATING SYSTEM ENGINEERING

Program Name	**MCA AI ML**	**Semester**	**II**
Course Name:	**Course Code**	**L-T-P**	**Credits**
Advanced Operating System Engineering	CSE607	3-0-2	4
Type of Course:	Major		
Pre-requisite(s): Basic knowledge of operating systems, computer architecture, process management, memory management, and computer networks.			

Course Description:

This course provides an advanced understanding of modern operating system concepts, architectures, and engineering practices used in contemporary computing environments. The course focuses on process and thread management, synchronization, distributed operating systems, virtualization, cloud infrastructure, containerization, security, and real-time systems. Students gain practical exposure to Linux-based system programming, process scheduling, resource management, virtualization platforms, and container orchestration technologies. The course emphasizes real-world operating system engineering practices required for scalable, reliable, and secure computing systems.

The Course Outcomes (COs).

On completion of the course the participants will be:

COs	**Course Outcome**	**Bloom's Level**
CO1	Apply advanced operating system concepts for process management, scheduling, synchronization, and resource allocation.	L3 – Apply
CO2	Develop concurrent and distributed system solutions using multithreading, IPC, virtualization, and containerization techniques.	L3 – Apply

CO3	Build secure and reliable operating system environments using memory management, file systems, and security mechanisms.	L3 – Apply
CO4	Analyze and optimize modern operating system architectures for cloud, real-time, and high-performance computing environments.	L4 – Analyze

CO–PO Articulation Matrix

COs \ POs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**
CO1	3	3	2	1	2	–	–	–	1	1	2
CO2	3	2	3	2	3	–	–	1	2	1	2
CO3	3	3	2	2	3	1	1	2	1	1	2
CO4	3	3	2	3	3	2	2	1	2	2	3

CO–PO Articulation Matrix

COs / PSOs	**PSO1**	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	3	2	1	–	1
CO2	3	3	2	1	2
CO3	3	3	2	2	3
CO4	2	3	3	2	2

Course Outline:

Unit Number: 1	Title: Advanced Process & Resource Management	No. of hours: 12

Content:

- **Advanced Operating System Architecture:** monolithic, layered, microkernel, and hybrid architectures
- **Process Management:** process states, process control block, context switching, and process scheduling
- **CPU Scheduling Algorithms:** priority scheduling, multilevel queue scheduling, round robin, and real-time scheduling
- **Thread Management:** multithreading models, thread pools, and lightweight processes
- **Process Synchronization:** semaphores, monitors, mutex locks, and critical section problems
- **Deadlock Handling:** prevention, avoidance, detection, and recovery techniques
- **Resource Allocation:** resource allocation graphs and banker's algorithm

Unit Number: 2	Title: Distributed Systems, Virtualization & Containers	No. of hours: 11

Content:

- **Distributed Operating Systems:** distributed architectures and communication models
- **Inter-Process Communication (IPC):** pipes, message queues, shared memory, and sockets
- **Remote Procedure Calls:** client-server communication and distributed coordination
- **Virtualization Technologies:** hypervisors, virtual machines, and hardware virtualization
- **Containerization:** Docker architecture, container lifecycle, and image management
- **Container Orchestration:** Kubernetes basics and container deployment models
- **Cloud Computing Support:** operating system support for cloud infrastructure
- **Fault Tolerance & Reliability:** distributed recovery and high availability concepts

Unit Number: 3	Title: Memory Management, File Systems & Security	No. of hours: 11

Content:

- **Advanced Memory Management:** paging, segmentation, virtual memory, and demand paging
- **Memory Allocation Techniques:** page replacement algorithms and thrashing control
- **File System Architecture:** file organization, directory structures, and journaling file systems
- **Disk Scheduling Algorithms:** FCFS, SSTF, SCAN, and C-SCAN scheduling techniques
- **Operating System Security:** authentication, authorization, and access control mechanisms
- **Linux Security Concepts:** permissions, user management, and process security
- **Malware & Threat Protection:** intrusion detection and secure system practices
- **Backup & Recovery:** system backup strategies and disaster recovery concepts

Unit Number: 4	**Title: Real-Time, Cloud & High-Performance Operating Systems**	**No. of hours: 11**

Content:

- **Real-Time Operating Systems (RTOS):** hard and soft real-time systems and scheduling
- **Embedded Operating Systems:** embedded Linux and IoT operating systems
- **Cloud Operating Systems:** distributed cloud resource management and virtualization support
- **High-Performance Computing:** multiprocessing and parallel processing concepts
- **GPU & Multicore Processing:** operating system support for parallel architectures
- **Performance Monitoring & Optimization:** benchmarking, profiling, and tuning techniques
- **Energy-Efficient Computing:** green operating system concepts and power management
- **Modern OS Trends:** edge computing, serverless systems, and autonomous infrastructure

Laboratory Experiments – CO Mapping Table

S. N	Experiment Title	Mapped CO(s)	Bloom's Level
1	Process Scheduling & Synchronization Simulator	CO1	L3 – Apply
2	Distributed System & Container Management Platform	CO2	L3 – Apply

3	Secure File System & Memory Management System	CO3	L3 – Apply
4	Cloud & Real-Time Operating System Performance Analyzer	CO4	L4 – Analyze
5	Capstone: Intelligent Cloud-based Operating System Environment	CO1, CO2, CO3, CO4	L4 – Analyze

Unit 1 Lab Assignment: Process Scheduling & Synchronization Simulator	CO1
Problem Statement	Design a Linux-based operating system simulator to demonstrate advanced process scheduling, synchronization, and resource management concepts.
Sub-Problems	• Implement CPU scheduling algorithms • Simulate process creation and termination • Develop synchronization using semaphores and mutexes • Implement deadlock detection and prevention mechanisms • Simulate thread creation and scheduling • Analyze resource allocation techniques • Generate scheduling and performance reports
Unit 2 Lab Assignment: Distributed System & Container Management Platform	CO2
Problem Statement	Develop distributed and containerized system environments using virtualization, IPC, Docker, and Kubernetes technologies.
Sub-Problems	• Implement IPC mechanisms using pipes and shared memory • Develop socket-based distributed communication systems • Configure virtual machines and hypervisors • Create and deploy Docker containers

	• Manage containerized applications using Kubernetes • Simulate distributed fault tolerance mechanisms • Generate deployment and orchestration reports	
Unit 3 Lab Assignment: Secure File System & Memory Management System		CO3
Problem Statement	Build secure operating system modules for memory management, file systems, access control, and system security.	
Sub-Problems	• Implement page replacement algorithms • Simulate virtual memory management techniques • Develop file handling and directory management modules • Implement disk scheduling algorithms • Configure Linux user permissions and security policies • Analyze system vulnerabilities and security mechanisms • Generate memory and security analysis reports	
Unit 4 Lab Assignment: Cloud & Real-Time Operating System Performance Analyzer		CO4
Problem Statement	Design operating system performance monitoring tools for real-time, cloud-based, and high-performance computing environments.	
Sub-Problems	• Configure real-time scheduling environments • Monitor system processes and resource utilization • Analyze multicore and parallel processing performance • Implement cloud-based workload management systems • Benchmark operating system performance metrics	

	• Optimize energy-efficient system configurations • Generate performance tuning and optimization reports	
Capstone Lab Assignment: Intelligent Cloud-based Operating System Environment		CO1, CO2, CO3, CO4
Problem Statement	Design and develop a scalable intelligent operating system environment integrating process management, virtualization, cloud orchestration, system security, and high-performance computing concepts for enterprise applications.	
Sub-Problems	• Design distributed operating system architecture • Implement process and thread management modules • Integrate virtualization and container orchestration systems • Develop secure memory and file management systems • Implement monitoring and performance optimization tools • Configure cloud-native and distributed computing environments • Apply fault tolerance and recovery mechanisms • Generate system analytics and administrative dashboards • Deploy scalable operating system services using containers • Present a complete enterprise operating system engineering solution	

Text Books

- Abraham Silberschatz, Peter B. Galvin and Greg Gagne, *Operating System Concepts*, 11th Edition, Wiley, 2024.
- Andrew S. Tanenbaum and Herbert Bos, *Modern Operating Systems*, 5th Edition, Pearson, 2022.

Recommended Reference Books

- William Stallings, *Operating Systems: Internals and Design Principles*, 10th Edition, Pearson, 2018.

ISBN: 9780134670959

- Thomas Anderson and Michael Dahlin, *Operating Systems: Principles and Practice*, 2nd Edition, Recursive Books, 2014. ISBN: 9780985673529
- Brendan Burns, *Designing Distributed Systems*, O'Reilly Media, 2018. ISBN: 9781491983645
- Nigel Poulton, *The Kubernetes Book*, Packt Publishing, 2022. ISBN: 9781803247359
- Jason Cannon, *Linux for Beginners*, Independently Published, 2019. ISBN: 9781795306171

Recommended Online Resources

- Linux Kernel Documentation
- Docker Documentation
- Kubernetes Documentation
- Red Hat System Administration Resources
- Oracle Linux Documentation
- Ubuntu Server Documentation
- Microsoft Learn – Operating System Fundamentals
- GeeksforGeeks Operating System Tutorials
- TutorialsPoint Linux & OS Tutorials
- IBM Cloud Operating System Resources

ADVANCED COMPUTER NETWORK

Program Name	**MCA AI ML**	**Semester**	**II**
Course Name:	**Course Code**	**L-T-P**	**Credits**
Advanced Computer Network	CSE608	3-0-2	4
Type of Course:	Major		
Pre-requisite(s): Basic knowledge of computer networks, network protocols, IP addressing, routing concepts, and operating systems.			

Course Description:

This course provides advanced knowledge of modern computer networking concepts, architectures, and communication technologies used in enterprise, cloud, and distributed computing environments. The course focuses on routing protocols, network virtualization, software-defined networking, wireless and mobile networks, network security, cloud networking, and high-performance communication systems. Students gain practical exposure to network configuration, traffic analysis, routing simulation, cloud-based networking, and network security tools required for modern networking infrastructure management.

The Course Outcomes (COs).

On completion of the course the participants will be:

COs	**Statements**	**Bloom's Level**
CO1	Applying advanced routing, switching, and network management techniques for scalable communication systems.	L3 – Apply
CO2	Designing secure and reliable network infrastructures using modern networking protocols and architectures.	L4 – Analyze
CO3	Implementing cloud, wireless, and software-defined networking solutions for	L3 – Apply

	distributed computing environments.	
CO4	Evaluating network performance, security, and fault tolerance in enterprise and high-performance network systems.	L4 – Analyze

CO–PO Articulation Matrix

COs \ POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	3	2	1	3	–	–	–	1	1	2
CO2	3	3	3	2	3	1	–	2	2	1	2
CO3	3	2	3	2	3	1	1	1	2	1	2
CO4	3	3	2	3	3	2	1	2	2	2	3

CO–PO Articulation Matrix

COs / PSOs	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	3	2	1	2
CO2	3	3	3	2	2
CO3	3	3	2	2	3
CO4	3	3	3	3	2

Course Outline:

Unit Number: 1	Title: Advanced Routing and Switching Technologies	No. of hours: 12
Content: • **Advanced Network Architectures:** enterprise, data centre, and distributed network architectures		

- **Routing Protocols:** RIP, OSPF, EIGRP, and BGP routing mechanisms
- **Switching Technologies:** VLANs, trunking, STP, and multilayer switching
- **IPv6 Networking:** IPv6 addressing, routing, and transition techniques
- **Traffic Engineering:** load balancing and quality of service (QoS) concepts
- **Network Address Translation:** NAT and PAT implementation strategies
- **Network Monitoring Tools:** packet analysis and traffic monitoring techniques

Unit Number: 2	Title: Network Security and Reliable Communication	No. of hours: 11

Content:

- **Network Security Fundamentals:** threats, vulnerabilities, and attack models
- **Firewall Technologies:** packet filtering, proxy firewalls, and next-generation firewalls
- **Virtual Private Networks (VPNs):** IPSec and SSL VPN configurations
- **Secure Communication Protocols:** HTTPS, SSH, TLS, and secure socket communication
- **Intrusion Detection & Prevention Systems:** IDS and IPS concepts
- **Authentication & Access Control:** AAA, RADIUS, and TACACS+ mechanisms
- **Reliable Communication Mechanisms:** error detection, congestion control, and fault tolerance
- **Wireless Network Security:** WPA2, WPA3, and secure wireless configurations

Unit Number: 3	Title: Cloud, Wireless, and Software-Defined Networking	No. of hours: 11

Content:

- **Cloud Networking Fundamentals:** virtual networks and cloud infrastructure connectivity
- **Software-Defined Networking (SDN):** SDN architecture, controllers, and OpenFlow concepts
- **Network Function Virtualization (NFV):** virtualized networking services
- **Wireless and Mobile Networks:** Wi-Fi standards, LTE, 5G, and mobile IP concepts

- **IoT Networking:** communication protocols for smart devices and sensor networks
- **Edge and Fog Computing:** distributed networking models for IoT environments
- **Container Networking**: Docker and Kubernetes networking basics
- **Cloud Security & Scalability:** secure cloud communication and distributed services

Unit Number: 4	Title: Network Performance, Virtualization, and High-Performance Networks	No. of hours: 11

Content:

- **Network Virtualization:** virtual LANs, virtual switches, and overlay networks
- **High-Performance Networking:** data centre networks and high-speed communication systems
- **Traffic Analysis & Optimization:** bandwidth management and latency reduction techniques
- **Fault Tolerance & Recovery:** redundancy, failover, and resilient networking
- **Network Automation:** scripting and automated network configuration management
- **Performance Evaluation:** throughput, delay, jitter, and packet loss analysis
- **Emerging Networking Trends:** AI-driven networking, intent-based networking, and smart networks
- **Green Networking:** energy-efficient communication systems and sustainable networking

Laboratory Experiments – CO Mapping Table

S. N	Experiment Title	Mapped CO(s)	Bloom's Level
1	Advanced Routing & Switching Configuration Lab	CO1	L3 – Apply
2	Secure Network Infrastructure & VPN Deployment	CO2	L4 – Analyze
3	SDN, Cloud & Wireless Network Implementation	CO3	L3 – Apply
4	Network Performance & Virtualization Analyzer	CO4	L4 – Analyze
5	Capstone: Intelligent Enterprise Network Infrastructure	CO1, CO2, CO3, CO4	L4 – Analyze

Unit 1 Lab Assignment: Advanced Routing & Switching Configuration Lab		CO1
Problem Statement	Design and configure enterprise-level routing and switching infrastructures using advanced routing protocols and network management tools.	
Sub-Problems	• Configure VLANs and trunking mechanisms • Implement OSPF and EIGRP routing protocols • Configure BGP for inter-network communication • Implement IPv6 addressing and routing • Analyze routing tables and packet forwarding • Perform QoS and traffic prioritization • Generate network topology and routing reports	
Unit 2 Lab Assignment: Secure Network Infrastructure & VPN Deployment		CO2
Problem Statement	Develop secure communication infrastructures using firewalls, VPNs, and intrusion detection mechanisms for enterprise networks.	
Sub-Problems	• Configure firewall security rules • Implement IPSec and SSL VPN connections • Deploy IDS and IPS systems • Configure secure communication using SSH and HTTPS • Implement AAA authentication services • Analyze network vulnerabilities and threats • Generate security audit and compliance reports	
Unit 3 Lab Assignment: SDN, Cloud & Wireless Network Implementation		CO3
Problem	Build cloud-enabled and software-defined networking environments integrating wireless	

Statement	communication and virtualized networking services.	
Sub-Problems	• Configure SDN controllers and OpenFlow switches • Implement virtualized network functions • Simulate wireless and mobile communication systems • Configure cloud virtual networking environments • Implement Docker and Kubernetes networking • Analyze IoT communication protocols • Generate SDN and cloud network deployment reports	
Unit 4 Lab Assignment: Network Performance & Virtualization Analyzer		CO4
Problem Statement	Design and analyze high-performance virtualized networks for optimized communication, fault tolerance, and network scalability.	
Sub-Problems	• Configure network virtualization environments • Analyze bandwidth and latency metrics • Implement failover and redundancy mechanisms • Automate network configurations using scripts • Monitor traffic and communication performance • Optimize data center network architectures • Generate performance analysis and optimization reports	
Capstone Lab Assignment: Intelligent Enterprise Network Infrastructure		CO1, CO2, CO3, CO4
Problem Statement	Design and develop a scalable intelligent enterprise networking solution integrating secure communication, SDN, virtualization, wireless technologies, and cloud networking for	

	modern distributed organizations.
Sub-Problems	• Design enterprise network architecture • Configure advanced routing and switching services • Implement secure VPN and firewall infrastructures • Deploy SDN-enabled cloud networking solutions • Integrate wireless and IoT communication systems • Implement network automation and monitoring tools • Optimize network performance and fault tolerance • Generate network analytics dashboards and reports • Apply security auditing and compliance mechanisms • Present a complete enterprise networking solution

Text Books

- Behrouz A. Forouzan, *Data Communications and Networking*, 6th Edition, McGraw-Hill Education, 2022.
- Andrew S. Tanenbaum and David J. Wetherall, *Computer Networks*, 6th Edition, Pearson Education, 2022.

Recommended Reference Books

- William Stallings, *Data and Computer Communications*, 11th Edition, Pearson, 2019. ISBN: 9780135165614
- Larry L. Peterson and Bruce S. Davie, *Computer Networks: A Systems Approach*, 6th Edition, Morgan Kaufmann, 2021. ISBN: 9780128182000
- Douglas Comer, *Internetworking with TCP/IP Volume 1*, 6th Edition, Pearson, 2014. ISBN:

9789332555365

- Feamster and Rexford, *Computer Networking: A Top-Down Approach*, 8th Edition, Pearson, 2021. ISBN: 9780136681557
- Thomas D. Nadeau and Ken Gray, *SDN: Software Defined Networks*, O'Reilly Media, 2013. ISBN: 9781491907415

Recommended Online Resources

- Cisco Networking Academy
- Wireshark Documentation
- Kubernetes Networking Documentation
- Docker Networking Documentation
- Open Networking Foundation (ONF) Resources
- Oracle Cloud Networking Documentation
- AWS Networking Fundamentals
- GeeksforGeeks Computer Network Tutorials
- Tutorials Point Networking Tutorials
- Microsoft Learn – Network Fundamentals

AI FLUENCY & DIGITAL INTELLIGENCE

Program Name	**MCA AI ML**	**Semester**	**II**
Course Name:	**Course Code**	**L-T-P**	**Credits**
AI Fluency & Digital Intelligence	SEAML001	2-0-2	3
Type of Course:	SEC		
Pre-requisite(s): Basic understanding of computer applications, internet technologies, digital communication tools, and fundamental programming or computational concepts. Familiarity with data handling, online collaboration platforms, and emerging digital technologies will be beneficial.			

Course Description:

This course introduces students to the practical and responsible use of Artificial Intelligence tools across academic, professional, and real-world workflows. The course focuses on developing AI fluency — the ability to effectively interact, collaborate, evaluate, and create using modern AI systems such as generative AI, large language models (LLMs), multimodal AI tools, and AI-powered productivity platforms.

Students will learn structured prompting, AI-assisted research, content creation, data analysis, automation workflows, and responsible AI practices through hands-on activities and interdisciplinary projects. The course also develops critical thinking skills required to evaluate AI outputs, identify hallucinations and biases, and make informed decisions about when and how AI should be used.

The course emphasizes experiential learning through AI labs, collaborative exercises, verification tasks, portfolio development, and capstone projects relevant to students' academic domains and career pathways.

The Course Outcomes (COs).

On completion of the course the participants will be:

CO	Course Outcome	Bloom's Level
CO1	Explain AI fundamentals and AI interaction models.	L2 – Understand
CO2	Apply structured prompting and AI collaboration techniques.	L3 – Apply
CO3	Use AI tools for research, writing and productivity tasks.	L3 – Apply
CO4	Analyze AI outputs and delegation decisions critically.	L4 – Analyze
CO5	Evaluate responsible and ethical AI practices.	L5 – Evaluate
CO6	Develop AI-assisted solutions for domain-specific problems.	L3 – Apply

CO–PO Articulation Matrix

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

CO–PSO Articulation Matrix

COs / PSOs	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	2	2	1	–	1
CO2	2	3	2	1	2

CO3	2	3	2	1	3
CO4	3	3	3	2	2
CO5	2	2	3	3	1
CO6	3	3	2	2	3

Course Outline:

Unit Number: 1	**Title: AI Foundations & Human-AI Interaction**	**No. of hours: 8**

Content:

- **Introduction to Artificial Intelligence:** AI, machine learning, deep learning and generative AI
- **How Large Language Models Work:** tokens, prompts, prediction and conversational AI basics
- **AI Ecosystem & Modern AI Tools:** ChatGPT, Claude, Gemini and multimodal AI platforms
- **Human-AI Interaction Models:** automation, augmentation and AI agency concepts
- **Applications of AI Across Disciplines**
- **AI Safety & Academic Integrity:** privacy, bias, misinformation and responsible AI usage
- **Understanding AI Limitations & Hallucinations**

Unit Number: 2	**Title: Structured Prompting & AI-Assisted Workflows**	**No. of hours: 8**

Content:

- **Anatomy of Effective Prompts:** role, task, context, constraints and output format
- **Structured Prompting Techniques:** iterative prompting and refinement strategies
- **Collaboration Framing & Meta Prompting**
- **AI for Research & Information Gathering**
- **AI-Assisted Writing & Summarization**

- **Verification of AI-Generated Content & Citations**
- **AI for Learning, Productivity & Communication**

Unit Number: 3	Title: AI-Augmented Creation & Productivity	No. of hours: 8

Content:

- **AI for Data Analysis & Visualization**
- **AI for Presentations & Content Creation**
- **Multimodal AI:** image, audio and video generation tools
- **AI Coding Assistants & Automation Tools**
- **No-Code & Low-Code AI Platforms**
- **AI for Workflow Automation & Productivity**
- **Evaluating AI-Generated Artifacts & Outputs**

Unit Number: 4	Title: Critical Evaluation, Ethics & AI Applications	No. of hours: 6

Content:

- **Critical Evaluation of AI Outputs:** hallucinations, reasoning errors and factual verification
- **Delegation Decisions:** when to use AI and when human effort is required
- **Responsible AI Practices:** transparency, attribution, IP ownership and privacy governance
- **AI & Society:** workforce impact, ethical concerns and policy perspectives
- **AI Applications in Different Domains & Industries**
- **AI Portfolio & Professional Development**
- **Capstone Development Using AI Tools**
- **Presentation & Demonstration of AI-Assisted Solutions**

Laboratory Experiments – CO Mapping Table

S. No.	Experiment Title	Mapped CO(s)	Bloom's Level
1	Exploring AI Tools & Interaction Models	CO1	L2 – Understand
2	Structured Prompting & Prompt Optimization	CO2	L3 – Apply
3	AI-Assisted Research & Writing Workflow	CO2, CO3	L3 – Apply
4	AI-Based Data Analysis & Visualization	CO3	L3 – Apply
5	Multimodal Content Creation Using AI	CO3	L3 – Apply
6	AI Coding & Automation Applications	CO3, CO6	L3 – Apply
7	AI Output Verification & Ethical Analysis	CO4, CO5	L4 – Analyze
8	Capstone: AI-Assisted Domain-Specific Solution	CO1, CO2, CO3, CO4, CO5, CO6	L4 – Analyze

Unit 1 Lab Assignment: AI Foundations & Exploration		CO1
Problem Statement	Explore modern AI platforms and analyze different AI interaction models and applications.	
Sub-Problems	• Compare outputs from multiple AI tools • Analyze automation, augmentation and AI agency examples • Identify strengths and limitations of AI systems • Explore AI applications in different domains • Analyze hallucinations and inaccurate outputs • Document AI usage observations and findings	
Unit 2 Lab Assignment: Structured Prompting & AI Workflows		CO2
Problem Statement	Develop effective prompting strategies and AI-assisted workflows for research and	

	productivity tasks.	
Sub-Problems	• Design structured prompts with constraints and roles • Perform iterative prompt refinement • Generate research summaries using AI tools • Verify AI-generated citations and content • Create AI-assisted writing workflows • Compare prompt effectiveness and output quality	
Unit 3 Lab Assignment: AI-Augmented Creation & Automation		CO2, CO3
Problem Statement	Use AI tools for content generation, automation and productivity enhancement.	
Sub-Problems	• Analyze datasets using AI tools • Generate presentations and visual content using AI • Explore image, audio and video AI tools • Use AI coding assistants for development tasks • Build simple automation workflows using no-code tools • Evaluate AI-generated artifacts critically	
Unit 4 Lab Assignment: Ethical AI & Domain Applications		CO3, CO6
Problem Statement	Analyze responsible AI practices and develop domain-specific AI-assisted solutions.	
Sub-Problems	• Evaluate AI outputs for factual correctness and bias • Analyze ethical and privacy-related AI scenarios • Identify appropriate AI delegation decisions • Explore AI applications in specific disciplines • Develop AI-assisted professional portfolio components	

	• Present responsible AI usage strategies	
Capstone Lab Assignment: AI-Assisted Domain-Specific Solution		CO1, CO2, CO3, CO4, CO5, CO6
Problem Statement	Develop an AI-assisted solution for a real-world domain problem using responsible AI practices and structured workflows.	
Sub-Problems	• Identify real-world problem and workflow requirements • Select suitable AI tools and prompting strategies • Build AI-assisted solution prototype • Evaluate outputs and verify correctness • Document ethical considerations and AI usage decisions • Present final project with demonstration and reflection	

Text Book

- Ethan Mollick, *Co-Intelligence: Living and Working with AI*, Portfolio/Penguin, 2024.
- UNESCO, *AI Competency Framework for Students*, 2024.
- Jules White et al., *Prompt Engineering for Generative AI*, Vanderbilt University Resources.

Recommended Reference Books

- Bernard Marr, *Artificial Intelligence in Practice*, Wiley.
- Russel & Norvig, *Artificial Intelligence: A Modern Approach*, Pearson Education.
- David Shapiro, *Generative AI for Everyone*, O'Reilly Media.

Recommended Online Resources

- OpenAI — https://platform.openai.com/docs

- Anthropic Claude — https://docs.anthropic.com/
- Google Gemini — https://deepmind.google/technologies/gemini/
- Hugging Face — https://huggingface.co/
- DeepLearning.AI — https://www.deeplearning.ai/
- Prompt Engineering Guide — https://www.promptingguide.ai/
- Google AI Studio — https://aistudio.google.com/
- Microsoft Learn AI — https://learn.microsoft.com/ai/

Minor Project

Program Name	MCA AI ML	Semester	II
Course Name:	**Course Code**	**L-T-P**	**Credits**
Minor Project	PRAML051	0-0-4	2
Type of Course:	Proj		

Course Description

The course is designed to provide students with an opportunity to apply theoretical knowledge, technical competencies, and problem-solving abilities to the development of a comprehensive solution for a real-world problem. The course emphasizes innovation, research orientation, design thinking, and interdisciplinary learning through the execution of individual or team-based projects under faculty mentorship. Carrying academic credit as prescribed in the curriculum, the course serves as an essential component for enhancing practical knowledge, technical proficiency, teamwork, and professional preparedness.

The course focuses on experiential and outcome-based learning by engaging students in activities such as problem identification, requirement analysis, literature review, system design, implementation, testing, documentation, and presentation. Students are encouraged to explore emerging technologies, industry-oriented challenges, research problems, entrepreneurial ideas, or socially relevant applications while developing their projects.

Through the project work, students gain hands-on exposure to project planning, technical documentation, collaborative development, research methodologies, and professional ethics. The course also strengthens communication skills, analytical thinking, creativity, and decision-making abilities while fostering

independent learning and innovation. By the end of the project, students are expected to demonstrate the successful integration of domain knowledge and technical skills through the development and presentation of a functional prototype, software solution, research outcome, or engineering model aligned with real-world applications and professional standards.

The Course Outcomes (COs).

On completion of the course the participants will learn to:

CO	Course Outcome	Bloom's Level
CO1	Analyze and define a real-world problem statement by applying domain knowledge, research understanding, and requirement analysis techniques.	L4 – Analyze
CO2	Design and develop an appropriate project solution using suitable tools, technologies, methodologies, and engineering practices.	L3 – Apply
CO3	Evaluate the performance, functionality, and effectiveness of the developed project through testing, validation, and result analysis.	L5 – Evaluate
CO4	Prepare technical documentation and present the project outcomes effectively while demonstrating teamwork, communication skills, and professional ethics.	L3 – Apply

CO–PO Articulation Matrix

COs \ POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	3	2	3	2	1	1	1	2	1	3
CO2	3	3	3	2	3	1	1	2	2	2	3
CO3	3	3	3	3	3	2	1	2	2	2	3
CO4	1	2	2	1	2	2	3	3	3	3	2

CO–PSO Articulation Matrix

COs \ PSOs	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	2	3	2	1
CO2	3	3	2	2	3
CO3	2	2	3	3	2
CO4	2	2	1	2	2

1. Objectives

All student projects must:

- Employ industry-relevant technologies and tools.

- Reflect a well-defined development or research process from ideation to final deployment or publication.
- Promote innovation and real-world problem-solving.
- Deliver meaningful outcomes such as a functional application, published research, or validated prototype.

2. Project Scope and Categories

Projects must fall under one of the following approved categories:

A. Research-Oriented Projects

- Must address futuristic domains like Generative AI (e.g., GANs, Transformers, LLMs), cybersecurity, quantum computing, etc.
- **Multidisciplinary Research Integration:** Projects are encouraged to bridge multiple disciplines by combining core technology domains (e.g., AI, cybersecurity, quantum computing) with fields such as **Healthcare, Finance, Law & Ethics, Education, Social Sciences, or Environmental Studies.**
- Require mandatory publication of a research paper in a reputed journal or international conference before final submission.
- Supervised by **internal faculty mentor.**

B. Industry-Based Projects

- Should be based on real-world industry challenges with practical applications.
- Must demonstrate full implementation, deployment, and usability.
- Students will work under **external mentorship** from an industry along with a faculty mentor from SOET.
- External Industry **Mentorship Confirmation certificate which will** confirms that industry expert/mentor will be guiding the student(s) on their project work. The certificate must be **issued and signed by the external mentor**. It should be printed on the **official letterhead of the company/organization** (if available).
- A certificate of completion from the associated organization is mandatory.

C. University-Focused Interdisciplinary Projects / Community Solution Based Projects

- Projects under this category should aim to identify and address real-world challenges, enhancement opportunities, or operational gaps within K.R. Mangalam University.
- Students are encouraged to collaborate across disciplines, integrating knowledge from diverse domains to develop innovative, practical, and sustainable solutions for campus improvement.
- Projects must demonstrate a clear understanding of institutional needs and propose **scalable or implementable models** that can be piloted or adopted by university stakeholders.

Real-World Problem Orientation (for Community Solution Based Projects):

- Projects should address practical issues such as health, sanitation, education, environmental sustainability, digital literacy, social inclusion, transportation, waste management, or livelihood enhancement.

D. Start-Up Projects

- Projects under this category should focus on identifying real-world problems and developing innovative, technology-driven solutions with the potential for commercialization.
- Students are encouraged to think entrepreneurially, transforming their ideas into viable products or services by building functional prototypes or Minimum Viable Products (MVPs).
- Projects must include key elements of a start-up framework such as market research, user validation, business model design, and go-to-market strategy.
- The outcome should demonstrate not only technical feasibility but also market relevance, with the potential to seek incubation or funding support from the university's innovation/start-up ecosystem.

E. Deep-Tech & Systems (Learning-by-Replication)

- This category emphasizes deep technical mastery by replicating foundational technologies that power modern computing. Students build simplified versions of complex systems to understand

how they truly work "under the hood," going beyond API usage or high-level application development.

- Define a Clear Scope
  - The system replica should have a minimum viable core (e.g., for a compiler: lexer + parser + code-gen)
  - extensions or optimizations (indexes, caching, scheduling algorithms).

Examples of "Build Your Own X":

- Build your own database engine (query parser, storage engine, index manager)
- Build your own compiler (lexer, parser, AST, code generator)
- Build your own search engine (crawler, indexer, ranking algorithms)
- Build your own machine learning framework (autograd engine, tensor ops)
- Build your own virtual machine, interpreter, or runtime environment

F. Hardware & Software Integration Using IoT / Embedded Systems

This category focuses on building solutions that combine physical hardware (sensors, actuators, microcontrollers) with software systems (apps, cloud platforms, analytics engines). Students must design end-to-end solutions that sense, process, communicate, and act on real-world data.

- Projects without actual hardware implementation will not be accepted. A fully functional hardware prototype is mandatory, including working sensors/actuators, embedded programming, connectivity, and a software interface.

3. Technology Stack and Industry Tools

- Students must select a technology stack aligned with current industry standards, such as MERN, MEAN, Django, Flutter, or cloud-native architectures (AWS, Azure, GCP).
- Projects must incorporate industry-relevant tools and frameworks for version control (Git/GitHub), containerization (Docker), CI/CD (GitHub Actions, Jenkins), and agile development practices.
- Use of modern development environments, APIs, and deployment platforms is strongly encouraged to ensure scalability, maintainability, and real-world applicability of the solution.

4. Formation of Groups

4.1 Group Size & Composition

- Optimal Group Size: Groups will consist of 2 students to allow effective collaboration while maintaining manageable group dynamics.

- Skill Diversity: Form groups with a mix of skill sets (e.g., coding, design, research, communication, etc.) to enhance collaboration. Students should be encouraged to select group members based on complementary skills.

4.2 Group Formation Process

- Self-selection: Allow students to form their own groups, encouraging them to choose teammates based on project interests.
- Pre-formed Groups: Alternatively, groups can be assigned by the coordinator to ensure diversity of skills and ideas across all teams.

4.3 Team Finalization Rule

Once teams are formed and registered, no changes in team composition will be allowed under any circumstances.

4.4 Timeline

Week	Activity/Stage
Week 0	• Onboarding on Projexa (Project Management Tool) • Team formation • Faculty Mentor allotment

Week 1–2	• Problem Statement Finalization • Initial Mentor Interactions and discussion
Week 3–4	• Synopsis Submission • Project Proposal Evaluation and Approval
Week 5, 6 and 7	Implementation Sprint 1: Core Modules, MVP Build, Mentor Interaction
Week 8	Development Phase – Part 1 • Module-wise implementation begins
Week 9–10	Development Phase – Part 2 • Module-wise implementation begins
Week 11–12	Mid Term Evaluation
Week 13–14	• Testing and Debugging • Performance evaluation
Week 15–16	• Final Report Submission • Project Video (1-2 mins) demonstrating working project • Presentation & Viva Preparation

5. Guidelines for Project Selection

For any project category (Research-Based, Development-Based, Start-Up, or University-Focused Interdisciplinary), students may select their project through either of the following approaches:

- **Faculty-Suggested Problem Statements:**
  - Faculty members will provide curated problem statements based on their domain expertise, current research trends, or institutional needs.
  - Students can choose from these problems and discuss with the faculty mentor before finalizing.
- **Student-Proposed Problem Statements:**
  - Student teams are encouraged to identify real-world problems based on personal interests, industry trends, or societal challenges.
  - The proposed problem must be original, relevant, and feasible within the given timeline and resources.
  - Final approval is subject to evaluation by the project review committee.

6. Faculty Allocation Process

- **Faculty-Suggested Problem Statement:**

If a team selects a problem provided by a faculty member, the team will be allocated to that faculty after peer review and approval by the Project Coordinator.

- **Student-Proposed Problem Statement**:

  If the team proposes its own problem statement, then after peer review and validation by the Project Coordinator, an appropriate faculty mentor will be assigned based on domain expertise.

- **Inter School Problem Statement:**

  If a team chooses a problem statement proposed by a faculty member from another school/department, the team will be assigned to that faculty member as their project mentor only after a peer review and approval by the Project Coordinator.

7. Roles and Responsibilities of Faculty Mentors

- **Regular Interactions:**

  Conduct regular meetings with the assigned team(s) through Projexa in online/offline mode. All meeting records, including agenda, minutes, and attendance, must be documented and uploaded on Projexa.

  A minimum of five (5) interactions is mandatory, with at least two (2) interactions per month. If the required interactions are not completed within the specified timeline, the pending interaction slots will be automatically closed and marked as incomplete.

- **Task Monitoring:**

  Assign tasks with clear deadlines and track the progress and completion status for each team through the project cycle.

- **Evaluation After Interaction:**

  After every interaction, the faculty mentor must evaluate the team's progress and update the evaluation/comments directly on Projexa as part of the meeting record.

- **Meeting Confirmation & Rescheduling**:

  Faculty must respond to meeting requests initiated by student teams. If unable to conduct a scheduled meeting, it should be rescheduled within one week.

- **Justification for Meeting Cancellation:**

  Any cancellation of scheduled team meetings by the mentor must be supported with a valid and documented reason.

- **GitHub Activity Monitoring**:

  Regularly monitor each team's GitHub repository to verify individual contributions and maintain performance records for all team members.

8. Evaluation Metrics

Projects will be evaluated in 4 phases (Total 100 Marks):

- Synopsis Presentation (15 Marks)
- Mid Term Presentation (20 Marks)
- Final Term Presentation (40 Marks)
- Project Mentor Interaction (25 Marks)

For each evaluation, marks will be cumulated from the following components:

- Project Mentor Evaluation
- Project Evaluation Committee

Evaluation Stage	**Project Mentor (Interaction)**	**Project Evaluation Committee**
Synopsis Presentation (20)	5	15
Mid Term Presentation (30)	10	20
Final Term Presentation (40)	10	40
Total	**25**	**75**
Total Marks	**(25+75) = 100**	

Project Evaluation Committee Marking Scheme

Evaluation Stage	Criteria	Marks
Synopsis Presentation	Creative/Naive/Real-world Problem	5
	Clarity & Feasibility of Project Objectives	5
	Preparation & Presentation of PPT	5
	Total	**15**
Mid-Term Presentation	Project Efforts – UI/Implementation	10
	Effective Use of Modern Technology Stack	5
	Effective Coding Practices / Use of Git	5
	Total	**20**
Final-Term Presentation	Final Production Demonstration & Completion w.r.t Objectives	20
	Impact & Use Cases of the Project	10
	Deployment	10
	Total	**40**

9. Policy on Missed Presentation Schedules

Students are strictly required to adhere to the assigned schedules for all project presentations, including the Synopsis, Mid-Term, and Final-Term evaluations.

- Only one rescheduling opportunity will be granted in case a team or individual misses their assigned presentation slot. This rescheduling must be approved in advance (where possible) or justified immediately after the missed session with valid reasons.
- A penalty of 5 marks will be deducted for each missed presentation schedule, irrespective of the evaluation stage.
- Failure to appear even after the rescheduled opportunity will result in zero marks being awarded for that evaluation component.

This policy ensures fairness, accountability, and professionalism in the evaluation process.

10. Project Documentation Requirements – Stage-Wise

1. Synopsis Stage

To be submitted before the Synopsis Presentation:

- Synopsis Report
- Synopsis PPT

- Mentorship Confirmation Certificate (must be on industry letterhead, duly signed by the External Mentor)

2. Mid-Term Evaluation Stage

To be submitted during the Mid-Term Presentation:

- Mid Evaluation PPT
- Project Use Case & Deployment Confirmation Certificate
    - issued by the Industry Mentor, confirming project relevance and real-world deployment potential

3. Final-Term Evaluation Stage

To be submitted during or before Final Presentation:

- Final Project Report
- PPT for Final Presentation
- Video Demonstration (1–2 minutes)
    - should clearly document
        - Key features of Project
        - Execution workflow or demonstration of functionality
        - Real-world impact or use-case.

To be submitted in MP4 format or via a shared video link. (as per instructions)

- **Plagiarism Certificate**
    - Must be compulsorily appended to the Final Project Report
    - Should be obtained using institutional or approved plagiarism checking tools.

11. Plagiarism & Ethical Guidelines

- Maximum allowable plagiarism: ≤ 15%
- Check plagiarism using open-source tools and proper citations
- No reuse of previous projects
- Intellectual property rights (if applicable)
- Compulsory Plagiarism Certificate to be appended in the report file

12. Project Outcome Requirements by each Project Category

Project Category	Desired Outcome
Industry-Based Project	▪ Project completion certificate from Industry Mentor ▪ Working prototype/application demonstration
Research-Based Project	▪ Research paper submission in conference/journal

	▪ Technical report with findings and implementation
University-Focused Interdisciplinary Projects / Community Solution Based Projects	▪ Faculty Mentor approval certificate ▪ Functional solution addressing university/community problem
Start-Up Project	• Basic business idea validation or mentorship proof • Functional MVP (Minimum Viable Product) or prototype demonstration
Deep-Tech & Systems (Learning-by-Replication)	▪ Simplified working model/prototype demonstrating core functionality of the selected system.
Hardware & Software Integration Using IoT / Embedded Systems	• Functional hardware prototype with sensor/actuator integration • Basic software interface/dashboard demonstrating system operation.

ADVANCED COMMUNICATION AND INTERVIEW MASTERY

Program Name	MCA AI ML	Semester	II
Course Name:	Course Code	L-T-P	Credits
Advanced Communication and Interview Mastery		3-0-0	3
Type of Course:	AEC		

Course Description:

This course is designed as an advanced, activity-driven communication and employability program that prepares students for contemporary professional environments and placement processes. The course adopts a practice-first pedagogy in which learners engage in real workplace simulations, group discussions, mock interviews, professional writing tasks, and problem-solving challenges before examining the underlying communication concepts. Emphasis is placed on Indian professional contexts, digital communication practices, cross-cultural interaction, leadership communication, and interview preparedness. Through collaboration with Placement Cells, recruiters, alumni, and industry professionals, learners develop authentic workplace communication competence, critical thinking abilities, and recruiter-ready professional portfolios.

By the end of the course, students will be able to communicate effectively and confidently in placement processes, workplace interactions, and professional networking environments. They will demonstrate competence in professional writing, group discussions, interviews, presentations, and digital communication while also developing analytical thinking, creativity, and problem-solving abilities. Learners will create a recruiter-ready professional portfolio containing written communication samples, recorded presentations, interview performances, LinkedIn profiles, and workplace problem-solving projects that showcase authentic professional growth and employability readiness.

The Course Outcomes (COs).

On completion of the course the participants will be:

CO	Course Outcome	Bloom's Level
CO1	Demonstrate confident self-expression, active listening, and emotional intelligence in professional and workplace communication contexts.	L3 – Apply
CO2	Produce effective professional writing including emails, reports, applications, cover letters, and digital communication artefacts using audience-centred strategies.	L3 – Apply
CO3	Perform effectively in presentations, group discussions, interviews, and workplace conversations using structured and persuasive communication techniques.	L3 – Apply
CO4	Build a credible professional identity through digital presence, cross-cultural communication competence, and workplace professionalism.	L4 – Analyze
CO5	Apply analytical, creative, and problem-solving skills to real workplace and organisational challenges and communicate solutions effectively.	L3 – Apply

CO–PO Articulation Matrix

COs \ POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	1	1	1	–	1	2	–	3	3	3	2

CO2	1	2	2	1	2	2	–	3	2	3	2
CO3	1	2	2	1	2	2	–	2	3	3	2
CO4	1	2	2	2	2	3	1	3	2	3	3
CO5	2	3	3	2	2	2	1	2	3	3	3

CO–PSO Articulation Matrix

COs \ PSOs	**PSO1**	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	1	2	1	1	1
CO2	1	2	2	1	2
CO3	1	2	2	1	2
CO4	1	2	2	2	2

Course Outline:

Unit Number: 1	**Title: Confident Self-Expression and Professional Identity**	**No. of hours: 11**

Content:

Communication in Indian Professional Contexts

- Communication styles in Indian workplaces
- Formal and informal communication patterns

- Hierarchy, tone, and relational communication

Self-Expression and Professional Identity

- Building confidence in workplace interactions
- Introducing oneself in professional contexts
- Authenticity and personal communication style

Communication Barriers

- Language hierarchies and regional influences
- Code-switching and generational communication gaps
- Miscommunication in workplace environments

Active Listening

- Listening for understanding and interpretation
- Clarifying, summarising, and questioning
- Listening in interviews and group tasks

Emotional Intelligence in the Workplace

- Self-awareness and emotional regulation
- Empathy and workplace relationships
- Handling conflict and difficult conversations

Suggested Activities

- Recorded professional introductions
- Workplace communication simulations
- Listening and summarizing exercises
- Role-plays on workplace conflict scenarios

Unit Number: 2	Title: Professional Writing in Workplace Contexts	No. of hours: 11

Content:

Email Writing and Professional Etiquette

- Writing clear and concise emails
- Subject lines, tone, and structure
- Professional greetings and follow-up communication

Business Writing

- Principles of effective workplace writing
- Writing memos, notices, and formal correspondence
- Audience awareness and purpose-driven writing

Report Writing

- Structure and organisation of reports
- Writing observations, findings, and recommendations
- Executive summaries and workplace documentation

Cover Letters and Applications

- Writing job and internship applications
- Presenting authentic professional experiences
- Resume and cover letter integration

AI and Professional Writing

- Ethical use of AI writing tools
- Editing and proofreading strategies
- Maintaining authenticity and contextual relevance

Suggested Activities

- Email rewriting workshops
- Incident and workplace report writing
- Job application and cover letter drafting
- AI-assisted editing and peer review sessions

Unit Number: 3	Title: Digital Presence and Cross-Cultural Communication	No. of hours: 11

Content:

Digital Professional Identity

- LinkedIn profiles and networking
- Professional email communication

- Building an online professional image

Online Communication

- Virtual meeting etiquette
- Communication through Teams, Slack, and WhatsApp
- Managing professional communication online

Cross-Cultural Communication

- Communication across Indian cultural diversity
- High-context and low-context communication
- Global workplace communication practices

Professional Networking

- Building professional relationships
- Writing networking messages and follow-ups
- Communication in professional communities

Portfolio Development

- Curating professional communication artefacts
- Reflective communication statements
- Preparing recruiter-ready portfolios

Suggested Activities

- LinkedIn profile audits
- Virtual meeting simulations
- Cross-cultural communication case studies
- Portfolio presentation workshops

Unit Number: 4	Title: Problem-Solving and Creative Thinking at Work	No. of hours: 12

Content:

Problem Identification and Analysis

- Defining workplace problems
- Root cause analysis techniques
- Problem framing and analytical thinking

Structured Problem-Solving

- PDCA and design thinking approaches
- Applying frameworks to workplace scenarios
- Decision-making and solution development

Creative Thinking and Innovation

- Creativity in professional environments
- Constraint-based innovation and frugal thinking
- Generating and evaluating ideas

Solution Communication

- Presenting solutions persuasively
- Data-supported communication strategies
- Workplace pitching and recommendation delivery

Workplace Readiness and Collaboration

- Team-based problem-solving
- Managing ambiguity and pressure
- Communication in innovation and project teams

Suggested Activities

- Case-study analysis and solution design
- Community and campus problem-solving projects
- Innovation and idea-generation challenges
- Recruiter-evaluated solution presentations

Text Books

- Carnegie, Dale. *How to Win Friends and Influence People*. Simon & Schuster, 2022.
- Gallo, Carmine. *Talk Like TED: The 9 Public-Speaking Secrets of the World's Top Minds*. St. Martin's Press, 2014.
- Patterson, Kerry, et al. *Crucial Conversations: Tools for Talking When Stakes Are High*. McGraw-Hill Education, 2012.
- Rosenberg, Marshall B. *Nonviolent Communication: A Language of Life*. Puddle Dancer Press, 2015.

- Voss, Chris, and Tahl Raz. *Never Split the Difference: Negotiating as If Your Life Depended on It*. Harper Business, 2016.
- Zinsser, William. *On Writing Well: The Classic Guide to Writing Nonfiction*. Harper Perennial, 2006

Recommended Reference Books

- Cuddy, Amy. *Presence: Bringing Your Boldest Self to Your Biggest Challenges*. Little, Brown and Company, 2015.
- Duarte, Nancy. *HBR Guide to Persuasive Presentations*. Harvard Business Review Press, 2012.
- Heath, Chip, and Dan Heath. *Made to Stick*. Random House, 2007.
- Reynolds, Garr. *Presentation Zen*. New Riders, 2011.
- Turkle, Sherry. *Reclaiming Conversation: The Power of Talk in a Digital Age*. Penguin Press, 2015.
- Brown, Brené. *Dare to Lead*. Random House, 2018.

Open Educational Resources:

- TED Talks: Public speaking, storytelling, and leadership communication resources.
- LinkedIn Learning: Professional communication and employability courses.
- Coursera: Business Communication and Interview Skills courses.
- Harvard Business Review Communication Collection.
- BBC Learning English: Workplace communication and speaking activities.
- NPTEL Soft Skills and Personality Development Modules.

K.R. MANGALAM UNIVERSITY

Semester: III

DEEP LEARNING, NLP & GENERATIVE AI

Program Name	**MCA AI ML**	**Semester**	**III**
Course Name:	**Course Code**	**L-T-P**	**Credits**
Deep Learning, NLP & Generative AI	AML701	3-0-2	4
Type of Course:	DSE/SPC		
Pre-requisite(s): Basic programming knowledge and understanding of databases and data handling			

Course Description:

This course introduces students to the foundations and applications of Deep Learning, Natural Language Processing (NLP), and Generative Artificial Intelligence. The course is designed to provide theoretical understanding and practical exposure to modern AI techniques used in intelligent systems, language understanding, and content generation. Students learn neural network architectures, optimization techniques, computer vision basics, sequence modeling, transformer architectures, and large language models using modern AI frameworks and tools.

The course emphasizes practical implementation through AI model development, dataset processing, prompt engineering, and real-world application development. Students explore deep neural networks, convolutional and recurrent architectures, NLP pipelines, embeddings, transformers, generative models, and AI ethics. By the end of the course, students will be able to design, train, evaluate, and deploy intelligent AI-powered applications using modern deep learning ecosystems.

The Course Outcomes (COs).

On completion of the course the participants will be:

CO	Course Outcome	Bloom's Level
CO1	Explain deep learning fundamentals, neural network architectures, and learning mechanisms used in intelligent systems.	L2 – Understand
CO2	Develop deep learning models using modern frameworks for prediction, classification, and computer vision applications.	L3 – Apply
CO3	Analyze and implement NLP techniques and transformer-based architectures for language understanding and text processing tasks.	L4 – Analyze
CO4	Build generative AI applications using large language models, prompt engineering, and modern AI deployment practices.	L3 – Apply

CO–PO Articulation Matrix

COs \ POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	2	1	2	1	1	–	1	–	2	1
CO2	3	3	3	3	2	1	1	1	1	2	2
CO3	3	3	2	3	2	2	1	2	1	3	3
CO4	3	3	3	3	2	2	2	3	2	3	3

CO–PSO Articulation Matrix

COs / PSOs	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	2	1	–	1
CO2	3	3	2	1	3
CO3	3	3	3	2	3
CO4	2	3	3	2	3

Unit Number: 1	Title: Introduction to Deep Learning & Neural Networks	No. of hours: 12

Content:

- **Introduction to Artificial Intelligence, Machine Learning and Deep Learning**: evolution and applications of AI systems, supervised and unsupervised learning concepts, overview of intelligent systems and AI ecosystems
- **Biological neuron vs artificial neuron:** perceptron model, activation functions, forward propagation and learning process
- **Neural network fundamentals:** multilayer perceptrons, loss functions, optimization algorithms, gradient descent and backpropagation
- **Deep learning development ecosystem:** introduction to TensorFlow, Keras and PyTorch, AI model development workflow and GPU-based training concepts
- **Overfitting and regularization techniques:** dropout, batch normalization and model evaluation strategies
- **Introduction to modern AI applications:** recommendation systems, autonomous systems and predictive analytics

Unit Number: 2	Title: Deep Neural Networks & Computer Vision	No. of hours: 12

Content:

- **Feedforward neural networks and deep architectures:** hidden layers, hyperparameter tuning and training strategies
- **Convolutional Neural Networks (CNNs):** convolution operations, pooling layers, feature extraction and image classification concepts
- **Image processing and computer vision basics:** object detection, image recognition and transfer learning
- **Pretrained AI models and transfer learning techniques:** ResNet, VGG and EfficientNet overview
- **Performance evaluation metrics:** accuracy, precision, recall, F1-score and confusion matrix analysis
- **Deep learning deployment basics:** model saving, loading and inference pipeline concepts

Unit Number: 3	Title: Natural Language Processing & Transformers	No. of hours: 11

Content:

- **Introduction to Natural Language Processing:** NLP pipeline, text preprocessing and language understanding fundamentals
- **Text preprocessing techniques:** tokenization, stemming, lemmatization and stop-word removal
- **Text representation models:** Bag of Words, TF-IDF and word embeddings
- **Word embeddings and semantic representation**: Word2Vec, GloVe and contextual embeddings
- **Sequence models:** Recurrent Neural Networks (RNN), LSTM and GRU architectures Attention mechanism and transformer architecture fundamentals

• **Transformer-based NLP models:** BERT, GPT and encoder-decoder architectures		
Unit Number: 4	**Title: Modern Web & Type-Safe Programming using TypeScript**	**No. of hours: 10**
Content: • Introduction to Generative AI and foundation models • **Generative models:** GANs, Variational Autoencoders (VAEs) and diffusion model basics • **Large Language Models (LLMs):** GPT architecture, prompt engineering and AI-assisted content generation • **Prompt engineering techniques:** zero-shot, few-shot and chain-of-thought prompting • Fine-tuning and retrieval-augmented generation (RAG) concepts • Conversational AI and chatbot development fundamentals • **Ethical AI and responsible generative AI practices:** fairness, bias, hallucination and AI governance • **AI model deployment concepts:** APIs, cloud-based AI services and introduction to MLOps		

Laboratory Experiments – CO Mapping Table

S. N	Experiment Title	Mapped CO(s)	Bloom's Level
1	Neural Network & Prediction System	CO1	L2 – Understand
2	Deep Learning & Image Classification System	CO2	L3 – Apply
3	NLP & Transformer-Based Text Analytics System	CO3	L4 – Analyze
4	Generative AI & Conversational Assistant Development	CO4	L3 – Apply

5	Capstone: Intelligent AI-Powered Application Development	CO1, CO2, CO3, CO4	L4 – Analyze

Unit 1 Lab Assignment: Neural Network & Prediction System		CO1
Problem Statement	Design a deep learning application that performs prediction and classification tasks using neural network fundamentals and optimization techniques.	
Sub-Problems	• Install and configure AI development environments • Implement perceptron and multilayer neural network models • Train neural networks using gradient descent and backpropagation • Apply activation and loss functions • Evaluate model performance using validation techniques • Implement regularization techniques for improved accuracy • Generate prediction reports and analytical summaries	
Unit 2 Lab Assignment: Deep Learning & Image Classification System		CO2
Problem Statement	Develop a deep learning application for image classification and computer vision tasks using convolutional neural networks.	
Sub-Problems	• Build CNN architectures for image processing • Perform image preprocessing and augmentation • Train and evaluate image classification models • Apply transfer learning using pretrained models • Analyze model performance using evaluation metrics • Save and deploy trained deep learning models	

	• Develop minicomputer vision applications	
Unit 3 Lab Assignment: NLP & Transformer-Based Text Analytics System		CO3
Problem Statement	Create an NLP-based intelligent system that processes and analyzes textual data using sequence models and transformer architectures.	
Sub-Problems	• Perform text preprocessing and tokenization • Implement Bag of Words and TF-IDF techniques • Apply word embeddings and semantic analysis • Build RNN/LSTM-based sequence models • Implement transformer-based text processing models • Perform sentiment analysis and text classification • Generate analytical text summaries and reports	
Unit 4 Lab Assignment: Generative AI & Conversational Assistant Development		CO3
Problem Statement	Build a generative AI-based conversational assistant using large language models and prompt engineering techniques.	
Sub-Problems	• Design prompt engineering workflows • Implement text generation using pretrained LLMs • Develop conversational AI chatbot modules • Integrate APIs for AI-assisted applications • Apply ethical AI and hallucination reduction techniques • Build retrieval-augmented conversational systems • Deploy AI applications using cloud or local environments	
Capstone Lab Assignment: Intelligent AI-Powered Application Development		CO1, CO2, CO3, CO4

Problem Statement	Design and develop a comprehensive AI-powered system integrating deep learning, NLP and generative AI techniques for solving real-world problems.
Sub-Problems	• Develop intelligent AI-based application modules • Implement deep learning prediction and classification systems • Apply NLP pipelines and transformer architectures • Integrate generative AI and conversational interfaces • Perform model evaluation and optimization • Implement deployment and API integration strategies • Generate reports, analytics and visualization dashboards • Apply responsible AI and ethical AI principles in system design

Text Book

- Generative Deep Learning, David Foster, 2nd Edition, O'Reilly Media, 2022. ISBN: 9781098134181
- Speech and Language Processing, Daniel Jurafsky & James H. Martin, 3rd Edition, Pearson, 2025. ISBN: 9780131873216
- Deep Learning, Ian Goodfellow, Yoshua Bengio & Aaron Courville, MIT Press, 2016. ISBN: 9780262035613
- Natural Language Processing with Python, Steven Bird, Ewan Klein & Edward Loper, O'Reilly Media, 2009. ISBN: 9780596516499

Recommended Reference Books

1. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, Aurélien Géron, 3rd

Edition, O'Reilly Media, 2022. ISBN: 9781098125974

2. Transformers for Natural Language Processing, Denis Rothman, 2nd Edition, Packt Publishing, 2021. ISBN: 9781800565791
3. Deep Learning with Python, François Chollet, 2nd Edition, Manning Publications, 2021. ISBN: 9781617296864
4. Build a Large Language Model (From Scratch), Sebastian Raschka, Manning Publications, 2024. ISBN: 9781633437166
5. Natural Language Processing in Action, Hobson Lane, Cole Howard & Hannes Hapke, Manning Publications, 2019. ISBN: 9781617294631

Recommended Online Resources

- TensorFlow Documentation
- PyTorch Documentation
- Hugging Face Transformers
- OpenAI Platform Documentation
- Keras Documentation GitHub Open Source Projects — https://github.com/explore

AI ENGINEERING, MLOPS & INTELLIGENT SYSTEMS DEVELOPMENT

Program Name	**MCA AI ML**	**Semester**	**III**
Course Name:	**Course Code**	**L-T-P**	**Credits**
AI Engineering, MLOps & Intelligent Systems Development	AML702	3-0-2	4
Type of Course:	DSE/SPC		
Pre-requisite(s): Basic programming knowledge, understanding of machine learning fundamentals, databases, APIs, and software development practices			

Course Description:

This course introduces students to AI engineering practices, MLOps workflows, and intelligent systems development using modern AI technologies and deployment ecosystems. The course focuses on designing, developing, deploying, monitoring, and maintaining scalable AI-powered systems using industry-standard frameworks, cloud services, automation pipelines, and intelligent application architectures.

Students learn model lifecycle management, data pipelines, model training workflows, containerization, CI/CD for AI systems, orchestration platforms, cloud-based AI deployment, monitoring strategies, and responsible AI engineering practices. The course also introduces intelligent agents, AI-driven automation systems, retrieval-based architectures, and production-ready AI applications.

The course emphasizes practical implementation through real-world AI engineering workflows, deployment pipelines, cloud-native AI systems, and intelligent application development projects. By the end of the course, students will be able to design, automate, deploy, monitor, and optimize intelligent AI systems using modern AI engineering and MLOps practices.

The Course Outcomes (COs).

On completion of the course the participants will be:

CO	Course Outcome	Bloom's Level
CO1	Explain AI engineering workflows, intelligent system architectures, and MLOps lifecycle management concepts.	L2 – Understand
CO2	Develop scalable AI pipelines and deploy machine learning models using modern AI engineering tools and cloud platforms.	L3 – Apply
CO3	Analyze intelligent systems, monitoring frameworks, automation pipelines, and AI model performance in production environments.	L4 – Analyze
CO4	Build production-ready intelligent AI applications using containerization, orchestration, APIs, and responsible AI practices.	L3 – Apply

CO–PO Articulation Matrix

COs \ POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	2	1	2	1	1	–	1	–	2	1
CO2	3	3	3	3	2	1	1	1	1	2	2
CO3	3	3	2	3	2	2	1	2	1	3	3

CO4	3	3	3	3	2	2	2	3	2	3	3

CO–PSO Articulation Matrix

COs / PSOs	**PSO1**	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	3	2	1	–	1
CO2	3	3	2	1	3
CO3	3	3	3	2	3
CO4	2	3	3	2	3

Course Outline:

Unit Number: 1	**Title: Introduction to AI Engineering & Intelligent Systems**	**No. of hours: 12**

Content:

- **Introduction to AI Engineering:** AI system lifecycle, intelligent systems overview and AI application ecosystems
- **Fundamentals of MLOps**: machine learning lifecycle, automation workflows and AI engineering practices
- **AI development environments and toolchains:** Python ecosystem, Jupyter, ML frameworks and cloud AI platforms
- **Data engineering basics for AI systems**: data collection, preprocessing, validation and feature pipelines
- **AI model development workflow:** experimentation, versioning and reproducibility concepts

- Introduction to intelligent agents, recommendation systems and AI automation architectures
- **Responsible AI engineering concepts:** fairness, transparency and ethical AI development practices

Unit Number: 2	Title: Model Deployment, Containerization & Cloud AI Systems	No. of hours: 12

Content:

- **AI model deployment fundamentals:** model serialization, inference pipelines and deployment strategies
- REST APIs and AI services using Flask/FastAPI
- **Containerization using Docker:** Docker images, containers and AI application packaging
- Kubernetes fundamentals and orchestration concepts for scalable AI systems
- **Cloud-based AI deployment platforms:** AWS, Azure and Google Cloud AI services overview
- **CI/CD pipelines for machine learning systems:** GitHub Actions and deployment automation concepts
- AI infrastructure scaling, load balancing and deployment optimization strategies

Unit Number: 3	Title: MLOps, Monitoring & Intelligent Automation	No. of hours: 11

Content:

- MLOps lifecycle management and workflow orchestration
- Experiment tracking and model versioning concepts using MLflow and DVC
- Data drift, concept drift and AI monitoring techniques
- AI model evaluation, logging and performance analytics
- Workflow orchestration concepts using Apache Airflow and pipeline automation
- Intelligent automation systems and AI-powered business workflows

• Monitoring production AI systems and debugging intelligent applications • Security and governance in AI systems deployment		
Unit Number: 4	**Title: Intelligent Systems Development & Generative AI Applications**	**No. of hours: 10**
Content: • Intelligent system architectures and AI-driven application development • Retrieval-Augmented Generation (RAG) systems and vector databases • AI agents and autonomous workflow systems • Conversational AI systems and enterprise chatbot architectures • Generative AI integration using APIs and foundation models • AI application optimization and scalable inference techniques • Responsible AI deployment, AI governance and compliance considerations • Future trends in AI engineering, edge AI and autonomous intelligent systems		

Laboratory Experiments – CO Mapping Table

S. N	Experiment Title	Mapped CO(s)	Bloom's Level
1	AI Workflow & Intelligent Pipeline Development	CO1	L2 – Understand
2	AI Model Deployment & Cloud Integration System	CO2	L3 – Apply
3	MLOps Monitoring & Automation Pipeline	CO3	L4 – Analyze

	System		
4	Intelligent AI Application & Agent Development	CO4	L3 – Apply
5	Capstone: Enterprise Intelligent AI System Deployment	CO1, CO2, CO3, CO4	L4 – Analyze

Unit 1 Lab Assignment: AI Workflow & Intelligent Pipeline Development	CO1
Problem Statement	Design an AI engineering workflow that manages intelligent data pipelines and machine learning development processes.
Sub-Problems	• Install and configure AI engineering environments • Create AI data preprocessing pipelines • Implement model experimentation workflows • Apply dataset validation and transformation techniques • Develop intelligent recommendation workflows • Track AI experiments and model configurations • Generate workflow reports and system analytics
Unit 2 Lab Assignment: AI Model Deployment & Cloud Integration System	CO2
Problem Statement	Develop and deploy AI-powered applications using containerization, APIs and cloud-based deployment platforms.
Sub-Problems	• Build AI inference APIs using Flask or FastAPI • Deploy AI models using Docker containers • Implement cloud deployment workflows

	• Configure CI/CD deployment pipelines • Develop scalable AI service architectures • Integrate AI applications with cloud APIs • Monitor deployed AI applications and services
Unit 3 Lab Assignment: MLOps Monitoring & Automation Pipeline System	CO3
Problem Statement	Create an MLOps pipeline that automates AI workflows, monitors deployed systems and manages intelligent automation processes.
Sub-Problems	• Implement model versioning and experiment tracking • Create workflow orchestration pipelines • Monitor AI system performance and drift detection • Automate retraining and deployment workflows • Generate monitoring dashboards and reports • Apply AI governance and security practices • Debug and optimize intelligent AI systems
Unit 4 Lab Assignment: Intelligent AI Application & Agent Development	CO3
Problem Statement	Build intelligent AI-driven applications using generative AI, retrieval systems and autonomous agent architectures.
Sub-Problems	• Develop conversational AI and chatbot systems • Implement retrieval-augmented generation workflows • Build intelligent AI agents and automation systems • Integrate vector databases and semantic search • Develop AI-powered enterprise assistants

	• Apply responsible AI deployment techniques • Optimize AI applications for scalable deployment	
Capstone Lab Assignment: Enterprise Intelligent AI System Deployment		CO1, CO2, CO3, CO4
Problem Statement	Design and deploy a production-ready intelligent AI system integrating AI engineering, MLOps workflows and intelligent automation architectures.	
Sub-Problems	• Develop enterprise-grade AI applications • Implement scalable MLOps pipelines • Deploy intelligent systems using cloud infrastructure • Integrate AI APIs, monitoring and automation systems • Apply model governance and security mechanisms • Generate dashboards, reports and deployment analytics • Optimize intelligent systems for production environments • Implement responsible AI and compliance strategies	

Text Book

- Designing Machine Learning Systems, Chip Huyen, O'Reilly Media, 2022. ISBN: 9781098107963
- Practical MLOps, Noah Gift & Alfredo Deza, O'Reilly Media, 2021. ISBN: 9781098103019
- Machine Learning Engineering, Andriy Burkov, True Positive Inc., 2020. ISBN: 9781999579579
- Building Machine Learning Powered Applications, Emmanuel Ameisen, O'Reilly Media, 2020. ISBN: 9781492045113

Recommended Reference Books

- Machine Learning Design Patterns, Valliappa Lakshmanan, Sara Robinson & Michael Munn, O'Reilly Media, 2020. ISBN: 9781098115784
- Introducing MLOps, Mark Treveil & The MLOps Community, O'Reilly Media, 2023. ISBN: 9781098153014
- Kubernetes Up and Running, Kelsey Hightower, Brendan Burns & Joe Beda, 3rd Edition, O'Reilly Media, 2022. ISBN: 9781098110208
- AI Engineering, Chip Huyen, O'Reilly Media, 2025. ISBN: 9781098166304
- Building LLM Powered Applications, Valentino Zocca & Moez Ali, Packt Publishing, 2024. ISBN: 9781836200079.

Recommended Online Resources

- MLflow Documentation
- Docker Documentation
- Kubernetes Documentation
- FastAPI Documentation
- Hugging Face Documentationtools and Linux security environment.

K.R. MANGALAM UNIVERSITY

Value Added Course-I

Program Name	MCA AI ML	Semester	III
Course Name:	**Course Code**	**L-T-P**	**Credits**
VAC-I	VAC	2-0-0	2
Type of Course:	Value Added Course-I		

Standard Operating Procedure (SOP)

1. Introduction

In alignment with the vision and recommendations of the National Education Policy (NEP) 2020 and the guidelines of the University Grants Commission (UGC), the School of Engineering & Technology (SOET) offers Value Added Courses (VAC) as an integral component of holistic and multidisciplinary education. These courses are designed to complement the core academic curriculum by enabling students to acquire additional knowledge, practical skills, ethical values and interdisciplinary competencies essential for personal growth, professional development and lifelong learning.

VAC courses are intended to broaden students' perspectives beyond their primary discipline and help them develop competencies relevant to emerging technologies, societal needs, innovation, entrepreneurship, communication, sustainability, digital literacy, employability and responsible citizenship.

2. Objectives of VAC Courses

The VAC framework aims to:

• Promote multidisciplinary and holistic learning as envisioned in NEP 2020

• Enhance students' employability, practical exposure and professional competencies

• Develop critical thinking, creativity, innovation and problem-solving abilities

• Encourage experiential, skill-based and application-oriented learning

• Provide exposure to emerging technologies, industry trends and societal challenges

• Strengthen ethical values, environmental consciousness and social responsibility

• Support self-learning, lifelong learning and professional growth

3. Nature of VAC Courses

• VAC courses shall be interdisciplinary and skill-oriented in nature

• The courses may focus on:

- emerging technologies,
- innovation,
- entrepreneurship,
- digital skills,
- sustainability,
- communication,
- design thinking,
- ethics,
- problem solving,
- financial literacy,
- professional skills and related areas

• The courses shall emphasize practical implementation, experiential learning and real-world applications rather than purely theoretical delivery

• VAC courses may be offered in:

- offline,
- online,
- blended or
- bootcamp-based modes

4. Course Structure & Delivery

• VAC courses shall generally be short-duration and flexible learning courses

• The delivery model may include:

- workshops,
- practical sessions,
- mini projects,
- case studies,
- simulations,
- industry interactions,
- self-learning activities,
- hackathons,
- assignments and

- collaborative learning exercises

• Industry experts, practitioners, entrepreneurs and domain specialists may be invited for course delivery and mentoring support

• The courses shall encourage active participation, creativity and peer learning

5. Learning Outcomes

VAC courses are expected to help students:

• Acquire additional practical and interdisciplinary competencies

• Develop industry-relevant and future-ready skills

• Improve communication, teamwork and professional confidence

• Apply knowledge in solving real-world and societal problems

• Strengthen innovation, entrepreneurial thinking and digital readiness

• Enhance adaptability and lifelong learning capabilities

6. Assessment & Evaluation

• The assessment methodology shall focus primarily on continuous and competency-based evaluation

• Evaluation components may include:

- assignments,
- practical activities,
- projects,
- quizzes,
- presentations,
- participation,

- case studies,
- demonstrations and
- reflective learning reports

• The evaluation process shall emphasize practical understanding, implementation skills and active engagement rather than rote learning

7. Self-Learning & Skill Development

• VAC courses may include self-learning components through:

- MOOCs,
- certification programs,
- online learning platforms,
- industry learning pathways and
- guided independent learning activities

• Students may be encouraged to complete industry-recognized certifications and skill-development programs aligned with the course objectives

8. Industry & Experiential Learning Integration

• VAC courses may incorporate:

- industry projects,
- field activities,
- innovation challenges,

- hackathons,
- product-building exercises,
- startup-oriented learning and
- practical implementation tasks

• The courses shall promote experiential and application-oriented learning environments aligned with modern industry practices and emerging technology ecosystems

9. Quality Assurance & Continuous Improvement

• VAC courses shall be periodically reviewed and updated based on:

- industry trends,
- technological advancements,
- learner feedback,
- societal requirements and
- employability needs

• The School may seek feedback from:

- students,
- faculty,
- industry experts and
- stakeholders for continuous enhancement of course quality and relevance

10. Alignment with NEP 2020

The VAC framework supports the core principles of NEP 2020 by promoting:

• Holistic and multidisciplinary education

• Skill development and employability enhancement

• Flexibility and learner-centric education

• Experiential and competency-based learning

• Critical thinking and problem-solving abilities

• Digital learning and self-learning culture

• Innovation, entrepreneurship and lifelong learning

• Value-based and socially responsible education

11. Expected Outcomes

Through the VAC ecosystem, students are expected to become:

• Industry-ready and skill-oriented professionals

• Creative thinkers and problem solvers

• Socially responsible and ethically aware individuals

• Self-driven lifelong learners

• Innovators capable of adapting to emerging technologies and evolving global challenges

PROFESSIONAL PROBLEM SOLVING & CAREER READINESS-I

Program Name	MCA AI ML	Semester	III
Course Name:	**Course Code**	**L-T-P**	**Credits**
Professional Problem Solving & Career Readiness-I	CSE013	2-0-4	4
Nature of Course:	Practical & Placement-Oriented Skill Development Course		
Evaluation Type	Continuous Internal Evaluation		
Mode	Classroom + Coding Lab + Mock Assessment + Mentorship		

Course Perspective

The course "Professional Problem Solving & Career Readiness – I" is designed to prepare students for modern technical recruitment processes through rigorous practice in problem-solving, object-oriented programming, core computer science fundamentals, coding interviews and professional communication.

The course follows a highly practical and industry-focused approach where students continuously solve coding problems, work on mini-projects, revise interview-oriented technical concepts and participate in mock technical and HR interviews. The course integrates aptitude building, debugging skills, DSA-based problem solving, resume preparation and placement readiness activities into a unified professional development framework.

The course emphasizes:

- Industry-oriented coding and debugging practice
- Interview-focused revision of OOPs, OS and DBMS

- DSA-based problem-solving patterns
- Technical communication and mock interview preparation
- Continuous assessment and performance monitoring
- Placement-oriented skill enhancement

The course is designed to improve employability readiness and technical confidence for campus placements, internships and professional software engineering roles.

Course Objectives

The course aims to:

- Develop professional problem-solving and coding skills required for technical recruitment processes.
- Strengthen students' understanding of core computer science concepts from placement and interview perspectives.
- Enhance debugging, logical reasoning and algorithmic thinking abilities through practical coding exercises.
- Build confidence in technical communication, resume presentation and interview performance.
- Prepare students for industry-level coding assessments and placement workflows.

Course Outcomes (Cos)

CO	Course Outcome	Bloom's Level
CO1	Apply object-oriented programming concepts for software problem solving.	L3 – Apply
CO2	Solve coding and DSA-based interview problems.	L3 – Apply
CO3	Analyze OS and DBMS concepts for technical interviews.	L4 – Analyze
CO4	Demonstrate professional readiness in coding and interview environments.	L3 – Apply

CO–PO Articulation Matrix

COs \ POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	2	2	1	3	1	1	1	1	1	2
CO2	3	3	3	2	3	1	1	1	1	1	3
CO3	3	3	2	2	3	1	1	1	1	1	2
CO4	2	2	2	1	2	1	2	3	3	2	3

Course Structure & Execution Model

Component	Focus Area	Hours	Marks
Module 1	OOPs & Software Development Fundamentals	10 Hours	30
Module 2	Coding & Problem Solving	20 Hours	50
Module 3	Operating Systems & DBMS for Interviews	20 Hours	50
Module 4	Mock Interviews & Career Readiness	10 Hours	40
Module 5	Final Placement Readiness Assessment	—	30
Total		60 Hours	200

Course Outline:

Unit Number: 1	Title: Object-Oriented Programming & Software Development Fundamentals	No. of hours: 10

Content:

- OOP Paradigm & Object-Oriented Design
- Classes, Objects & Constructors
- Encapsulation & Data Hiding
- Inheritance & Polymorphism

- Constructor Chaining & Method Overriding
- Composition vs Aggregation
- Static Members & Memory Management
- Console-Based Java Mini Projects
- Code Reusability & Modular Programming

Mini Projects

Project 1: Employee Payroll Management System

Develop a console-based payroll management system using Java OOP concepts including inheritance, abstraction, polymorphism and encapsulation.

Project 2: Car Rental Management System

Develop a console-based rental management system for managing cars, customers and rental operations using collection-based data handling and OOP principles.

Unit Number: 2	**Title: Coding, DSA & Problem-Solving Patterns**	**No. of hours: 20**

Content:

- Mathematical & Number-Based Problems
- Recursion & Pattern-Based Problems
- Array Traversal & Prefix Sum Problems
- String Manipulation & Hashing
- Searching & Sorting Techniques

- Sliding Window & Two Pointer Techniques
- Stack & Queue Problem Solving
- Binary Search on Answer
- BFS & DFS Basics
- Backtracking & Subset Problems
- Debugging & Optimization Techniques

Coding Platforms

- HackerRank
- LeetCode
- CodeChef
- Internal Coding LMS

Practical Focus

- Daily coding practice
- Timed coding assessments
- Debugging exercises
- Interview-style coding rounds

Unit Number: 3	**Title: Technical Fundamentals for Interviews (OS & DBMS)**	**No. of hours: 20**
Content:		

Operating Systems

- Process & Thread Management
- CPU Scheduling Algorithms
- Process Synchronization
- Deadlocks & Banker's Algorithm
- Memory Management & Paging
- Virtual Memory & Page Replacement
- File Systems & Disk Scheduling
- I/O Systems & Interrupts

Database Management Systems

- ER Model & Relational Model
- SQL Queries & Joins
- Aggregate Functions & Subqueries
- Functional Dependencies & Normalization
- Transactions & ACID Properties
- Concurrency Control & Locks
- Recovery Techniques

Practical Focus

- SQL query writing practice
- OS interview question discussions
- Scenario-based technical quizzes
- Interview-oriented MCQs and debugging exercises

Unit Number: 4	Title: Career Readiness & Mock Interview Preparation	No. of hours: 10

Content:

- Resume Building & LinkedIn Optimization
- Technical Mock Interviews
- HR & Behavioral Interview Preparation
- Self-Introduction & Professional Communication
- Group Discussion & Extempore Practice
- Email Writing & Corporate Etiquette
- Personality Grooming & Professional Conduct
- Placement Assessment Preparation

Practical Activities

- Technical Interview Simulations
- HR Mock Interviews
- Resume Evaluation Sessions

- Communication Practice Sessions

Industry Assessment Simulations

Lab Activities & Practical Assignments

Lab No.	Experiment / Activity	Tools / Platforms Used	CO Mapping
1	OOP Mini Project Development	Java, IntelliJ IDEA / Eclipse	CO1
2	Coding Problem Solving Practice	HackerRank, LeetCode	CO2
3	DSA Pattern-Based Coding	Java/Python/C++	CO2
4	OS & DBMS Technical Quiz Practice	Google Forms, LMS	CO3
5	SQL Query & Normalization Exercises	MySQL / PostgreSQL	CO3
6	Resume & LinkedIn Preparation Workshop	Canva, LinkedIn	CO4
7	Mock Technical Interview	Interview Panels	CO4
8	HR & Communication Assessment	GD / Mock HR	CO4

Assessment & Evaluation Scheme

Component	Weightage
Coding Assessments	50 Marks

Technical Assessments (OS/DBMS/OOPs)	50 Marks
Mock Interviews & Communication	40 Marks
Mini Projects & Lab Activities	30 Marks
Final Placement Readiness Assessment	30 Marks
Total	200 Marks

Evaluation Guidelines

- Continuous internal evaluation shall be followed.
- Students must actively participate in coding labs and mock interviews.
- Coding assessments shall be conducted on coding platforms or LMS systems.
- Mock interviews shall be rubric-based and evaluated by faculty/trainers.
- Minimum 60% overall marks are required for successful completion of the course.
- Performance dashboards may be maintained for tracking coding progress, interview readiness and technical assessments.

Recommended Tools & Platforms

- HackerRank
- LeetCode
- CodeChef
- IntelliJ IDEA / Eclipse

- MySQL / PostgreSQL
- Google Forms
- Canva
- LinkedIn
- Internal LMS

Suggested Text Books

1. Cracking the Coding Interview Gayle Laakmann McDowell, **6th Edition**, CareerCup, **2015**.
2. Head First Object-Oriented Analysis and Design O'Reilly Media, **2006**.
3. Operating System Concepts **10th Edition**, Wiley, **2018**.
4. Database System Concepts **7th Edition**, McGraw-Hill, **2019**.

Recommended Online Resources

- HackerRank — https://www.hackerrank.com/
- LeetCode — https://leetcode.com/
- CodeChef — https://www.codechef.com/
- GeeksforGeeks Interview Preparation — https://www.geeksforgeeks.org/interview-preparation-for-software-developer/
- PrepInsta Placement Preparation — https://prepinsta.com/
- IndiaBix Aptitude Preparation — https://www.indiabix.com/

SUMMER INTERNSHIP

Program Name	MCA AI ML	Semester	III
Course Name:	**Course Code**	**L-T-P**	**Credits**
Summer Internship	SIAML051	0-0-4	2
Type of Course:	INT		
Pre-requisite(s): Problem Solving with C++/Python, Data Structures basics			

Course Description

The Summer Internship course is designed to provide students with practical exposure to real-world professional environments by integrating academic learning with industry-oriented experiences. Conducted during the summer break, the internship enables students to apply theoretical concepts, technical knowledge, and problem-solving skills in practical situations across industry, research, entrepreneurial, or organizational settings. Carrying a weightage of 2 academic credits, the course serves as a mandatory component of the curriculum and focuses on enhancing professional competence, workplace ethics, communication abilities, and industry readiness.

The course emphasizes experiential learning through active participation in technical, managerial, research, or innovation-based tasks under professional supervision. Students gain exposure to real-world practices, organizational workflows, and collaborative work environments, helping them develop self-learning abilities, critical thinking, adaptability, and professional responsibility. The internship duration of 6–8 weeks allows students to strengthen the connection between academic knowledge and practical applications while preparing them for future career opportunities and higher professional engagement.

The Course Outcomes (COs).

On completion of the course the participants will learn to:

CO	Course Outcome	Bloom's Level
CO1	Apply academic knowledge and technical skills to perform assigned tasks in a professional work environment.	L3 – Apply
CO2	Demonstrate professional ethics, communication skills, and teamwork during internship activities.	L3 – Apply
CO3	Analyze real-world industry practices, organizational workflows, and problem-solving approaches observed during the internship.	L4 – Analyze
CO4	Prepare and present a comprehensive internship report reflecting learning experiences, outcomes, and professional growth.	L4 – Analyze

CO–PO Articulation Matrix

COs \ POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	2	2	1	2	1	2	2	2	2	3
CO2	1	2	1	1	1	2	3	3	3	2	2
CO3	2	3	2	3	2	2	2	2	2	2	3
CO4	1	2	2	2	1	1	2	2	3	3	3

CO–PSO Articulation Matrix

COs \ PSOs	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	2	3	1	1	1
CO2	1	2	2	1	1
CO3	2	2	3	2	1
CO4	1	2	2	3	2

Objectives

The Summer Internship aims to:

- Provide students with hands-on industry exposure and practical learning experiences.
- Bridge the gap between academic concepts and real-world applications.
- Enhance technical competency, communication skills, teamwork, and professional ethics.
- Develop analytical thinking, adaptability, and workplace readiness.
- Familiarize students with organizational workflows, project execution, and modern tools/technologies.
- Encourage experiential learning through research, development, innovation, and problem-solving activities.
- Strengthen documentation, reporting, and presentation skills through structured evaluation processes.

Internship Scope and Categories

Students may opt for any one of the following approved internship categories:

External (Outside KRMU) Tracks

Track 1: Industrial Internship

Track 2: Government / Research Institution

Track 3: Start-ups

Internal (Inside KRMU) Tracks

Track 4: Bootcamp

Track 5: University Specific Projects

Track 6: Ideas 4.0

Track 7: Global Certifications

Track 1: Industrial Internship

- Internship in reputed companies, IT firms, software industries, consulting organizations, or technical enterprises.
- Students will work on real-world industry problems, software development, testing, analytics, cloud computing, cybersecurity, AI/ML, or other domain-specific tasks.
- Mandatory submission of internship completion certificate and mentor verification.

Track 2: Government / Research Institution Internship

- Internship in government departments, public sector organizations, research labs, or academic/research institutions.

- Students may contribute to research activities, data analysis, technical documentation, simulations, or development projects.
- Exposure to research methodologies and institutional workflows is encouraged.

Track 3: Start-Up Internship

- Internship in start-ups or entrepreneurial organizations focusing on innovation and product development.
- Students may participate in prototype development, product design, business analytics, marketing technology, or software implementation.
- Encourages innovation, adaptability, and entrepreneurial thinking.

Track 4: Bootcamp (Internal – KRMU)

- Intensive skill-based training programs organized by the university.
- Focus on modern technologies such as Data Science, AI/ML, Web Development, Cloud Computing, DevOps, Cybersecurity, IoT, etc.
- Includes hands-on assignments, assessments, and mini projects.

Track 5: University Specific Projects

- Students work on projects addressing university-specific requirements or operational enhancements.
- Encourages interdisciplinary collaboration and solution development for institutional improvement.
- Projects may include automation systems, dashboards, portals, analytics platforms, or smart campus solutions.

Track 6. Ideas 4.0

- Innovation and idea development program focused on problem-solving and entrepreneurial creativity.
- Students develop proof-of-concept models, prototypes, or innovative technical solutions.
- Encourages design thinking, creativity, and innovation-driven learning.

Track 7: Global Certifications

- Students may complete internationally recognized certifications aligned with their specialization.
- Certifications may include cloud platforms, programming technologies, AI/ML tools, cybersecurity, data analytics, or software development frameworks.
- Practical implementation and assessment-based learning are mandatory.

Technology Exposure and Professional Tools

Students are encouraged to gain exposure to:

- Industry-standard software tools and development environments.
- Version control systems such as Git/GitHub.
- Cloud platforms such as AWS, Azure, or GCP.
- Data analytics, AI/ML frameworks, or software engineering tools.
- Collaboration and project management platforms.
- Documentation, reporting, and presentation tools.

Use of emerging technologies and industry practices is strongly encouraged to improve professional readiness and technical competency.

Internship Duration and Timeline

Week	Activity/Stage
Week 0	Internship Registration on Projexa and Mentor Allocation
Week 1	Joining Organization / Orientation / Goal Setting
Week 2–3	Task Understanding and Initial Learning
Week 4–5	Practical Work / Technical Assignments / Development Activities
Week 6	Mid Internship Review and Mentor Interaction
Week 7	Advanced Task Execution / Documentation
Week 8	Final Submission, Report Preparation, Presentation & Viva

Internship Guidelines

Internship Approval

- Students must obtain prior approval from the department before joining any internship organization/program.
- Internship domain must align with the student's specialization or technical learning objectives.

Attendance Requirement

- Minimum attendance and active participation throughout the internship duration are mandatory.

- Attendance records must be certified by the organization/mentor.

Mentor Allocation

- Each student will be assigned a faculty mentor for monitoring and evaluation.
- External internships will additionally require an industry mentor/supervisor.

Progress Monitoring

- Students must participate in regular mentor interactions through Projexa.
- Faculty mentors will monitor progress, documentation, and learning outcomes.

Professional Conduct

Students are expected to:

- Follow workplace ethics and discipline.
- Maintain confidentiality of organizational data.
- Demonstrate professionalism, punctuality, and responsibility.
- Respect organizational policies and reporting structures.

Roles and Responsibilities of Faculty Mentors

Regular Interactions

- Conduct regular mentor interactions through Projexa in online/offline mode.
- Monitor student progress and provide academic guidance.
- A minimum of four (4) mentor interactions is mandatory.

Progress Evaluation

- Review assigned tasks, internship activities, and technical learning outcomes.
- Maintain interaction records and evaluation comments on Projexa.

Report Verification

- Verify internship report authenticity and learning outcomes.
- Ensure proper documentation and completion of all required submissions.

Presentation Evaluation

- Assess the student's presentation, technical understanding, communication skills, and professional growth.

Evaluation Criteria: Each student must be evaluated on the following documents submitted after completing their internship/certification:

Deliverable	Description	Marks
Summer Internship Report File	A detailed Summer Internship report in the prescribed format including internship objectives, work completed, learning outcomes, and a signed one-page logbook with progress details endorsed by the industry mentor.	20
Attendance Record	Signed and stamped attendance record certified by the industry mentor/organization.	10
Hard Bound Completion Certificate	Submission of hard-bound copy of the internship completion certificate issued by the organization/industry.	10

Mentor Interaction through Projexa	Participation in four scheduled mentor interaction meetings conducted through Projexa for progress monitoring and evaluation.	40
Viva / Final Presentation	Final presentation and viva demonstrating internship work, technical learning, outcomes, and professional experience gained during the internship.	20

Rubrics:

Assessment Component	Excellent (9–10)	Good (7–8)	Satisfactory (5–6)	Needs Improvement (0–4)
Summer Internship Report File (20 Marks)	Report is highly organized, comprehensive, professionally formatted, and includes detailed learning outcomes, activities, and signed logbook with complete progress details.	Report is well-structured with relevant content and proper documentation; minor details may be missing.	Report covers basic internship details but lacks depth, clarity, or proper formatting.	Report is incomplete, poorly organized, or missing significant details/logbook.

Attendance Record (10 Marks)	Attendance is fully signed and stamped with complete authenticity and timely submission.	Attendance record is complete with minor formatting/documentation issues.	Attendance record is partially complete or submitted late.	Attendance proof is incomplete, unclear, or not authenticated.
Hard Bound Completion Certificate (10 Marks)	Properly submitted hard-bound certificate with all required details and professional presentation.	Certificate submitted with minor issues in formatting or presentation.	Certificate submitted but lacks proper binding or some required details.	Certificate not submitted or significantly incomplete.
Mentor Interaction through Projexa (40 Marks)	Attended all mentor meetings actively with excellent progress updates, responsiveness, and task completion.	Participated regularly with good interaction and satisfactory progress updates.	Limited participation or inconsistent progress reporting during meetings.	Poor participation or absence in mentor interactions.
Viva / Final Presentation (20 Marks)	Presentation is clear, confident, technically strong, and demonstrates	Presentation is well-delivered with good understanding and communication skills.	Presentation communicates basic work done but lacks	Presentation is unclear, incomplete, or demonstrates

	excellent understanding of internship work and outcomes.		clarity or confidence.	insufficient understanding.

Internship Documentation Requirements

To be Submitted After Completion of Internship

Mandatory Documents

- Summer Internship Report File
- Signed One-Page Logbook
- Attendance Record
- Internship Completion Certificate
- Mentor Evaluation/Verification (if applicable)
- Presentation PPT
- Any supporting documents/certifications relevant to the internship

Plagiarism & Ethical Guidelines

- Internship reports must be original and based on actual work carried out during the internship.
- Proper acknowledgment and citations must be included wherever required.
- Submission of false or fabricated internship records/documents will lead to disciplinary action.

- Maximum allowable plagiarism should not exceed institutional norms (≤15%).
- Professional ethics, confidentiality, and organizational policies must be strictly followed.

Expected Learning Outcomes

By the end of the Summer Internship, students should be able to:

- Demonstrate practical understanding of industry practices and professional workflows.
- Apply technical and analytical skills in real-world scenarios.
- Communicate effectively in professional environments.
- Develop teamwork, adaptability, and problem-solving abilities.
- Present structured documentation of internship experiences and learning outcomes.
- Enhance employability, professional confidence, and industry readiness.

CAPSTONE PROJECT-I

Program Name	MCA AI ML	Semester	III
Course Name:	**Course Code**	**L-T-P**	**Credits**
Capstone Project-I	PRAML052	0-0-8	4
Type of Course:	Proj		
Pre-requisite(s): Problem Solving with C++/Python, Data Structures basics			

Course Description

The course is designed to provide students with an opportunity to apply theoretical knowledge, technical competencies, and problem-solving abilities to the development of a comprehensive solution for a real-world problem. The course emphasizes innovation, research orientation, design thinking, and interdisciplinary learning through the execution of individual or team-based projects under faculty mentorship. Carrying academic credit as prescribed in the curriculum, the course serves as an essential component for enhancing practical knowledge, technical proficiency, teamwork, and professional preparedness.

The course focuses on experiential and outcome-based learning by engaging students in activities such as problem identification, requirement analysis, literature review, system design, implementation, testing, documentation, and presentation. Students are encouraged to explore emerging technologies, industry-oriented challenges, research problems, entrepreneurial ideas, or socially relevant applications while developing their projects.

Through the project work, students gain hands-on exposure to project planning, technical documentation, collaborative development, research methodologies, and professional ethics. The course also strengthens communication skills, analytical thinking, creativity, and decision-making abilities while fostering independent learning and innovation. By the end of the project, students are expected to demonstrate the successful integration of domain knowledge and technical skills through the development and presentation of a functional prototype,

software solution, research outcome, or engineering model aligned with real-world applications and professional standards.

The Course Outcomes (COs).

On completion of the course the participants will learn to:

CO	Course Outcome	Bloom's Level
CO1	Analyze and define a real-world problem statement by applying domain knowledge, research understanding, and requirement analysis techniques.	L4 – Analyze
CO2	Design and develop an appropriate project solution using suitable tools, technologies, methodologies, and engineering practices.	L3 – Apply
CO3	Evaluate the performance, functionality, and effectiveness of the developed project through testing, validation, and result analysis.	L5 – Evaluate
CO4	Prepare technical documentation and present the project outcomes effectively while demonstrating teamwork, communication skills, and professional ethics.	L3 – Apply

CO–PO Articulation Matrix

COs \ POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	3	2	3	2	1	1	1	2	1	3
CO2	3	3	3	2	3	1	1	2	2	2	3
CO3	3	3	3	3	3	2	1	2	2	2	3
CO4	1	2	2	1	2	2	3	3	3	3	2

CO–PSO Articulation Matrix

COs \ PSOs	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	2	3	2	1
CO2	3	3	2	2	3
CO3	2	2	3	3	2
CO4	2	2	1	2	2

1. Objectives

All student projects must:

- Employ industry-relevant technologies and tools.
- Reflect a well-defined development or research process from ideation to final deployment or publication.

- Promote innovation and real-world problem-solving.
- Deliver meaningful outcomes such as a functional application, published research, or validated prototype.

2. Project Scope and Categories

Projects must fall under one of the following approved categories:

A. Research-Oriented Projects

- Must address futuristic domains like Generative AI (e.g., GANs, Transformers, LLMs), cybersecurity, quantum computing, etc.
- Multidisciplinary Research Integration: Projects are encouraged to bridge multiple disciplines by combining core technology domains (e.g., AI, cybersecurity, quantum computing) with fields such as Healthcare, Finance, Law & Ethics, Education, Social Sciences, or Environmental Studies.
- Require mandatory publication of a research paper in a reputed journal or international conference before final submission.
- Supervised by internal faculty mentor.

B. Industry-Based Projects

- Should be based on real-world industry challenges with practical applications.
- Must demonstrate full implementation, deployment, and usability.
- Students will work under external mentorship from an industry along with a faculty mentor from SOET.

- External Industry Mentorship Confirmation certificate confirms that industry expert/mentor will be guiding the student(s) on their project work. The certificate must be issued and signed by the external mentor and printed on the official letterhead of the company/organization (if available).
- A certificate of completion from the associated organization is mandatory.

C. University-Focused Interdisciplinary Projects / Community Solution Based Projects

- Projects under this category should aim to identify and address real-world challenges, enhancement opportunities, or operational gaps within K.R. Mangalam University.
- Students are encouraged to collaborate across disciplines, integrating knowledge from diverse domains to develop innovative, practical, and sustainable solutions for campus improvement.
- Projects must demonstrate a clear understanding of institutional needs and propose scalable or implementable models that can be piloted or adopted by university stakeholders.

Real-World Problem Orientation (for Community Solution Based Projects):

- Projects should address practical issues such as health, sanitation, education, environmental sustainability, digital literacy, social inclusion, transportation, waste management, or livelihood enhancement.

D. Start-Up Projects

- Projects under this category should focus on identifying real-world problems and developing innovative, technology-driven solutions with the potential for commercialization.
- Students are encouraged to think entrepreneurially, transforming their ideas into viable products or services by building functional prototypes or Minimum Viable Products (MVPs).

- Projects must include key elements of a start-up framework such as market research, user validation, business model design, and go-to-market strategy.
- The outcome should demonstrate not only technical feasibility but also market relevance, with the potential to seek incubation or funding support from the university's innovation/start-up ecosystem.

E. Deep-Tech & Systems (Learning-by-Replication)

- This category emphasizes deep technical mastery by replicating foundational technologies that power modern computing. Students build simplified versions of complex systems to understand how they truly work "under the hood," going beyond API usage or high-level application development.
- Define a Clear Scope: The system replica should have a minimum viable core (e.g., for a compiler: lexer + parser + code-gen) and extensions or optimizations (indexes, caching, scheduling algorithms).

Examples of "Build Your Own X":

- Build your own database engine (query parser, storage engine, index manager)
- Build your own compiler (lexer, parser, AST, code generator)
- Build your own search engine (crawler, indexer, ranking algorithms)
- Build your own machine learning framework (autograd engine, tensor ops)
- Build your own virtual machine, interpreter, or runtime environment

F. Hardware & Software Integration Using IoT / Embedded Systems

- This category focuses on building solutions that combine physical hardware (sensors, actuators, microcontrollers) with software systems (apps, cloud platforms, analytics engines). Students must design end-to-end solutions that sense, process, communicate, and act on real-world data.
- Projects without actual hardware implementation will not be accepted. A fully functional hardware prototype is mandatory, including working sensors/actuators, embedded programming, connectivity, and a software interface.

3. Technology Stack and Industry Tools

- Students must select a technology stack aligned with current industry standards, such as MERN, MEAN, Django, Flutter, or cloud-native architectures (AWS, Azure, GCP).
- Projects must incorporate industry-relevant tools and frameworks for version control (Git/GitHub), containerization (Docker), CI/CD (GitHub Actions, Jenkins), and agile development practices.
- Use of modern development environments, APIs, and deployment platforms is strongly encouraged to ensure scalability, maintainability, and real-world applicability of the solution.

4. Formation of Groups

4.1 Group Size & Composition

- **Optimal Group Size:** Groups will consist of 2 students to allow effective collaboration while maintaining manageable group dynamics.
- **Skill Diversity:** Form groups with a mix of skill sets (e.g., coding, design, research, communication, etc.) to enhance collaboration. Students should be encouraged to select group members based on complementary skills.

4.2 Group Formation Process

- **Self-selection:** Allow students to form their own groups, encouraging them to choose teammates based on project interests.
- **Pre-formed Groups:** Alternatively, groups can be assigned by the coordinator to ensure diversity of skills and ideas across all teams.

4.3 Team Finalization Rule

Once teams are formed and registered, no changes in team composition will be allowed under any circumstances.

4.4 Timeline

Week	**Activity/Stage**
Week 0	• Onboarding on Projexa (Project Management Tool) • Team formation, • Faculty Mentor allotment
Week 1–2	• Problem Statement Finalization • Initial Mentor Interactions and discussion
Week 3–4	• Synopsis Submission • Project Proposal Evaluation and Approval
Week 5, 6 and 7	Implementation Sprint 1: Core Modules, MVP Build, Mentor Interaction
Week 8	Development Phase – Part 1: • Module-wise implementation begins
Week 9–10	Development Phase – Part 2:

	• Module-wise implementation begins
Week 11–12	Mid Term Evaluation
Week 13–14	• Testing and Debugging • Performance evaluation
Week 15–16	• Final Report Submission, • Project Video (1-2 mins) demonstrating working project • Presentation & Viva Preparation

5. Guidelines for Project Selection

For any project category (Research-Based, Development-Based, Start-Up, or University-Focused Interdisciplinary), students may select their project through either of the following approaches:

- **Faculty-Suggested Problem Statements:**
    - Faculty members will provide curated problem statements based on their domain expertise, current research trends, or institutional needs.
    - Students can choose from these problems and discuss with the faculty mentor before finalizing.
- **Student-Proposed Problem Statements**:

Student teams are encouraged to identify real-world problems based on personal interests, industry trends, or societal challenges.

The proposed problem must be original, relevant, and feasible within the given timeline and resources.

Final approval is subject to evaluation by the project review committee.

6. Faculty Allocation Process

- Faculty-Suggested Problem Statement: If a team selects a problem provided by a faculty member, the team will be allocated to that faculty after peer review and approval by the Project Coordinator.
- Student-Proposed Problem Statement: If the team proposes its own problem statement, then after peer review and validation by the Project Coordinator, an appropriate faculty mentor will be assigned based on domain expertise.
- Inter School Problem Statement: If a team chooses a problem statement proposed by a faculty member from another school/department, the team will be assigned to that faculty member as their project mentor only after a peer review and approval by the Project Coordinator.

7. Roles and Responsibilities of Faculty Mentors

- Regular Interactions: Conduct regular meetings with the assigned team(s) through Projexa in online/offline mode. All meeting records, including agenda, minutes, and attendance, must be documented and uploaded on Projexa. A minimum of five (5) interactions is mandatory, with at least two (2) interactions per month. If the required interactions are not completed within the specified timeline, the pending interaction slots will be automatically closed and marked as incomplete.
- Task Monitoring: Assign tasks with clear deadlines and track the progress and completion status for each team through the project cycle.
- Evaluation After Interaction: After every interaction, the faculty mentor must evaluate the team's progress and update the evaluation/comments directly on Projexa as part of the meeting record.

- Meeting Confirmation & Rescheduling: Faculty must respond to meeting requests initiated by student teams. If unable to conduct a scheduled meeting, it should be rescheduled within one week.
- Justification for Meeting Cancellation: Any cancellation of scheduled team meetings by the mentor must be supported with a valid and documented reason.
- GitHub Activity Monitoring: Regularly monitor each team's GitHub repository to verify individual contributions and maintain performance records for all team members.

8. Evaluation Metrics

Projects will be evaluated in 4 phases (Total 100 Marks):

- Synopsis Presentation (20 Marks)
- Mid Term Presentation (30 Marks)
- Final Term Presentation (40 Marks)
- Project Mentor Interaction (25 Marks)

For each evaluation, marks will be cumulated from the following components:

- Project Mentor Evaluation
- Project Evaluation Committee

Evaluation Stage	**Project Mentor (Interaction)**	**Project Evaluation Committee**
Synopsis Presentation (20)	5	15
Mid Term Presentation (30)	10	20

Final Term Presentation (40)	10	40
Total	**25**	**75**
Total Marks	**(25+75) = 100**	

Project Evaluation Committee Marking Scheme

Evaluation Stage	**Criteria**	**Marks**
Synopsis Presentation	Creative/Naive/Real-world Problem	5
	Clarity & Feasibility of Project Objectives	5
	Preparation & Presentation of PPT	5
	Total	**15**
Mid-Term Presentation	Project Efforts – UI/Implementation	10
	Effective Use of Modern Technology Stack	5
	Effective Coding Practices / Use of Git	5
	Total	**20**
Final-Term Presentation	Final Production Demonstration & Completion w.r.t Objectives	20
	Impact & Use Cases of the Project	10
	Deployment	10
	Total	**40**

9. Policy on Missed Presentation Schedules

Students are strictly required to adhere to the assigned schedules for all project presentations, including the

Synopsis, Mid-Term, and Final-Term evaluations.

- Only one rescheduling opportunity will be granted in case a team or individual misses their assigned presentation slot. This rescheduling must be approved in advance (where possible) or justified immediately after the missed session with valid reasons.
- A penalty of 5 marks will be deducted for each missed presentation schedule, irrespective of the evaluation stage.
- Failure to appear even after the rescheduled opportunity will result in zero marks being awarded for that evaluation component.

This policy ensures fairness, accountability, and professionalism in the evaluation process.

10. Project Documentation Requirements – Stage-Wise

1. Synopsis Stage

To be submitted before the Synopsis Presentation:

- Synopsis Report
- Synopsis PPT
- Mentorship Confirmation Certificate (must be on industry letterhead, duly signed by the External Mentor)

2. Mid-Term Evaluation Stage

To be submitted during the Mid-Term Presentation:

- Mid Evaluation PPT
- Project Use Case & Deployment Confirmation Certificate (issued by the Industry Mentor, confirming project relevance and real-world deployment potential)

3. Final-Term Evaluation Stage

To be submitted during or before Final Presentation:

- Final Project Report
- PPT for Final Presentation
- Video Demonstration (1–2 minutes) – should clearly document key features, execution workflow or demonstration of functionality, and real-world impact or use-case. To be submitted in MP4 format or via a shared video link.
- Plagiarism Certificate – must be compulsorily appended to the Final Project Report, obtained using institutional or approved plagiarism checking tools.

11. Plagiarism & Ethical Guidelines

- Maximum allowable plagiarism: ≤ 15%
- Check plagiarism using open-source tools and proper citations
- No reuse of previous projects
- Intellectual property rights (if applicable)
- Compulsory Plagiarism Certificate to be appended in the report file

12. Project Outcome Requirements by each Project Category

Project Category	**Desired Outcome**
Industry-Based Project	Project Use Case & Deployment Confirmation Certificate from Industry Mentor; Fully deployed solution (mandatory for final evaluation)

Research-Based Project	Proof of publication in a reputed conference or journal (UGC CARE / Scopus / IEEE / Springer, etc.)
University-Focused Interdisciplinary Projects / Community Solution Based Projects	Project Use Case & Deployment Confirmation Certificate from Faculty Mentor; Fully deployed solution (mandatory for final evaluation)
Start-Up Project	Incubation Letter from recognized incubator/mentor/startup cell; Product delivery and deployment with functional MVP; Patent Grant & Publish: The project must result in a patent application, and obtaining a patent grant is compulsory for the project to qualify under this category.
Deep-Tech & Systems (Learning-by-Replication)	Working System Prototype: A functional, simplified replica of a core deep-tech system (e.g., compiler, database, OS module) demonstrating internal mechanisms and correctness.
Hardware & Software Integration Using IoT / Embedded Systems	A fully functional hardware prototype is mandatory, including working sensors/actuators, embedded programming, connectivity, and a software interface.

Semester: IV

CAPSTONE PROJECT-II

Program Name	MCA AI ML	Semester	IV
Course Name:	Course Code	L-T-P	Credits
Capstone Project-II	INAML051	0-0-24	12
Type of Course:	Proj		

Course Description

Capstone Project – II is a major industry-integrated project course designed to provide students with intensive exposure to solving real-world industrial problems through end-to-end development, deployment and validation of technology solutions. The course emphasizes professional engineering practices, industry collaboration, scalable deployment, research orientation, innovation and production-ready implementation.

Unlike exploratory or academic mini-projects, this course strictly focuses on industry-specific problem statements sourced from companies, startups, industries, research labs, government organizations or professional mentors. Students are expected to work on practical business, engineering, automation, AI/ML, cybersecurity, cloud, data engineering, embedded systems or product-development challenges directly relevant to industrial environments.

The course follows a complete software/product engineering lifecycle including:

- problem identification,
- requirement analysis,
- architecture design,
- implementation,
- deployment,
- testing,

- validation,
- optimization,
- documentation and
- professional presentation.

Students will work under joint mentorship of:

- an **Industry Mentor** (mandatory), and
- a **Faculty Mentor**.

The project outcome must demonstrate:

- deployable implementation,
- measurable impact,
- production readiness,
- scalability and
- real-world usability.

The course also emphasizes:

- collaborative development,
- GitHub-based project management,
- agile workflows,
- DevOps practices,
- technical documentation,
- client communication and
- professional ethics.

By the end of the course, students must demonstrate a complete working solution aligned with real industrial use-cases and industry standards.

The Course Outcomes (COs).

On completion of the course the participants will learn to:

CO	Course Outcome	Bloom's Level
CO1	Analyze and define an industry-specific real-world problem.	L4 – Analyze
CO2	Design and implement a deployable industry-grade solution.	L3 – Apply
CO3	Evaluate scalability, performance and usability of the developed system.	L5 – Evaluate
CO4	Demonstrate professional project execution, documentation and deployment practices.	L3 – Apply

CO–PO Articulation Matrix

COs \ POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	3	2	3	2	1	1	1	2	1	3
CO2	3	3	3	2	3	1	1	2	2	2	3
CO3	3	3	3	3	3	2	1	2	2	2	3
CO4	1	2	2	1	2	2	3	3	3	3	2

CO–PSO Articulation Matrix

COs \ PSOs	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	2	3	2	1
CO2	3	3	2	2	3
CO3	2	2	3	3	2
CO4	2	2	1	2	2

1. **Objectives**

All projects under Capstone Project – II must:

- Address a real-world industry problem provided by a company, startup, research lab, government organization or industrial mentor
- Demonstrate complete implementation, deployment and validation of the proposed solution
- Use industry-standard tools, frameworks and engineering workflows
- Follow professional software/product development lifecycle practices
- Deliver measurable outcomes such as deployment, automation, analytics, optimization or operational improvements
- Include proper documentation, deployment workflows and scalability considerations
- Encourage innovation, research orientation and product thinking

2. **Mandatory Industry Integration Requirement**

• Every project must be strictly mapped to a real industry problem statement

• The problem statement must be approved by:

- Industry Mentor
- Faculty Mentor
- Project Review Committee

• Each team must have:

- one external industry mentor (mandatory)
- one faculty mentor

• The industry mentor must provide:

- mentorship confirmation certificate
- deployment/use-case validation certificate
- completion certificate at the end of the project

• Projects without valid industry mapping and industry mentorship will not be approved under this course

3. **Project Categories (Industry-Focused)**

Projects may belong to the following domains only if mapped to real industrial use-cases:

A. AI/ML & Data Science Solutions

Examples:

- Predictive analytics systems

- Recommendation engines
- AI chatbots
- Fraud detection systems
- Generative AI solutions

B. Cloud & DevOps Automation

Examples:

- CI/CD automation
- Cloud deployment systems
- Infrastructure monitoring platforms
- Kubernetes-based deployment systems

C. Cybersecurity Solutions

Examples:

- Threat detection systems
- Secure authentication systems
- Vulnerability analysis tools
- SIEM dashboards

D. Full Stack & Enterprise Applications

Examples:

- ERP modules

- CRM systems
- SaaS platforms
- Enterprise workflow automation

E. IoT & Embedded Systems

Examples:

- Smart monitoring systems
- Industrial automation solutions
- Sensor-based analytics systems
- Smart agriculture systems

F. Deep-Tech & Systems Engineering

Examples:

- Database engines
- Search engines
- Distributed systems
- Compiler/interpreter systems

G. Research & Innovation Projects

Examples:

- Research prototypes
- Experimental AI systems

- Emerging technology applications
- Interdisciplinary engineering systems

4. Technology Stack & Engineering Practices

Projects must incorporate modern engineering workflows including:

• Git & GitHub for version control and collaboration

• Cloud deployment platforms (AWS, Azure, GCP, Render, Vercel, etc.)

• Containerization using Docker

• CI/CD workflows using GitHub Actions or Jenkins

• API-driven architecture and scalable backend workflows

• Database systems and monitoring tools

• Agile development practices and sprint planning

• Technical documentation and issue tracking systems

5. Group Formation

5.1 Group Size

• 2–4 students per team depending on project complexity

5.2 Team Composition

• Teams should contain complementary technical skill sets

5.3 Team Finalization

• No team modification will be allowed after registration approval

6. Project Lifecycle & Timeline

Week	Activity
Week 0	Team formation, onboarding and mentor allocation
Week 1–2	Industry problem statement finalization
Week 3–4	Requirement analysis and architecture planning
Week 5–7	Sprint 1: MVP implementation
Week 8–10	Sprint 2: Core implementation & deployment
Week 11–12	Mid-term evaluation
Week 13–14	Testing, optimization and scalability analysis
Week 15	Deployment validation and documentation
Week 16	Final presentation, viva and industry demonstration

7. Mandatory Deliverables

- **Synopsis Stage**

  • Project Synopsis Report

  • Synopsis PPT

  • Industry Mentorship Confirmation Certificate

  • Requirement Analysis Document

Mid-Term Stage

- Working MVP Demonstration
- Mid-Term PPT
- GitHub Repository Review
- Deployment Progress Report
- Industry Use-Case Validation Certificate

Final Stage

- Final Project Report
- Final PPT
- Complete Source Code Repository
- Deployment Link / Product Demonstration
- 1–2 Minute Product Demo Video
- Industry Completion Certificate
- Plagiarism Report
- User Manual & Technical Documentation

8. Evaluation Scheme (Total: 200 Marks)

Evaluation Component	**Marks**
Synopsis Evaluation	30
Mid-Term Evaluation	50
Final Demonstration & Deployment	80
Industry Mentor Evaluation	20
Faculty Mentor Evaluation	20
Total	200

9. Final Evaluation Rubrics

Criteria	Marks
Industry Problem Relevance	10
Innovation & Technical Complexity	15
Implementation Quality	15
Deployment & Scalability	15
UI/UX & Usability	10
Industry Impact & Real-World Applicability	10
GitHub & Engineering Practices	5
Documentation & Reporting	5
Presentation & Viva	5
Total	90

10. Mandatory Industry Outcome Requirements

Every project must demonstrate at least one of the following:

- Production deployment
- Industry pilot testing
- Usability validation from industry mentor
- Research publication
- Start-up incubation support
- Patent filing
- Client-side implementation or proof-of-concept validation

11. GitHub & Development Workflow Requirements

All projects must maintain:

- private/public GitHub repository
- commit history
- issue tracking
- sprint updates
- documentation

Faculty mentors will regularly monitor:

- commits
- pull requests
- contribution patterns
- implementation progress

12. Plagiarism & Ethical Guidelines

• Maximum plagiarism allowed: ≤ 15%

• Use of copied source code without citation is strictly prohibited

• Use of AI-generated code/tools must be properly documented and justified

• All datasets, APIs and third-party resources must be properly cited

• Plagiarism certificate is mandatory during final submission

13. Expected Final Outcomes

The project must result in:

- A fully functional deployed system/product
- Industry validation or deployment certificate
- Professional technical documentation
- GitHub-based collaborative engineering workflow
- Demonstration of real-world impact and scalability
- Industry-grade engineering practices and deployment standards

INDUSTRY INTERNSHIP

Program Name	**MCA AI ML**	**Semester**	**IV**
Course Name:	**Course Code**	**L-T-P**	**Credits**
Industry Internship	INAML051	0-0-24	12
Type of Course:	INT		
Nature of Course	Industry Immersion & Professional Skill Development		
Evaluation Type	Continuous Evaluation + Final Assessment		
Mode	Full-Time Industry Internship		

Course Perspective

The "Industry Internship / Industrial Training" course is designed to provide students with extensive real-world industrial exposure through full-time professional internships in reputed companies, startups, industries, research organizations or technology-driven enterprises.

The course aims to bridge the gap between academic learning and industry expectations by enabling students to work on real projects, development workflows, business problems and engineering tasks under professional supervision. Students are expected to apply technical knowledge, problem-solving skills, communication abilities and engineering practices in actual workplace environments.

The internship shall primarily focus on:

- practical implementation,
- industry exposure,
- professional conduct,
- workplace collaboration,
- product development workflows,

- software engineering practices,
- business communication and
- organizational learning.

Students are strongly encouraged to secure internships on a **paid basis** to ensure professional accountability, industry engagement and meaningful work exposure.

The internship experience should contribute towards:

- employability enhancement,
- industry readiness,
- technical competency,
- professional networking and
- long-term career growth.

2. Objectives

The internship course aims to:

- Provide practical industry exposure through long-duration professional internships
- Enable students to apply academic concepts to real-world industrial problems
- Develop professional skills including communication, teamwork and workplace ethics
- Expose students to industrial workflows, tools, technologies and project management practices
- Enhance employability, technical competency and career readiness
- Encourage students to contribute to real products, services or organizational processes

3. Course Outcomes (COs)

CO	Course Outcome	Bloom's Level
CO1	Apply technical knowledge to real industrial tasks and projects.	L3 – Apply
CO2	Analyze industry workflows, tools and professional engineering practices.	L4 – Analyze
CO3	Evaluate workplace challenges and develop effective technical solutions.	L5 – Evaluate

CO4	Demonstrate professional communication, teamwork and ethical conduct.	L3 – Apply

4. CO–PO Articulation Matrix

COs \ POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	2	3	1	2	1	-	-	1	1	2
CO2	2	3	2	3	2	1	-	1	1	1	2
CO3	2	3	3	3	2	2	1	1	1	1	2
CO4	1	1	1	-	1	2	2	3	3	3	2

5. CO–PSO Articulation Matrix

COs \ PSOs	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	3	2	2	2
CO2	3	3	3	2	2
CO3	2	3	3	3	2
CO4	2	2	2	2	1

6. Internship Requirements

6.1 Mandatory Internship Duration

• Minimum internship duration shall be **5–6 months full-time**

• The internship must be completed during:

- VIII Semester for B.Tech students
- VI Semester for BCA students

6.2 Internship Type

Students may pursue internships in:

• Product-based companies

• Service-based companies

• Startups

• Research organizations

• Government organizations

• MSMEs and technology firms

• AI/ML, Cloud, Cybersecurity, Data Science, Software Development, IoT or related domains

6.3 Paid Internship Preference

• Students are strongly encouraged to secure internships on a paid/stipend basis

• Paid internships shall be given preference during approval and evaluation processes

• Unpaid internships may be approved only if:

- the company is technically relevant, and
- the work profile is meaningful and industry-oriented

6.4 Work Nature

The internship must involve:

- real project contribution,
- technical development,
- engineering workflows,
- deployment/support activities,
- automation,
- analytics or
- software/product engineering tasks.

Purely non-technical internships shall not be approved.

7. Internship Approval Process

Stage 1: Internship Identification

Students must:

• Apply to companies independently or through university placement support

• Identify suitable internship opportunities relevant to their specialization/domain

Stage 2: Internship Approval

Students must submit:

• Offer Letter / Internship Confirmation Letter

• Company Profile

• Job Description / Role Description

• Internship Duration Details

• Industry Mentor Details

Approval shall be granted by:

- Department Internship Coordinator
- HoD/Dean (where applicable)

Stage 3: Faculty Mentor Allocation

• Each student/team shall be assigned:

- one faculty mentor from the department

• The faculty mentor shall:

- monitor progress,
- conduct review meetings and
- evaluate internship learning outcomes

8. Roles & Responsibilities

8.1 Student Responsibilities

Students must:

• Maintain professional conduct throughout the internship period

• Actively contribute to assigned tasks/projects

• Attend regular review meetings with faculty mentor

• Maintain internship logs and progress records

• Submit reports and required documentation on time

• Follow company policies, confidentiality and ethical practices

8.2 Industry Mentor Responsibilities

Industry mentors shall:

• Assign technical tasks/projects to interns

• Provide technical guidance and workplace mentorship

• Evaluate student performance and contribution

• Provide internship completion certificate and evaluation feedback

8.3 Faculty Mentor Responsibilities

Faculty mentors shall:

• Monitor internship progress regularly

• Conduct periodic review interactions

• Verify learning outcomes and technical involvement

• Guide students in documentation and reporting

• Evaluate presentations and final reports

9. Internship Monitoring & Review Process

Stage	Activity
Week 1–2	Internship registration & approval
Week 3–6	Initial review & task verification
Week 7–10	Mid-term progress review
Week 11–16	Performance monitoring & mentor interactions

Week 17–20	Final report preparation
Week 21–24	Final presentation & viva

10. Mandatory Deliverables

Initial Submission

• Internship offer letter

• Internship role description

• Company profile

• Internship approval form

Mid-Term Submission

• Mid-term internship report

• Task/activity summary

• Faculty mentor review record

• Industry mentor progress feedback

Final Submission

• Final Internship Report

• Internship Completion Certificate

• Industry Mentor Evaluation Form

• Monthly Work Logs / Activity Records

• Presentation PPT

• Internship Experience Video (2–3 minutes)

• Offer/Conversion Letter (if applicable)

11. Internship Report Guidelines

The final report should include:

• Company overview

• Internship objectives

• Technologies/tools used

• Tasks and responsibilities

• Project contribution details

• Learning outcomes

• Challenges faced and solutions implemented

• Professional skills gained

• Conclusion and future scope

12. Evaluation Scheme (Total: 200 Marks)

Evaluation Component	**Marks**
Internship Registration & Initial Documentation	20
Mid-Term Evaluation	40
Faculty Mentor Evaluation	30
Industry Mentor Evaluation	50
Final Internship Report	20
Final Presentation & Viva	40
Total	200

13. Industry Mentor Evaluation Rubrics (50 Marks)

Criteria	**Marks**
Technical Contribution	15
Problem-Solving Ability	10
Professional Conduct	10

Communication & Teamwork	5
Learning Attitude & Adaptability	5
Attendance & Discipline	5
Total	50

14. Final Presentation Evaluation Rubrics

Criteria	**Marks**
Technical Understanding	10
Internship Contribution	10
Practical Learning Outcomes	5
Presentation Skills	5
Problem-Solving & Innovation	5
Documentation Quality	5
Total	40

15. Internship Completion Criteria

To successfully complete the course, students must:

• Complete minimum internship duration requirements

• Obtain internship completion certificate from company

• Receive satisfactory evaluation from industry mentor

• Submit all reports and documents

• Successfully complete final presentation and viva

16. Policy Guidelines

• Fake internships or fabricated certificates shall lead to disciplinary action

• Students leaving internship midway without approval may be declared unsuccessful in the course

• Internship domain must align with the student's program specialization

• Internships may be conducted in hybrid/remote/on-site mode with approval

• Students are encouraged to secure Pre-Placement Offers (PPOs) through internship performance

17. Suggested Internship Domains

• Software Development

• AI & Machine Learning

• Data Science & Analytics

• Cloud Computing & DevOps

• Cybersecurity

• Full Stack Development

• UI/UX Engineering

• IoT & Embedded Systems

• Blockchain Applications

• Computer Vision & NLP Applications

18. Expected Outcomes

By the end of the internship, students should be able to:

• Work professionally in industry environments

• Contribute to real engineering or business problems

• Use modern tools, frameworks and workflows

• Demonstrate technical and professional competency

• Improve employability and placement readiness

K.R. MANGALAM UNIVERSITY

Value Added Course-II

Program Name	MCA AI ML	Semester	IV
Course Name:	**Course Code**	**L-T-P**	**Credits**
VAC-II	VAC	2-0-0	2
Type of Course:	Value Added Course-II		

Standard Operating Procedure (SOP)

1. Introduction

In alignment with the vision and recommendations of the National Education Policy (NEP) 2020 and the guidelines of the University Grants Commission (UGC), the School of Engineering & Technology (SOET) offers Value Added Courses (VAC) as an integral component of holistic and multidisciplinary education. These courses are designed to complement the core academic curriculum by enabling students to acquire additional knowledge, practical skills, ethical values and interdisciplinary competencies essential for personal growth, professional development and lifelong learning.

VAC courses are intended to broaden students' perspectives beyond their primary discipline and help them develop competencies relevant to emerging technologies, societal needs, innovation, entrepreneurship, communication, sustainability, digital literacy, employability and responsible citizenship.

2. Objectives of VAC Courses

The VAC framework aims to:

• Promote multidisciplinary and holistic learning as envisioned in NEP 2020

• Enhance students' employability, practical exposure and professional competencies

• Develop critical thinking, creativity, innovation and problem-solving abilities

• Encourage experiential, skill-based and application-oriented learning

• Provide exposure to emerging technologies, industry trends and societal challenges

• Strengthen ethical values, environmental consciousness and social responsibility

• Support self-learning, lifelong learning and professional growth

3. Nature of VAC Courses

• VAC courses shall be interdisciplinary and skill-oriented in nature

• The courses may focus on:

- emerging technologies,
- innovation,
- entrepreneurship,
- digital skills,
- sustainability,
- communication,
- design thinking,
- ethics,
- problem solving,
- financial literacy,
- professional skills and related areas

• The courses shall emphasize practical implementation, experiential learning and real-world applications rather than purely theoretical delivery

• VAC courses may be offered in:

- offline,
- online,
- blended or
- bootcamp-based modes

4. Course Structure & Delivery

• VAC courses shall generally be short-duration and flexible learning courses

• The delivery model may include:

- workshops,
- practical sessions,
- mini projects,
- case studies,
- simulations,
- industry interactions,
- self-learning activities,
- hackathons,
- assignments and

- collaborative learning exercises

• Industry experts, practitioners, entrepreneurs and domain specialists may be invited for course delivery and mentoring support

• The courses shall encourage active participation, creativity and peer learning

5. Learning Outcomes

VAC courses are expected to help students:

• Acquire additional practical and interdisciplinary competencies

• Develop industry-relevant and future-ready skills

• Improve communication, teamwork and professional confidence

• Apply knowledge in solving real-world and societal problems

• Strengthen innovation, entrepreneurial thinking and digital readiness

• Enhance adaptability and lifelong learning capabilities

6. Assessment & Evaluation

• The assessment methodology shall focus primarily on continuous and competency-based evaluation

• Evaluation components may include:

- assignments,
- practical activities,
- projects,
- quizzes,
- presentations,
- participation,

- case studies,
- demonstrations and
- reflective learning reports

• The evaluation process shall emphasize practical understanding, implementation skills and active engagement rather than rote learning

7. Self-Learning & Skill Development

• VAC courses may include self-learning components through:

- MOOCs,
- certification programs,
- online learning platforms,
- industry learning pathways and
- guided independent learning activities

• Students may be encouraged to complete industry-recognized certifications and skill-development programs aligned with the course objectives

8. Industry & Experiential Learning Integration

• VAC courses may incorporate:

- industry projects,
- field activities,
- innovation challenges,

- hackathons,
- product-building exercises,
- startup-oriented learning and
- practical implementation tasks

• The courses shall promote experiential and application-oriented learning environments aligned with modern industry practices and emerging technology ecosystems

9. Quality Assurance & Continuous Improvement

• VAC courses shall be periodically reviewed and updated based on:

- industry trends,
- technological advancements,
- learner feedback,
- societal requirements and
- employability needs

• The School may seek feedback from:

- students,
- faculty,
- industry experts and
- stakeholders for continuous enhancement of course quality and relevance

10. Alignment with NEP 2020

The VAC framework supports the core principles of NEP 2020 by promoting:

• Holistic and multidisciplinary education

• Skill development and employability enhancement

• Flexibility and learner-centric education

• Experiential and competency-based learning

• Critical thinking and problem-solving abilities

• Digital learning and self-learning culture

• Innovation, entrepreneurship and lifelong learning

• Value-based and socially responsible education

11. Expected Outcomes

Through the VAC ecosystem, students are expected to become:

• Industry-ready and skill-oriented professionals

• Creative thinkers and problem solvers

• Socially responsible and ethically aware individuals

• Self-driven lifelong learners

• Innovators capable of adapting to emerging technologies and evolving global challenges