AC:03/26
Item No.05

The Bombay Salesian Society's

Don Bosco Institute of Technology

(An Autonomous Institute affiliated to University of Mumbai)

Curriculum Structure of Third Year Engineering
Semester V

Department of Information Technology Engineering

(Scheme DB25-V1)
Effective from Academic Year 2026 – 2027

1. Preamble

Don Bosco Institute of Technology, Kurla, Mumbai, proudly celebrates the achievement of autonomous status—an academic milestone that reaffirms our steadfast commitment to excellence, holistic development, and student-centric learning. This autonomy empowers us to craft and implement a curriculum that is forward-looking, contextually relevant, and deeply rooted in our institutional values and the aspirations of our nation.

As an autonomous institution affiliated with the University of Mumbai, DBIT embraces the opportunity to restructure its academic framework in alignment with the University Grants Commission (UGC) guidelines and the National Education Policy (NEP) 2020. This curriculum framework outlines the undergraduate engineering programs for the EXTC, COMP, IT, and MECH branches. It reflects NEP's emphasis on multidisciplinary learning, flexibility, and outcome-based education, while staying true to the Don Bosco educational philosophy.

The curriculum adopts a top-down approach, beginning with the institutional Vision and Mission, which guides the definition of Program Educational Objectives (PEOs) and Program Outcomes (POs). These outcomes are used to shape Course Outcomes (COs) and the content and assessment methods of each course. This ensures that all academic efforts remain aligned with the broader goals of transforming learners into technically sound, ethically responsible and socially aware citizens. Importantly, this curriculum has been shaped through extensive consultations with stakeholders, including industry experts, academic peers, alumni, and students—to ensure that it remains aligned with contemporary industry requirements and societal expectations. Their inputs have been instrumental in designing a framework that bridges the gap between academic learning and practical applicability.

Key Objectives in developing syllabus are:

1. **Develop Strong Technical Foundations:** Equip students with robust knowledge and skills in core engineering domains to solve real-world problems through design, analysis, and innovation.

2.**Foster Research, Innovation, and Entrepreneurship:** Cultivate a spirit of inquiry, critical thinking, and entrepreneurial mindset to promote research-based problem-solving and startup culture.

3.**Enhance Interdisciplinary and Industry-ReadyCompetencies:** Integrate emerging technologies, multidisciplinary learning, and practical exposure to prepare students for dynamic industry requirements and lifelong learning.

4. **Promote Ethical, Sustainable, and Socially Responsible Engineering Practice:** Inculcate ethics, human values, and environmental consciousness to enable students to contribute meaningfully to society and sustainable development.

5. **Empower Communication, Leadership, and Teamwork Abilities:** Strengthen students' soft skills, collaboration, and leadership to perform effectively in diverse professional and global environments.

Academic design includes:

1. A Choice-Based Credit System (CBCS) for flexibility
2. A range of Minor and Honors options to encourage specialization and research
3. Opportunities for field engagement, internships, and experiential learning
4. Emphasis on skill enhancement and future workforce needs
5. Integration of ethical reasoning, social awareness, and environmental consciousness

As an institution inspired by the values of Saint John Bosco, we strive to create a joyful and inclusive learning environment that fosters creativity, curiosity, and compassion. Through this curriculum framework, we renew our pledge to produce graduates who are not only professionally competent but also committed to the greater good of society.

2. Vision and Mission

Vision:
DBIT will be known to have an innovative, enjoyable and holistic learning environment that transforms individuals into socially conscious citizens the Don Bosco way, and will lead in research and entrepreneurship in the area of sustainable technologies.

Mission:
1.To create future engineers who work with honesty and integrity and excel in the use of technology for the benefit of the underprivileged.

2.To train engineers to be innovative problem-solvers and entrepreneurs who engage in research and lifelong learning.

3.To provide a diverse and stimulating environment for staff and students to grow holistically.

3. Curriculum Design Philosophy

The curriculum is structured in alignment with the National Education Policy (NEP) 2020 and UGC guidelines. It follows a top-down approach wherein the institutional Vision and Mission guide the Program Educational Objectives (PEOs) and Program Outcomes (POs). These then shape the Course Outcomes (COs) and form the foundation for course structure, delivery, and assessments.

Key design principles include:

- Emphasis on Outcome-Based Education (OBE) with clear mappings of COs to POs.
- Integration of core technical knowledge with interdisciplinary electives.
- Inclusion of vocational skills, internships, and community engagement.
- Development of entrepreneurship and research aptitude through minor and honors pathways.
- Encouragement of ethical, sustainable, and socially responsible engineering practices.

This approach ensures that the curriculum remains academically rigorous, industry-relevant, and value-driven.

4. Credit Allocation Guidelines

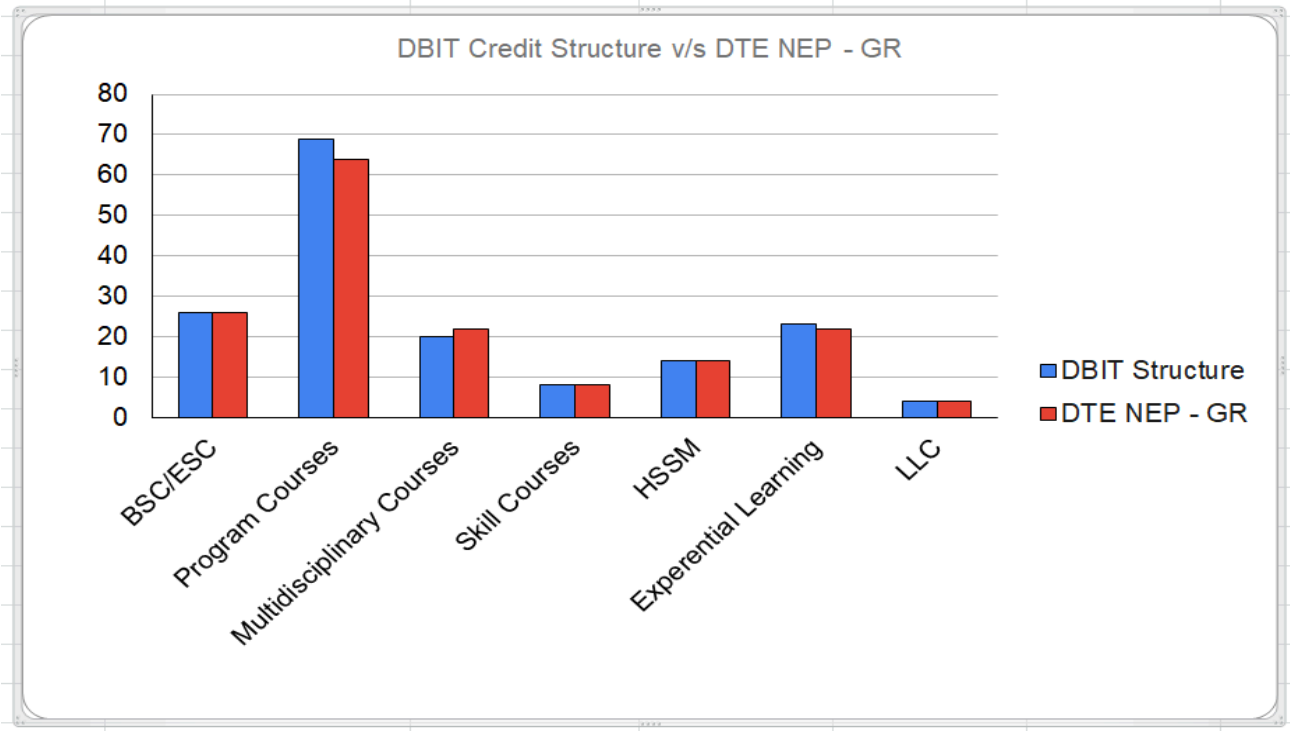
The curriculum is delivered through a structured credit system as follows:

Activity Type	Credit Definition
Theory Course	1 Credit = 15 Contact Hours
Laboratory / Studio / Workshop	1 Credit = 30 Contact Hours
Internship / Field Work	1 Credit = 40 Hours or 2 weeks
Seminar / Group Discussions	1 Credit = 15 Hours
Community Engagement / Field Project	1 Credit = 30 Contact

DBIT Curriculum Credit Structure: (FE to BE)

Semester		I	II	III	IV	V	VI	VII	VIII	Total Credits	DTE Credits
Basic Science Course	BSC/ESC	9	6							15	14-18
Engineering Science Course		7	4							11	12 - 16
Programme Core Course (PCC)	Program Courses		3	16	14	8	8	7		56	44-56
Programme Elective Course (PEC)						3	3	6	6	18	20
Multidisciplinary Minor (MD M)	Multidisciplinary Courses				3	4	4	3		14	14
Open Elective (OE) Other than a particular program						2	2	2		6	8
Vocational and Skill Enhancement Course (VSEC)	Skill Courses	3	3	2						8	8
Ability Enhancement Course (AEC -01, AEC-02)	Humanities Social Science and Management (HSSM)		2			2				4	4
Entrepreneurship/Economics/ Management Courses					2		2			4	4
Indian Knowledge System (IKS)			2							2	2
Value Education Course (VEC)		2		2						4	4
Research Methodology	Experiential Learning Courses					2				2	4
Community. Engagement. Project (CEP)/ Field Project (FP) (Mini - Project)				1	1	1				3	2
Project							3	3		6	4
Internship/ OJT									12	12	12
Co-curricular Courses (CC)	Liberal Learning Courses		1		1		1		1	4	4
Total Credits (Major)		**21**	**21**	**21**	**21**	**22**	**23**	**21**	**19**	**169**	**160- 176**

Graphical Representation :



5. Degree Options and Exit Pathways

Students are offered flexible learning pathways through the following options:

Undergraduate Degree Options:

- B.E (Major) – Minimum 169 Credits
- B.E with Minor/Honors 187 Credits
- B.E with Minor/Honors with Research – 187 Credits

Multiple Entry-Exit Options (Aligned with NEP 2020

Exit Options	Credits Structure
Certificate after Year 1:	• 42 Credits + 08 credits (04 credit Exit course + 04 Summer internship).
Diploma after Year 2:	• 84 credits + 08 credits (04 credit Exit course + 04 Summer internship).
B. Vocational Degree after Year 3:	• 125 credits + 08 credits (04 credit Exit course + 04 Summer internship).

Credits earned are banked in the Academic Bank of Credits (ABC) for lifelong learning flexibility.

Abbreviations Used:

AEC	Ability Enhancement Course
AEL	Ability Enhancement Laboratory
BSC	Basic Science Course
BSL	Basic Science Laboratory
CEP	Community Engagement Project
CC	Co-curricular Courses
CIE	Continuous Internal Evaluation
EEM	Entrepreneurship, Economics and Management
ELC	Experiential Learning Courses
ESC	Engineering Science Course
ESE	End Semester Examination
ESL	Engineering Science Laboratory
FP	Field Project
HSSM	Humanities Social Science and Management
IKS	Indian Knowledge System
L	Lecture
LLC	Liberal Learning Courses
MDM	Multidisciplinary Minor
MSE	Mid Semester Exam
OE	Open Elective
OJT	On Job Training
P	Practical
PCC	Program Core Course
PCL	Program Core Laboratory
PEC	Program Elective Course
T	Tutorial
VEC	Value Education Course

UG Third Year IT Program

Curriculum Scheme and Structure: Semester V									
Course-code	**Course Vertical**	**Course Name**	**Teaching Scheme (Contact Hours)**			**Credits Assigned**			
			L	**P**	**T**	**L**	**P**	**T**	**Total**
25IT5PCC01	PCC	Software Engineering	3	2	--	3	1	--	4
25IT5PCC02	PCC	Web Technology	3	2	--	3	1	--	4
25IT5PECYY	PEC	Program Elective #	2	2	--	2	1	--	3
25XX5MDMYY	MDM	MDM 02@	3	2	--	3	1	--	4
25IL5OEXX	OE	Open Elective 01*	2	--	--	2	--	--	2
25IT5AEC02	AEC	Professional Communication Skills	--	2*+2	--	--	2	--	2
25IT5ELC	ELC	Research Methodology	2	--	--	2	--	--	2
25IT5CEP03	CEP	Mini Project - Prototyping of Systems & Solutions	--	2	--	--	1	--	1
Total			**15**	**14**	**–**	**15**	**7**	**–**	**22**

Students must select **one Program Elective (PEC)** course offered by the department.
@ Students shall opt for **one Multidisciplinary Minor (MDM)** offered by an Engineering Department other than their parent department
*__Open Elective (OE)__ Courses Offered through NPTEL/SWAYAM (Approved by the Internal Committee).

Examination and Assessment Structure

Examination Marking Scheme: Semester V									
Course Code	**Course Vertical**	**Course Name**	**Examination Scheme**						
			CA	**MSE**	**ESE**	**TW**	**Or**	**Pr**	**Total**
25IT5PCC01	PCC	Software Engineering	20	30	50	25	--	25	150
25IT5PCC02	PCC	Web Technology	20	30	50	25	--	25	150
25IT5PECYY	PEC	Program Elective #	20	30	50	25	25	--	150
25XX5MDMYY	MDM	MDM 02@	20	30	50	25	--	--	125
25IL5OEXX	OE	Open Elective 01*	20	--	30	--	--	--	50
25IT5AEC02	AEC	Professional Communication Skills	--	--	--	50	--	--	50
25IT5ELC	ELC	Research Methodology	--	--	--	50	--	--	50
25IT5CEP03	CEP	Mini Project - Prototyping of Systems & Solutions	–	--	--	25	–	25	50
Total			**100**	**120**	**230**	**225**	**25**	**75**	**775**

LIST OF PROGRAM ELECTIVE COURSES (PEC) OFFERED (SEMESTER V)

Course Code	Name of the Program Elective Course
25IT5PEC01	Data Mining & Business Intelligence
25IT5PEC02	Foundation of ER & Immersive Content Development
25IT5PEC03	Blockchain and Cryptocurrency

LIST OF MULTIDISCIPLINE MINOR (MDM) COURSES OFFERED (SEMESTER V)

Course Code	Name of the Multidisciplinary Course
25CE5MDM01	Android Application Development
25ET5MDM01	IoT System Design
25ET5MDM02	Data Acquisition and Industrial Monitoring
25ME5MDM01	Quality Engineering

LIST OF OPEN ELECTIVES (OE) COURSES OFFERED (SEMSETER V)

Course Code	Name of Open Elective Courses
25IL5OE01	Energy Economics and Policy
25IL5OE02	Ethics in Engineering Practice
25IL5OE03	Intellectual Property, Rights and Competition Law
25IL5OE04	Principle of Human Resource Management
25IL5OE05	Patent Search for Engineers and Lawyers
25IL5OE06	Developing Soft Skills and Personality
25IL5OE07	Mastering Speaking and Presentation Skills

HONOUR \ MINOR COURSE : CYBER SECURITY

SEMESTER	Course Code	Course Name	Credit Score
V	25HM5CYSE01	Ethical Hacking	**4**
VI	25HM6CYSE02	Digital Forensic	**4**
VII	25HM7CYSE03	Security MGMT System Vulnerability assessment & Penetration Testing	**6 (Theory 4, Lab 2)**
VIII	25HM8CYSE01	Application Security & Testing	**4**

Assessment Methodology		
Type of Course	**Assessment Tool**	**Marks Distribution**
Theory	**CA-20**	Certification: NPTEL (20 Marks) (Approved by instructor) OR Any two Pedagogies (10 marks each) MCQ /Class Test Case study/Assignment GATE based Assignment Certification: Udemy/Coursera (Approved by instructor) Open Book Test Working model / simulation of a course-based concept.
Theory (VEC)	**CA-50**	Active Participation = 5 marks MCQ /Class Test= 10 marks Instructor Assessment of the Activity carried out by student = 25 marks Assignment = 10 marks
Workshop	**CA-50**	Active Participation = 5 marks Trade 1# = 15 marks Trade 2# = 15 marks Trade 3# = 15 marks # Based on the performance and satisfactory completion of trade wise tasks.

Liberal Learning Courses (LLC)	**CA-50**	Active Participation = 5 marks Mentor Assessment of the Activity carried out by student = 25 marks Cultural Event (Euphoria) Participation = 10 marks Technical Event (Colosseum) Participation = 10 marks
Theory	**MSE**	Question Paper Pattern is as follows: All Questions are compulsory. Q1 A or B - 10 marks Q2 A or B - 10 marks Q3 A or B - 10 marks For each question, A and B should be based on the same CO. MSE should be based on 50% syllabus. Time: 90 minutes (1 hour 30 minutes) Total Marks: 30
Theory	**ESE**	Question Paper Pattern is as follows: All Questions are compulsory. Q1 A or B - 10 marks Q2 A or B - 10 marks Q3 A or B - 10 marks Q4 A or B - 10 marks Q5 A or B - 10 marks For each question, A and B should be based on the same CO. ESE should be based on 30% syllabus of MSE and 70% syllabus after MSE. Time: 120 minutes (2 hours) Total Marks: 50
Course - Laboratory	**TW- 25**	Active Participation (Lab) = 5 marks Laboratory Report = 10 marks Laboratory performance = 10 marks Based on the performance and satisfactory completion of assigned laboratory work
Mini-project	**TW-25**	Active Participation = 5 marks Project Report = 10 marks Progress presentation (Minimum 02) & demonstration = 10 marks
Tutorial	**TW-25**	Active Participation = 5 marks Tutorial Submission = 20 marks Tutorial based on the entire syllabus
Laboratory	**OR-25**	Oral examination will be based on the entire syllabus.

Guideline for All Assessment*

Course Outcomes	Percentage
CO-1, CO-2	20-30
CO-3, CO-4	40-50
CO-5, CO-6	20-30

***Note:** Total Weightage of All CO's should be 100%

<table>
<tr><td rowspan="2">Course Code</td><td rowspan="2">Course Name</td><td colspan="3">Teaching Scheme (Hrs. / Week)</td><td colspan="4">Credits Assigned</td></tr>
<tr><td>L</td><td>P</td><td>T</td><td>L</td><td>P</td><td>T</td><td>Total</td></tr>
<tr><td rowspan="8">25IT5PCC01</td><td rowspan="8">Software Engineering</td><td>3</td><td>2</td><td>–</td><td>3</td><td>1</td><td>–</td><td>4</td></tr>
<tr><td colspan="7">Examination Scheme</td></tr>
<tr><td colspan="3" rowspan="2">Theory</td><td>CA</td><td>MSE</td><td>ESE</td><td>Total</td></tr>
<tr><td>20</td><td>30</td><td>50</td><td>100</td></tr>
<tr><td colspan="3" rowspan="2">Tutorial</td><td>TW</td><td>OR</td><td>PR</td><td>Total</td></tr>
<tr><td>25</td><td>25</td><td>–</td><td>50</td></tr>
<tr><td colspan="3">Total</td><td colspan="4">150</td></tr>
</table>

Pre-Requisite Courses:	25FE1VSEC02 Basic Programming Knowledge C / Python Fundamentals
	25FE2VSEC02 Object-Oriented Programming Concepts

Course Overview:

The Software Engineering (SE) course provides a comprehensive foundation in the principles, methods, and tools used in professional software development. It covers the full lifecycle from requirements engineering and project estimation to design, risk management, testing, and maintenance. In the current era of AI and Large Language Models (LLMs), the course has been enhanced to include AI-augmented development practices, coding agents, LLM-assisted design and testing, and ML-based estimation - equipping students to engineer quality software in an AI-driven world.

Module 1: Introduction to Software Engineering

Covers the software engineering discipline, process models (Waterfall, Agile, Scrum, Kanban), and CMM. Enhanced with AI-augmented development models, the role of coding agents (GitHub Copilot, Cursor, Devin), and how LLMs are reshaping the software process.

Module 2: Requirement Analysis

Addresses requirements engineering including elicitation, validation, SRS preparation, and modelling. Extended to include LLM-driven requirements elicitation, AI-assisted SRS generation, and automated ambiguity detection in requirement documents.

Module 3: Software Estimation and Scheduling

Covers LOC, Function Points, COCOMO II, and project scheduling. Augmented with ML-based effort estimation models that learn from historical project data, offering a modern alternative to classical parametric models.

Module 4: Design Engineering

Explores architectural, component-level, and UI design. Extended with LLM-assisted architectural design critique, generative UI tools (v0.dev, Figma AI), and AI-driven pattern recommendation.

Module 5: Software Risk and Configuration Management

Covers risk management, SCM, SQA, and FTR. Enhanced with AI-powered code review tools (CodeRabbit, GitHub Copilot PR review, Snyk AI) and LLM-based vulnerability detection as modern extensions of quality assurance.

Module 6: Software Testing and Maintenance

Covers testing strategies, debugging, and maintenance. Significantly enhanced with AI-generated test cases (Codium AI, Copilot), autonomous debugging agents (SWE-agent), and LLM-assisted legacy code modernization and reengineering.

This syllabus bridges classical software engineering theory with modern AI-augmented practices, preparing students to work effectively in development environments where LLMs and coding agents are first-class participants in the software lifecycle.

Course Outcomes	After successful completion of the course, the students will be able to	
	CO1	List the fundamental concepts of software engineering, process models, agile methodologies, and describe the role of AI coding agents in modern software development.
	CO2	Explain software requirements engineering processes and use LLM-based tools to assist in elicitation, validation, and SRS generation.
	CO3	Apply software project estimation techniques (LOC, FP, COCOMO II, ML-based) and develop project schedules using timeline charts and earned value analysis.
	CO4	Analyze software design principles and leverage LLM-assisted tools to develop architectural, component-level, and UI designs.
	CO5	Evaluate software risks, apply AI-powered code review and vulnerability detection tools, and implement SCM and SQA strategies.
	CO6	Design test plans, use AI-generated test cases and autonomous debugging agents, and apply LLM-assisted software maintenance and reengineering strategies.

Syllabus:

Module No.	Unit No.	Topics	Hours
	Introduction to Software Engineering		

1	**After completion of the module the student will be able to :** 1. Define software, software engineering, software process, and CMM levels. 2. Compare prescriptive process models: Waterfall, V-model, Incremental, and Evolutionary. 3. Explain Agile methodologies: XP, Scrum, and Kanban. 4. Describe AI-augmented development models and the role of coding agents in modern SDLC.		7
	1.1	Nature of Software, Software Engineering, Software Process, Capability Maturity Model (CMM)	
	1.2	Generic Process Model, Prescriptive Process Models: Waterfall Model, V-Model, Incremental Process Models, Evolutionary Process Models, Concurrent Models	
	1.3	Agile Process, Agility Principles, Extreme Programming (XP), Scrum, Kanban Model	
	1.4	AI-Augmented Development Models: Role of Coding Agents (GitHub Copilot, Cursor, Devin), Prompt-Driven Development, Vibe Coding, Ethical and IP Considerations, AI SDLC (Artificial Intelligence Software Development Life Cycle), The BMad Method (Build More Architect Dreams).	
	Self Study Topics: Personal and Team Process Models, AI Coding Agents in the SDLC - Are LLMs Junior Developers? Impact on software process maturity.		
2	**Requirement Analysis**		9
	After completion of the module the student will be able to : 1. Identify and classify functional and non-functional software requirements. 2. Explain the user-system requirement engineering process including feasibility studies. 3. Apply elicitation, validation, and management techniques for software requirements. 4. Use LLM-based tools to assist in requirements elicitation and SRS generation.		
	2.1	Software Requirements: Functional & Non-Functional Requirements, User-System Requirement Engineering Process, Feasibility Studies	
	2.2	Requirements Elicitation, Validation & Management, Software Prototyping, S/W Documentation	

	2.3	Analysis and Modelling, Software Requirement Specification (SRS)	
	2.4	LLM-Assisted Requirements Engineering: Using ChatGPT / Claude for elicitation, Automated SRS generation from user stories, AI-based ambiguity detection in requirement documents	
	Self Study Topics: Prioritizing requirements using Kano Diagram – real life application case study, AI-Assisted SRS Generation: Prompt patterns for requirements extraction, validation, and conflict detection using LLMs.		
3	**Software Estimation and Scheduling**		7
	After completion of the module the student will be able to : 1. Apply software metrics for process and project measurement. 2. Use LOC, Function Points, and COCOMO II for project size and effort estimation. 3. Develop project schedules using timeline charts and Earned Value Analysis. 4. Compare ML-based effort estimation with classical parametric models.		
	3.1	Management Spectrum, 3Ps (People, Product, Process), Process and Project Metrics	
	3.2	Software Project Estimation: LOC, Function Points (FP), Empirical Estimation Models, COCOMO II Model, Specialized Estimation Techniques, Object-Based Estimation, Use-Case Based Estimation	
	3.3	Project Scheduling: Defining a Task Set, Timeline Charts, Tracking the Schedule, Earned Value Analysis	
	3.4	ML-Based Effort Estimation: Regression and Neural Network models trained on historical project data (PROMISE dataset), AI-powered project management tools (Jira AI, GitHub Projects AI)	
	Self Study Topics: Cost Estimation Tools and Techniques, Typical Problems with IT Cost Estimates, Machine Learning for Software Effort Estimation: Story-point prediction with NLP, benchmark comparison with COCOMO II.		
4	**Design Engineering**		8
	After completion of the module the student will be able to : 1. Apply design process guidelines, quality attributes, and fundamental design concepts. 2. Develop architectural designs using patterns, views, and application architecture models. 3. Create component-level designs using class-based components.		

	4. Use LLM-assisted tools and generative UI platforms to support design decisions and UI generation.		
	4.1	Design Process & Quality, Design Concepts, The Design Model, Pattern-Based Software Design	
	4.2	Architectural Design: Design Decisions, Views, Patterns, Application Architectures, Modeling	
	4.3	Component Level Design: Components, Designing Class-Based Components, Conducting Component-Level Design	
	4.4	User Interface Design: The Golden Rules, Interface Design Steps & Analysis, Design Evaluation	
	4.5	LLM-Assisted Design: Using AI as an architectural sparring partner, Generative UI tools (v0.dev, Figma AI, Locofy), AI-driven design pattern recommendation, Risks of AI-generated architecture	
	Self Study Topics: Refinement, Aspects, Refactoring **AI / Emerging Technology:** AI-Generated UI and Component Design: v0.dev, Figma AI - opportunities, quality evaluation, and limitations.		
5	**Software Risk and Configuration Management**		7
	After completion of the module the student will be able to : 1. Identify and assess software risks and develop RMMM strategies. 2. Apply SCM processes using SCM repositories. 3. Evaluate software quality using SQA tasks, metrics, and reliability models. 4. Apply AI-powered code review and vulnerability detection tools as extensions of SQA.		
	5.1	Risk Identification, Risk Assessment, Risk Projection, Risk Mitigation Monitoring and Management (RMMM)	
	5.2	Software Configuration Management, SCM Repositories, SCM Process	
	5.3	Software Quality Assurance: SQA Tasks and Plan, Metrics, Software Reliability, Formal Technical Review (FTR), Walkthrough	
	5.4	AI-Powered Quality Assurance: Automated Code Review (CodeRabbit, GitHub Copilot PR review), LLM-based Vulnerability Detection (Snyk AI, Semgrep), Hallucination risks in AI-assisted QA	

	Self Study Topics: Configuration Management for WebApps, LLMs as Automated Reviewers: AI-driven FTR, comparing AI vs. human review quality, hallucination risks in security-critical code review.		
6	**Software Testing and Maintenance**		7
	After completion of the module the student will be able to : 1. Apply testing strategies for conventional, OO, and web-based software. 2. Conduct validation testing, system testing, and structured debugging. 3. Use AI-generated test cases and autonomous debugging agents. 4. Apply LLM-assisted reengineering techniques for legacy code modernization.		
	6.1	Testing: Software Quality, Strategic Approach, Strategic Issues, Testing Strategies for Conventional Software, Object-Oriented Software, and Web Apps	
	6.2	Validating Testing, System Testing, Art of Debugging	
	6.3	Software Maintenance: Supportability, Reengineering, Business Process Reengineering, Software Reengineering, Reverse Engineering, Restructuring, Forward Engineering	
	6.4	AI-Augmented Testing and Maintenance: AI-generated test cases (Codium AI, GitHub Copilot Tests), Autonomous Debugging Agents (SWE-agent), LLM-assisted legacy code modernization, Prompt-based forward engineering	
	Self Study Topics: Test Strategies for WebApps, Autonomous Testing and Debugging Agents: LLMs for test case generation, bug localization, and AI-assisted legacy code modernization.		
		Total	**45**

Suggested List of Experiments:

Expt. No.	List of Experiments
1	Prepare a Use Case Diagram and Activity Diagram for a given software problem **Objective:** To model functional requirements using UML behavioral diagrams. **Outcome:** Students will identify actors, use cases, and workflows and represent them using UML Use Case and Activity Diagrams.

2	Prepare a Class Diagram and Sequence Diagram for a given software problem. **Objective:** To model the static and dynamic structure of a software system using UML structural and interaction diagrams. **Outcome:** Students will identify classes, attributes, methods, and message interactions and represent them using Class and Sequence Diagrams.
3	Prepare a Software Requirement Specification (SRS) document for a given case study. **Objective:** To document functional and non-functional requirements following IEEE SRS standards. **Outcome:** Students will prepare a complete and unambiguous SRS document for a real-world software problem.
4	Develop a Project Plan and Schedule using MS Project or any equivalent project management tool. **Objective:** To plan a software project by defining tasks, estimating durations, assigning resources, and generating a Gantt chart. **Outcome:** Students will create a project schedule with task dependencies, milestones, and critical paths using a project management tool.
5	Calculate Function Point (FP) and LOC estimation for a given software project description. **Objective:** To apply Function Point Analysis and Lines of Code metrics to estimate software size and effort. **Outcome:** Students will compute unadjusted and adjusted function points and derive effort/cost estimates using productivity data.
6	Apply the COCOMO II model to estimate effort, duration, and cost for a given software project. **Objective:** To use COCOMO II for empirical project estimation using scale factors and cost drivers. **Outcome:** Students will calculate effort in person-months, project duration, and team size using the COCOMO II post-architecture model.
7	Design and implement test cases for a given software module using JUnit / PyTest or equivalent. **Objective:** To apply unit testing by designing black-box and white-box test cases and executing them using a testing framework.

	Outcome: Students will write and run unit tests, verify correctness, and report code coverage.
8	Perform integration testing and system testing for a given application. **Objective:** To verify that integrated modules work correctly together and validate the system against the SRS. **Outcome:** Students will design integration stubs and drivers, execute system-level test cases, and document test results.
9	Use Git for Software Configuration Management: branching, merging, commit history, and conflict resolution. **Objective:** To practice SCM principles using Git for tracking changes and managing code baselines. **Outcome:** Students will initialize a Git repository, create branches, merge changes, resolve conflicts, and maintain a clean commit history.
10	Simulate an Agile/Scrum sprint: product backlog, sprint planning, daily standup, and retrospective. **Objective:** To experience the Agile Scrum framework by enacting key ceremonies and artefacts for a mini software project. **Outcome:** Students will create a product backlog, estimate story points, plan a sprint, and conduct a retrospective.
11	Use an LLM (ChatGPT / Claude) to generate an SRS document from a brief problem statement; evaluate and compare it with a manually written SRS. **Objective:** To experience AI-assisted requirements engineering and critically evaluate the quality, completeness, and ambiguities in LLM-generated SRS documents. **Outcome:** Students will be able to prompt an LLM for structured SRS output, identify gaps or hallucinations, and produce a corrected hybrid SRS that combines AI output with domain judgment.
12	Use GitHub Copilot / Codium AI to auto-generate unit tests for a given software module and compare AI-generated test coverage with manually written tests. **Objective:** To evaluate the effectiveness of AI-generated test suites versus human-authored test cases in terms of coverage, correctness, and edge-case handling. **Outcome:** Students will run both test suites, generate coverage reports, document differences, and assess when AI-generated tests are sufficient versus when manual tests are essential.

13	Apply an LLM to refactor / re-engineer a given legacy code snippet; document before-and-after quality metrics (readability, complexity, maintainability index). **Objective:** To experience LLM-assisted forward engineering and evaluate the quality improvement in modernized code produced by an AI agent. **Outcome:** Students will feed legacy code to an LLM with targeted prompts, apply the suggested refactoring, and measure improvements using a static analysis tool (e.g., Pylint, SonarQube Lite).
14	Build a simple ML-based software effort estimation model (Linear Regression / Random Forest) trained on the PROMISE dataset and compare its predictions against COCOMO II. **Objective:** To explore ML-based alternatives to classical parametric estimation and understand the role of training data quality in prediction accuracy. **Outcome:** Students will preprocess the PROMISE dataset, train and evaluate a regression model, compare predictions with COCOMO II estimates, and report Mean Absolute Error (MAE) for both approaches.
15	Use an AI-powered code review tool (CodeRabbit / GitHub Copilot PR Review / Snyk) to review a given code module; document findings and compare with a manual peer review. **Objective:** To evaluate AI-assisted SQA tools as a modern alternative to traditional Formal Technical Reviews (FTR) and assess their coverage of security, style, and logic issues. **Outcome:** Students will submit code to an AI review tool, document flagged issues by category (security, performance, style), perform a parallel manual review, and critically compare the two outputs.

Assessment Methodology:

CA-20	Certification: NPTEL (20 Marks) (Approved by instructor) OR Any two Pedagogies (10 marks each) * MCQ / Class Test * Case Study / Assignment * GATE Based Assignment * Certification: Udemy / Coursera / NPTEL (Approved by instructor) * Open Book Test * Working model / AI-assisted prototype demonstrating a course-based concept.
MSE	All Questions are compulsory.

	* Q1 A or B – 10 marks * Q2 A or B – 10 marks * Q3 A or B – 10 marks * A and B for each question should be from the same CO. * MSE should be based on 50% syllabus. Time: 90 minutes. Total Marks: 30.
ESE	All Questions are compulsory. * Q1–Q5 A or B – 10 marks each * A and B for each question should be from the same CO. * ESE based on 30% MSE syllabus + 70% post-MSE syllabus. * Time: 120 minutes. Total Marks: 50.
TW-25	Active Participation = 5 marks Laboratory Journal / Term Work Submission = 20 marks
OR-25	Oral examination will be based on experiments (including AI/LLM experiments) and term work performed during laboratory sessions.

Textbooks:

- Roger S. Pressman & Bruce R. Maxim, Software Engineering: A Practitioner's Approach, 8th Edition, McGraw Hill Education.
- Rajib Mall, Fundamentals of Software Engineering, 4th Edition, PHI Learning (Prentice Hall India).
- Ian Sommerville, Software Engineering, 10th Edition, Addison-Wesley.
- Pankaj Jalote, An Integrated Approach to Software Engineering, 3rd Edition, Springer / Narosa.

References:

- Shari Lawrence Pfleeger & Joanne M. Atlee, Software Engineering: Theory and Practice, 4th Edition, Pearson.
- Stephen R. Schach, Object-Oriented & Classical Software Engineering, 8th Edition, McGraw Hill.
- Carlo Ghezzi, Mehdi Jazayeri & Dino Mandrioli, Fundamentals of Software Engineering, Prentice Hall.
- Mark Harman et al., Search-Based Software Engineering – AI in Testing and Optimization (Reference Papers, IEEE/ACM).

Useful Links:

- https://nptel.ac.in/courses/106105182 (NPTEL – Software Engineering)
- https://www.coursera.org/learn/software-processes (Coursera – Software Development Processes)
- https://www.youtube.com/watch?v=wEr6mwquPLY
- https://github.com/features/copilot (GitHub Copilot – AI Coding Agent)

- https://www.codium.ai/ (Codium AI – AI Test Generation)
- https://swe-agent.com/ (SWE-agent – Autonomous Debugging Agent)
- https://v0.dev/ (v0.dev – Generative UI Design)
- https://paperswithcode.com/task/software-effort-estimation (ML-based Effort Estimation Papers
- https://docs.bmad-method.org/

<table>
<tr><th>Course Code</th><th>Course Name</th><th colspan="3">Teaching Scheme (Hrs. / Week)</th><th colspan="4">Credits Assigned</th></tr>
<tr><td rowspan="8">25IT5PCC02</td><td rowspan="8">Web Technology</td><td>L</td><td>P</td><td>T</td><td>L</td><td>P</td><td>T</td><td>Total</td></tr>
<tr><td>3</td><td>2</td><td>–</td><td>3</td><td>1</td><td>–</td><td>4</td></tr>
<tr><td colspan="7">Examination Scheme</td></tr>
<tr><td colspan="2">Theory</td><td>CA</td><td>MSE</td><td>ESE</td><td colspan="2">Total</td></tr>
<tr><td colspan="2"></td><td>20</td><td>30</td><td>50</td><td colspan="2">100</td></tr>
<tr><td colspan="2">Tutorial</td><td>TW</td><td>OR</td><td>PR</td><td colspan="2">Total</td></tr>
<tr><td colspan="2"></td><td>25</td><td>–</td><td>25</td><td colspan="2">50</td></tr>
<tr><td colspan="2">Total</td><td colspan="5">150</td></tr>
</table>

Pre-Requisite Courses:	25FE2VSEC01:Engineering Skills Workshop II (HTML, CSS, JavaScript Fundamentals)
	25FE2VSEC02:Object Oriented Programming using Python (C / Java / Python)

Course Overview: The Web Technology (WT) course provides a comprehensive and industry-relevant foundation in modern front-end and back-end web development. Starting with JavaScript fundamentals and advancing through TypeScript, the course introduces dynamic web interactions via AJAX and evolves into full-stack development using the MERN stack (MongoDB, Express.js, React.js, Node.js). Students gain hands-on expertise in ES6+ features, building RESTful APIs, and creating responsive, component-based user interfaces.**Module 1 – Web Programming Fundamentals :** Covers the fundamentals of web technologies including Internet and World Wide Web architecture, client-server communication, HTTP/HTTPS protocols, DNS, REST APIs, web application architectures, HTML5, CSS3, responsive web design and the modern web development ecosystem.

Module 2 – JavaScript and TypeScript : Covers JavaScript fundamentals including variables, data types, functions, scope, closures, objects, arrays, ES6+ features, and asynchronous programming concepts. TypeScript topics include static typing, interfaces, classes, modules, generics, and compiler configuration for scalable application development.

Module 3 – Asynchronous Web Programming : Covers AJAX fundamentals, XMLHttpRequest, Fetch API, JSON processing, Promises, async/await patterns, error handling,

CORS, and integration with RESTful web services for building dynamic and interactive web applications.

Module 4 – React.js Frontend Development : Covers React.js fundamentals including JSX, functional components, props, state management, event handling, React Hooks, forms, React Router, API integration, and best practices for developing responsive Single Page Applications (SPAs).

Module 5 – Express.js & MongoDB Backend Development : Covers Node.js architecture, Express.js framework, routing, middleware, RESTful API development, MongoDB database operations, Mongoose ODM, schema design, authentication using JWT, password hashing, and an introduction to enterprise backend development using NestJS.

Module 6 – Enterprise Full-Stack Development : Covers MERN stack architecture, frontend-backend integration, CRUD application development, authentication and security, state management concepts, API testing, version control using Git and GitHub, and deployment of complete full-stack web applications on cloud platforms.

The WT course bridges theoretical web concepts with practical full-stack development skills, preparing students for modern industry roles in web development and software engineering.

Course Outcomes	**After successful completion of the course, the students will be able to**	
	CO1	Recall the fundamental concepts of web technologies, Internet architecture, client-server communication, JavaScript, and modern web development frameworks.
	CO2	Explain the principles of JavaScript, TypeScript, asynchronous programming, React.js, and server-side web technologies used in modern web application development.
	CO3	Apply JavaScript, TypeScript, React.js, Node.js, Express.js, and MongoDB concepts to develop interactive web applications and RESTful APIs.
	CO4	Analyze the architecture and integration of frontend, backend, and database components to develop scalable and responsive MERN stack applications.
	CO5	Evaluate different web development approaches, authentication mechanisms, API communication methods, and deployment strategies for secure and efficient web applications.

	CO6	Design and develop a complete MERN stack web application by integrating React.js, Node.js, Express.js, and MongoDB using modern software engineering and deployment practices.

Syllabus:

Module No.	Unit No.	Topics	Hours
1		**Web Programming Fundamentals**	7
		After completion of the module the student will be able to: 1. Recall the fundamental concepts of web technologies, Internet architecture, web browsers, and communication protocols used in web applications. 2. Explain the working of the World Wide Web, HTTP/HTTPS protocols, DNS, REST APIs, and modern web communication mechanisms. 3. Differentiate between traditional and modern web application architectures, including static and dynamic websites, multi-page and single-page applications. 4. Apply the fundamental concepts of HTML5, CSS3, responsive web design, and modern JavaScript in the development of interactive web applications.	
	1.1	Web Communication Fundamentals : Working of the World Wide Web (WWW), Working of Web Browsers, Client–Server Communication, HTTP Protocol, HTTPS and TLS, Domain Name System (DNS), Uniform Resource Locator (URL), Uniform Resource Identifier (URI), Document Object Model (DOM), Introduction to XML and JSON, Introduction to RESTful APIs	
	1.2	Evolution of Modern Web Technologies: Evolution of the Web (Web 1.0, Web 2.0, Web 3.0 and Web X.0), Internet and World Wide Web (WWW), Website vs Web Application, Static vs Dynamic Websites, Multi-Page Applications (MPA) vs Single-Page Applications (SPA), Frontend, Backend and Full-Stack Development, Web Standards, Overview of Modern Web Development Ecosystem.	

	1.3	Modern Web Development Ecosystem: HTML5 and CSS3 Overview, Responsive Web Design Concepts, Introduction to JavaScript and ECMAScript (ES6+), Package Managers (NPM), Development Environment (VS Code, Node.js), Modern Frontend Frameworks (React, Angular and Vue), Enterprise Backend Frameworks (NestJS).	
	Self Study Topics: Progressive Web Applications (PWA); Browser Developer Tools (Chrome DevTools).		
2	**JavaScript & TypeScript**		**7**
	After completion of the module the student will be able to: 1. Define JavaScript variables, data types, scope, closures, and the prototype chain. 2. Apply object-oriented programming using ES6 classes, inheritance, and DOM manipulation. 3. Explain TypeScript's static typing system including interfaces, enums, and generics. 4. Use TypeScript decorators and advanced types to build scalable and type-safe applications.		
	2.1	JavaScript Fundamentals: Variables (var, let, const), Data Types, Type Coercion, Operators, Control Structures, Functions, Closures, Hoisting, Scope Chain, IIFE	
	2.2	JavaScript OOP: Objects, Prototypes, Prototype Chain, ES6 Classes, Inheritance, Encapsulation, Polymorphism, DOM Manipulation, Event Handling, Event Bubbling	
	2.3	TypeScript: Static Typing, Type Annotations, Interfaces, Type Aliases, Enums, Generics, Decorators, Modules, Compilation Pipeline, tsconfig.json configuration	
	Self Study Topics: JavaScript Design Patterns (Module, Singleton, Observer, Factory); TypeScript with React – Type-safe component props, state, and custom hooks		
	Asynchronous Web Programming		

	After completion of the module the student will be able to: 1. Explain the role of AJAX in building dynamic web pages without full page reloads. 2. Use XMLHttpRequest and Fetch API to perform HTTP requests to web servers. 3. Apply Promises, async/await, and structured error handling in asynchronous JavaScript. 4. Integrate third-party REST APIs and process JSON data in web applications.		
	3.1	AJAX Fundamentals: Synchronous vs Asynchronous Processing, XMLHttpRequest (XHR) Object, readyState, onreadystatechange, HTTP Methods (GET, POST, PUT, DELETE), JSON and XML Data Formats	
	3.2	Fetch API: fetch(), Request/Response Objects, Headers, Handling JSON, CORS (Cross-Origin Resource Sharing), Preflight Requests, Error Handling with try/catch	
	3.3	Asynchronous Patterns: Callbacks, Callback Hell, Promises (resolve, reject, chaining, .then/.catch/.finally), async/await, Promise.all(), Promise.race(), Introduction to WebSockets and Real-time Data	
	Self Study Topics: WebSockets and Real-time Web Applications		
4	**React.js Frontend Development**		8
	After completion of the module the student will be able to: 1. Explain the fundamentals of React.js, component-based architecture, and Single Page Applications (SPAs). 2. Develop reusable React components using JSX, props, state, and event handling. 3. Build interactive web applications using React Hooks, forms, routing, and API integration. 4. Create responsive React applications by following best practices and component-based design principles.		
	4.1	React Fundamentals: Installation, Installing libraries, Folder and file structure, Components, Component lifecycle, State and Props, React	

		Router and Single page applications, UI design, Forms, Events, Animations, Best practices.	
	4.2	Functional components: Refs, Use effects, Hooks, Flow architecture, Model-View Controller framework, Flux, Bundling the application. Web pack.	
	4.3	Component Reusability: Project Structure, Basic UI Design using Bootstrap/Tailwind CSS (Overview), Error Handling, React Developer Tools, Introduction to Application Build and Deployment.	
	Self Study Topics: React Native		
5	Express.js & MongoDB Backend Development		8
	After completion of the module the student will be able to: 1. Explain the architecture and features of Node.js, Express.js, MongoDB, and the role of modern backend frameworks in web application development. 2. Develop RESTful web services using Express.js by implementing routing, middleware, and CRUD operations. 3. Implement MongoDB database operations using Mongoose ODM and integrate them with Express.js applications. 4. Develop secure and scalable backend applications using authentication techniques.		
	5.1	Node.js and Express.js Fundamentals : Introduction to Server-Side Web Development, Node.js Runtime Environment, Event-Driven Architecture, Node Package Manager (NPM), Express.js Framework, Project Structure, Routing, Middleware, Request and Response Objects, RESTful API Development, CRUD Operations, Error Handling..	
	5.2	MongoDB and Database Integration : Introduction to NoSQL Databases, MongoDB Architecture, Collections and Documents, CRUD Operations, Query Operators, MongoDB Atlas (Overview), Introduction to Mongoose ODM, Schema Design, Data Validation, Integrating MongoDB with Express.js Applications.	

	5.3	Authentication and Enterprise Backend Development : JWT Authentication, User Registration and Login, Password Hashing using bcrypt, Environment Variables (.env), API Testing using Postman, Introduction to NestJS, NestJS Project Structure, Modules, Controllers, Services, Dependency Injection (Overview), Express.js vs. NestJS (Overview). .	
	Self Study Topics: Next.js for Server-Side Rendering (SSR)		
6	**Enterprise Full-Stack Development**		8
	After completion of the module the student will be able to: 1. Recall the components and architecture of enterprise full-stack web applications. 2. Explain the workflow and integration of React, NestJS, and MongoDB in enterprise application development. 3. Evaluate authentication, security, and deployment strategies for enterprise web applications. 4. Create and deploy a scalable enterprise full-stack web application following modern development practices.		
	6.1	Full Stack Integration : Full Stack Architecture, Frontend–Backend Communication, REST API Integration using Fetch API/Axios, Connecting React with Express.js Backend, CRUD Operations, Project Structure, Client–Server Data Flow.	
	6.2	Application Development and Security : User Authentication using JWT, Protected Routes, Form Validation, State Management using Context API (Overview), Environment Variables, Error Handling, Basic Application Security Best Practices.	
	6.3	**Testing, Version Control and Deployment :** API Testing using Postman, Version Control using Git and GitHub, Application Build Process, Frontend Deployment using Vercel, Backend Deployment using Render, Project Demonstration and Documentation.	
	Self Study Topics: Progressive Web Applications (PWA); Introduction to Serverless Web Applications.		
		Total	**45**

Suggested List of Experiments:

Expt. No.	List of Experiments
1	**JavaScript Programming using ES6 Features** **Objective:** To understand modern JavaScript programming concepts including functions, objects, closures, and ES6 features. **Outcome:** Students will be able to develop JavaScript programs using closures, higher-order functions, arrow functions, destructuring, and modules.
2	**TypeScript Application using Classes, Interfaces and Generics** **Objective:** To learn static typing and object-oriented programming concepts using TypeScript. **Outcome:** Students will be able to develop type-safe applications using TypeScript classes, interfaces, generics, and modules.
3	**Asynchronous Web Application using Fetch API and Async/Await** **Objective:** To implement asynchronous communication between client and server using modern JavaScript techniques. **Outcome:** Students will be able to consume REST APIs using Fetch API, Promises, async/await, and process JSON data dynamically.
4	**React Single Page Application using Functional Components and Hooks** **Objective:** To develop interactive Single Page Applications using React.js. **Outcome:** Students will be able to create reusable functional components using JSX, Props, State (useState), and useEffect Hook.
5	**React Application with Routing and Form Handling** **Objective:** To implement client-side routing and interactive forms in React applications. **Outcome:** Students will be able to build Single Page Applications using React Router, forms, event handling, and API integration using Axios/Fetch.
6	**RESTful API Development using Node.js and Express.js** **Objective:** To develop server-side applications using Express.js and REST architecture. **Outcome:** Students will be able to create RESTful APIs with routing, middleware, CRUD operations, and structured error handling.

7	**MongoDB Database Integration using Mongoose ODM** **Objective:** To integrate MongoDB with Express.js applications using Mongoose. **Outcome:** Students will be able to design MongoDB schemas, perform CRUD operations, validate data, and connect MongoDB with Express.js applications.
8	**User Authentication using JWT and Password Encryption** **Objective:** To implement secure authentication mechanisms in web applications. **Outcome:** Students will be able to implement user registration, login, password hashing using bcrypt, JWT authentication, and protected API routes.
9	**MERN Stack CRUD Application** **Objective:** To integrate React.js, Express.js, Node.js, and MongoDB into a complete full-stack application. **Outcome:** Students will be able to develop a full-stack MERN application supporting CRUD operations, frontend-backend communication, and database integration.
10	**Capstone MERN Stack ProjectObjective:** To design, develop, test, and deploy a complete MERN stack web application using industry-standard development practices. **Outcome:** Students will be able to develop, test, version-control using Git and GitHub, and deploy a secure MERN stack application on cloud platforms such as Vercel and Render.

Assessment Methodology:

CA-20	Certification: NPTEL (20 Marks) (Approved by instructor) OR Any two Pedagogies (10 marks each): • MCQ / Class Test • Case Study / Assignment • GATE Based Assignment • Certification: Udemy / Coursera (Approved by instructor) • Open Book Test • Working model / simulation of a course-based concept

MSE	Question Paper Pattern is as follows – All Questions are compulsory: • Q1 A or B – 10 marks • Q2 A or B – 10 marks • Q3 A or B – 10 marks • For each question, A and B should be based on the same CO. • MSE should be based on 50% of the syllabus. • Time: 90 minutes (1 hour 30 minutes) • Total Marks: 30
ESE	Question Paper Pattern is as follows – All Questions are compulsory: • Q1 A or B – 10 marks • Q2 A or B – 10 marks • Q3 A or B – 10 marks • Q4 A or B – 10 marks • Q5 A or B – 10 marks • For each question, A and B should be based on the same CO. • ESE covers 30% of MSE syllabus and 70% of syllabus after MSE. • Time: 120 minutes (2 hours) • Total Marks: 50
TW-25	• Active Participation = 5 marks • Laboratory Journal / Term Work Submission = 20 marks
OR-25	Oral examination will be based on the experiments and term work performed by the students during laboratory sessions.

Textbooks:

1. David Flanagan, JavaScript: The Definitive Guide, 7th Edition, O'Reilly Media.
2. Boris Cherny, Programming TypeScript: Making Your JavaScript Applications Scale, O'Reilly Media.
3. Ethan Brown, Web Development with Node and Express: Leveraging the JavaScript Stack, 2nd Edition, O'Reilly Media.
4. Shannon Bradshaw, Eoin Brazil and Kristina Chodorow, MongoDB: The Definitive Guide, 3rd Edition, O'Reilly Media.
5. Alex Banks and Eve Porcello, Learning React: Modern Patterns for Developing React Apps, 2nd Edition, O'Reilly Media.

References:

1. Jon Duckett, JavaScript and jQuery: Interactive Front-End Web Development, Wiley.

2. Marijn Haverbeke, Eloquent JavaScript: A Modern Introduction to Programming, 3rd Edition, No Starch Press.
3. Kyle Simpson, You Don't Know JS (YDKJS) Book Series, O'Reilly Media / GitHub (Open Access).
4. Stoyan Stefanov, JavaScript Patterns: Build Better Applications with Coding and Design Patterns, O'Reilly Media.
5. Robin Wieruch, The Road to React: Your Journey to Master Plain yet Pragmatic React.js, Self-published (leanpub.com).

Useful Links:

1. https://developer.mozilla.org/en-US/docs/Web/JavaScript (MDN Web Docs – JavaScript Reference)
2. https://www.typescriptlang.org/docs/ (TypeScript Official Documentation)
3. https://nodejs.org/en/docs/ (Node.js Official Documentation)
4. https://mongoosejs.com/docs/ (Mongoose ODM Documentation)
5. https://react.dev/ (React.js Official Documentation)
6. https://redux-toolkit.js.org/ (Redux Toolkit Official Documentation)
7. https://nptel.ac.in/courses/106/104/106104196/ (NPTEL – Introduction to Modern Application Development)
8. https://www.coursera.org/specializations/full-stack-react (Coursera – Full Stack Web Development with React)

Course Code	Course Name	Teaching Scheme (Hrs. / Week)			Credits Assigned			
25CE5MDM01	Android Application Development	L	P	T	L	P	T	Total
		3	2	_	3	1	_	4
		Examination Scheme						
			CA	MSE	ESE	TW	OR	PR	Total
		Theory	20	30	50	-	-	-	100
		Lab	-	-	-	25	-	-	25
		Total			125				

Pre-Requisite Courses:	25FEAECO1: JAVA PROGRAMMING

Course Overview

This course introduces students to Android application development, covering the Android ecosystem, user interface design, application components, data storage, background services, networking, and multimedia integration. Through hands-on activities and practical exercises, students develop the skills needed to design, build, test, and deploy basic Android applications using industry-relevant tools and practices.

Module 1 – Introduction to Android

This module builds the foundational knowledge required to begin Android development. Students will explore the history and growth of the Android ecosystem, understand the overall architecture of Android applications, and learn to configure the development environment including Android Studio and the SDK. The module covers project structure, the Android Manifest file, running apps on virtual and physical devices, and the Activity lifecycle. The SOLID design principles are also introduced to encourage good software design habits from the outset.

Module 2 – Android UI Fundamentals

This module focuses on designing and building effective user interfaces using Android's XML-based layout system. Students will work with the most commonly used UI components including TextView, EditText, Button, and ImageView, and will learn to structure screens using LinearLayout, RelativeLayout, and ConstraintLayout. Topics also include handling screen orientation changes, using the action bar, displaying Toast notifications, and applying basic UI

design principles to build clean and intuitive interfaces.

Module 3 – Activities, Intents & Fragments

This module explores the structural building blocks of Android applications. Students will develop a thorough understanding of the Activity lifecycle and learn to navigate between screens using explicit and implicit Intents. Intent filters and data passing between activities are covered in detail. The module also introduces Fragments, their lifecycle, management, and use in building flexible multi-screen applications, giving students the skills needed to construct well-structured, navigable apps

Module 4 – Data Storage

This module equips students with the techniques needed to persist data within Android applications. Topics include SharedPreferences for storing small key-value data, reading and writing to internal device storage, and creating and managing a SQLite relational database through direct SQL operations and the higher-level Room Persistence Library. Students will implement basic Create, Read, Update, and Delete (CRUD) operations and understand when to apply each storage mechanism based on the nature and volume of the data.

Module 5 – Android Services & Notifications

This module introduces the mechanisms Android provides for running operations in the background and communicating with the user. Students will learn about the Service lifecycle, implement Broadcast Receivers to respond to system and app-level events, and use NotificationManager to deliver alerts and updates. The module also covers scheduling tasks with AlarmManager, sending emails via Intent, and managing user alerts, skills essential for building responsive and professionally polished applications.

Module 6 – Networking & Multimedia

This module introduces students to connecting Android applications to the internet and handling rich media content. Topics include making HTTP requests to web servers, parsing JSON responses, and using the Retrofit library for efficient API consumption. Students will also explore Android's Media APIs to integrate audio and video playback into their apps. The module concludes with an overview of the AndroidX libraries and an introduction to RESTful web service integration, preparing students for building data-driven, connected applications.

This course introduces Mechanical and Electronics Engineering students to the fundamentals of Android application development through a structured and hands-on approach. It covers the Android ecosystem, user interface design, application components, data storage techniques, background services, notifications, networking, and multimedia integration. Students gain practical experience in designing, developing, testing, and deploying Android applications using modern development tools and industry-relevant practices, establishing a strong foundation for mobile application development in engineering domains.

Course Outcomes	**After successful completion of the course, the students will be able to.**	
	CO1	Understand the Android ecosystem, architecture, and set up a development environment to build and run basic Android applications
	CO2	Design and implement user interfaces using XML layouts and standard UI components including TextView, EditText, Button, and ImageView.
	CO3	Apply Activity lifecycle concepts, use explicit and implicit Intents, and build multi-screen applications using Fragments
	CO4	Implement data storage mechanisms using SharedPreferences, internal file storage, and SQLite database for data persistence.
	CO5	Understand and apply Android services, broadcast receivers, notification services, and alarm management in applications
	CO6	Integrate networking features to consume web services, parse JSON data, and use multimedia APIs for audio/video playback

Syllabus:

Module No.	Unit No.	Topics	Hours
1	**Introduction to Android**		06
	After completing this module, students will be able to, 1. Understand Android ecosystem and evolution 2. Explain Android architecture and components 3. Set up Android development environment 4. Apply SOLID principles in development		
	1.1	**Android Ecosystem Overview:** Devices, OS layers, Play Store ecosystem, Open-source nature	
	1.2	**History & Versions:** Evolution timeline, major releases, feature improvements, API levels	
	1.3	**Development Setup:** Android Studio installation, SDK configuration, Emulator setup, Running Hello World	
	1.4	**Project Structure & Architecture:** Manifest file, Project folders, AAC components, Activity lifecycle basics	

	Self-Learning Topics: Explore latest Android version features; Compare Android with other mobile OS; Install Android Studio and document steps		
2	**Module 2: Android UI Fundamental**		08
	After completing this module, students will be able to: 1. Design UI using XML layouts 2. Use common UI components 3. Handle screen orientation 4. Implement user interaction feedback		
	2.1	**Project Structure & Architecture:** Manifest file, Project folders, AAC components, Activity lifecycle basics	
	2.2	**UI Components:** TextView, EditText, Button, ImageView properties and usage	
	2.3	**Layout Types:** LinearLayout, RelativeLayout, ConstraintLayout, Use cases	
	2.4	**UI Interaction:** Orientation changes, Action bar, Toast messages, Basic notifications	
	Self-Learning Topics: Design sample mobile app screens; Create responsive layouts		
3	**Module 3: Activities, Intents & Fragments**		**06**
	After completing this module, students will be able to: **1.** Understand activity lifecycle **2.** Use intents for communication **3.** Implement fragments **4.** Design multi-screen apps		
	3.1	**Activity Lifecycle:** States, transitions, callbacks, lifecycle management	
	3.2	**Intents:** Explicit, implicit, intent filters, data passing.	
	3.3	**Application Flow:** Multi-screen navigation, back stack, UI transitions	
	3.4	**Fragment Communication:** Passing data between fragments, fragment-to-activity interaction, ViewModel usage basics	
	Self-Learning Topics: Implement fragment-based UI		

4	**Module 4: Data Storage**		06
	After completing this module, students will be able to: **1.** Store and retrieve data **2.** Use lightweight storage methods **3.** Perform database operations **4.** Implement persistence techniques		
	4.1	**SharedPreferences:** Key-value storage, reading/writing data, use cases	
	4.2	**Internal Storage:** File handling, read/write operations, security aspects	
	4.3	**SQLite Database:** Database creation, CRUD operations, queries	
	4.4	**Room Library:** ORM concepts, advantages, entity and DAO basics	
	Self-Learning Topics: Develop small data storage app; Compare SQLite and Room		
5	**Android Services & Notifications**		06
	After completing this module, students will be able to: 1. Understand background processing 1. Use services and receivers 2. Implement notifications 3. Handle user alerts		
	5.1	**Android Services:** Types, lifecycle, background execution, use cases	
	5.2	**Broadcast Receivers:** Event handling, system broadcasts, custom broadcasts.	
	5.3	**Notifications:** Notification manager, alerts, alarm services, user interactions	
	5.4	**Intent-based Communication:** Sending emails, implicit intents, external apps interaction	
	Self-Learning Topics: Explore background task handling		
	Module 6: Networking & Multimedia Basics		

6	**After completing this module, students will be able to:** 1. Consume web services 2. Handle JSON data **3.** Use networking libraries **4.** Work with multimedia APIs		04
	6.1	**Networking Basics:** HTTP methods, API calls, connectivity handling	
	6.2	**JSON Processing:** Parsing techniques, data mapping, error handling	
	6.3	**Retrofit Library:** Setup, API integration, advantages	
	6.4	**Multimedia:** Audio/video playback, media APIs, AndroidX libraries overview	
	Self-Learning Topics: Create API-based app; Integrate media playback		
		Total	**45**

Suggested List of Activities:

Expt. No.	List of suggested Experiments
1	To set up Android Studio and create a basic Android application. **Objective:** Install Android Studio, configure an emulator, create and run a Hello World application, understand the project structure and Manifest file, and view log statements using Logcat. **Outcome:** Students will be able to set up the Android development environment and execute a basic application successfully.
2	To develop a basic interactive UI application using layouts and button events. **Objective:** Design user interfaces using LinearLayout, RelativeLayout, and ConstraintLayout, display Toast messages, and handle button click events.

	Outcome: Students will be able to create interactive UI applications with event handling capabilities.
3	To design a user input form and transfer data between activities. **Objective:** Create forms with multiple input fields, collect user information, and pass data between activities using Explicit Intents. **Outcome:** Students will be able to design forms and implement activity-to-activity communication.
4	To implement implicit intents and intent filters in Android applications. **Objective:** Open web pages, launch maps using location intents, and configure intent filters for activity access .**Outcome:** Students will be able to interact with system-level applications using implicit intents.
5	To implement various input controls and picker dialogs in Android. **Objective:** Integrate Spinner, AlertDialog, DatePicker, and TimePicker components to capture user input effectively. **Outcome:** Students will be able to use advanced input controls and dialogs in Android applications.
6	To implement menus, selection controls, and navigation features. **Objective:** Create options menus, handle menu selections, use radio buttons, and implement Up navigation in the App Bar. **Outcome:** Students will be able to design structured navigation and selection interfaces.
7	To apply themes, styles, and custom drawable resources in Android applications. **Objective:** Create reusable themes and styles and incorporate drawable resources to improve user interface design. **Outcome:** Students will be able to enhance UI consistency and visual appearance using themes and styles.
8	To implement user preferences using SharedPreferences and a settings screen. **Objective:** Create a settings interface using PreferenceFragmentCompat, store and retrieve preferences, and apply them dynamically.

	Outcome: Students will be able to manage persistent user settings in Android applications.
9	To develop a CRUD-based Notes application using an SQLite database. **Objective:** Perform insert, update, delete, and retrieval operations and display stored data using RecyclerView. **Outcome:** Students will be able to develop database-driven applications supporting CRUD operations.
10	To perform background processing using AsyncTask or WorkManager. **Objective:** Create background tasks, update the user interface upon task completion, and understand threading concepts. **Outcome:** Students will be able to execute background operations efficiently in Android applications.

Assessment Methodology:

Type of Course	**Assessment Tool**	**Marks Distribution**
Theory	CA-20	Certification: NPTEL (20 Marks) (Approved by instructor) ***OR*** Any two Pedagogies (10 marks each) ● MCQ /Class Test ● Case study/Assignment ● GATE based Assignment ● Certification: Udemy/Coursera (Approved by instructor) ● Open Book Test ● Working model / simulation of a course-based concept.
Theory	MSE-30	Question Paper Pattern is as follows: All Questions are compulsory. 1. Q1 A or B - 10 marks 2. Q2 A or B - 10 marks 3. Q3 A or B - 10 marks 4. For each question, A and B should be based on the same CO. 5. MSE should be based on 50% syllabus. 6. Time: 90 minutes (1 hour 30 minutes)

Theory	ESE-50	Question Paper Pattern is as follows: All Questions are compulsory. ● Q1 A or B - 10 marks ● Q2 A or B - 10 marks ● Q3 A or B - 10 marks ● Q4 A or B - 10 marks ● Q5 A or B - 10 marks ● For each question, A and B should be based on the same CO. ● ESE should be based on 30% syllabus of MSE and 70% syllabus after MSE. ● Time: 120 minutes (2 hours) ● Total Marks: 50
Course - Laboratory	TW- 25	1. Active Participation (Lab) = 5 marks 2. Laboratory Report = 10 marks 3. Laboratory performance = 10 marks Based on the performance and satisfactory completion of assigned laboratory work

Text Books:

1. Wei Meng Lee – Beginning Android 4 Application Development, Wrox Publication.

Reference Books:

1. Joseph Annuzzi, Lauren Darcey, Shane Conder – Advanced Android Application Development, 4th Edition, Pearson.
2. John Horton – Android Programming for Beginners, 3rd Edition, Packt Publication.
3. Pradeep Kothari – Android Application Development Black Book, DreamTech

Course Code	Course Name	Teaching Scheme (Hrs. / Week)			Credits Assigned			
25IT5MDM01	Database Management System (*offered to other branches except IT and Cluster branches)	L	P	T	L	P	T	Total
		3	2	-	3	1	-	4
		Examination Scheme						
			CA	MSE	ESE	TW	OR	PR	Total
		Theory	20	30	50	--	--	--	100
		Lab	–	–	–	25	–	–	25
		Total	125						

Pre-Requisites	25FE1VESC02 - Problem Solving using C programming
	25FE2PCC04 - Data structure and algorithm

Course Overview

The Database Management System (DBMS) course provides a comprehensive understanding of modern data management principles, database design methodologies, and relational database technologies. Students learn to model real-world applications using ER/EER diagrams, design normalized databases, formulate and optimize SQL queries, and understand transaction processing, concurrency control, and recovery mechanisms. The course also introduces database programming using JDBC and exposes learners to emerging schema-less NoSQL databases, enabling them to build efficient, scalable, and reliable data-driven applications

Module 1: Introduction to Database Management Systems

This module introduces the fundamental concepts of database systems, highlighting the limitations of traditional file processing systems and the advantages offered by DBMS. Students gain an understanding of database architectures, different types of databases, and the roles of various database users, forming the foundation for advanced database concepts.

Module 2: Data Modeling, Database Design and Normalization

This module focuses on conceptual and logical database design techniques. Students learn to develop ER/EER models, transform them into relational schemas, and apply normalization techniques to eliminate redundancy and maintain data consistency. Real-world case studies help learners understand practical database design considerations.

Module 3: SQL and Relational Algebra

This module develops proficiency in SQL for creating, manipulating, and querying databases. It covers

SQL commands, aggregate operations, joins, views, stored procedures, triggers, and introduces relational algebra operations that provide the theoretical basis for relational query processing

Module 4: Transaction Processing, Concurrency and Database Recovery

This module explains the mechanisms that ensure reliable and consistent database operations in multi-user environments. Students study transaction properties, scheduling techniques, concurrency control protocols, deadlock handling, and recovery methods that preserve data integrity during failures.

Module 5: Database Programming

This module introduces database connectivity and application-level database interactions. Students learn to establish database connections, execute queries, manage records, and implement CRUD operations using JDBC and Java, enabling the development of database-driven applications.

Module 6: Introduction to Schema-less Databases

This module provides an overview of NoSQL databases and schema-less data models. Students explore the differences between SQL and NoSQL systems, understand their advantages and limitations, and perform basic operations using MongoDB, preparing them for modern large-scale data management environments.

The Database Management System (DBMS) course provides students with a strong foundation in database concepts, design methodologies, and relational database technologies. It enables learners to model, design, query, and manage databases efficiently using SQL, normalization techniques, and transaction management concepts. The course also introduces database programming with JDBC and modern NoSQL databases to support the development of scalable and data-driven applications

Course Outcome	**After successful completion, students will be able to .**	
	CO1	Recall fundamental concepts and architectural details related to database management systems.
	CO2	Explain the fundamental concepts, steps, and techniques involved in database design, query processing and transaction processing.
	CO3	Apply SQL commands to define, manipulate, and retrieve data using various database objects for effective information storage and processing.
	CO4	Analyze real-world system requirements to construct and optimize database design for consistency and efficiency.
	CO5	Evaluate database design and identify appropriate techniques for efficient information management/processing.
	CO6	Design and implement a normalized relational database to perform database transaction operations effectively.

Syllabus :

Module No.	Unit No.	Topics	Hours
1	**Introduction to Database Management System**		4
	After completion of the module the student will be able to : 1. Explain the need for DBMS and its advantages over traditional file systems. 2. Describe database architectures, types, and user roles. 3. Differentiate among various database models and application domains.		
	1.1	Introduction to DBMS, Advantages over file system, Database terminology and characteristics.	
	1.2	Database architecture, 3-level architecture, Database types and Database Users.	
	Self-Learning Topic : Case Study based on Amazon Database		
2	**Data Modeling, Database Design and Normalization**		12
	After completion of the module the student will be able to . 1. Develop ER/EER diagrams for real-world applications. 2. Transform conceptual models into relational schemas. 3. Apply normalization techniques to minimize redundancy and maintain data integrity.		
	2.1	Data modeling and models, E-R Modeling: Entities, Attributes, Relationships, Cardinality, Participation, ER diagrams, Limitations, Relational Modeling, Converting ER model to Relational model.	
	2.2	Extended Entity Relationship (EER), Generalization, Specialization and Association.	
	2.3	Case studies based on ER models (real life examples e.g. Amazon, Spotify, Swiggy, LinkedIn, IRCTC, etc.)	
	2.4	Database Anomalies, Normalization process, Advantages, Functional dependencies (Full, Partial and Transitive), Normal forms: 1NF, 2NF and 3NF.	
	Self-Learning Topic: BCNF and Normal forms based problems		
	SQL and Relational Algebra		
	After completion of the module the student will be able to.		

3	1. Use SQL commands to define, manipulate, and query database objects. 2. Implement advanced SQL features such as views, procedures, functions, and triggers. 3. Apply relational algebra operations to formulate and analyze database queries.		11
	3.1	SQL basics - datatypes, operators, integrity constraints; SQL commands - DDL, DML, DCL, Aggregate functions, Group By, Order By, Subqueries and SQL JOINs.	
	3.2	Advance SQL – Virtual Tables (Views), Stored Procedures, Functions and Triggers	
	3.3	Relational Algebra: Union, Selection, Projection, Cartesian Product, Joins	
	Self-Learning Topic: SET Operations		
4	**Transactions Processing, Concurrency and Database Recovery**		11
	After completion of the module the student will be able to. 1. Explain transaction properties and scheduling concepts in DBMS. 2. Analyze concurrency control techniques and deadlock handling mechanisms. 3. Evaluate database recovery methods to ensure consistency and reliability.		
	4.1	Database transaction, ACID properties, Transaction states diagram, TCL (Commit, Rollback, Savepoints), Schedules, Serializability.	
	4.2	Database recovery mechanism - Lag based, Checkpoints and shadow paging; Deadlock detection and prevention	
	4.3	Concurrency control techniques - Locking, Timestamping and Validation protocol.	
	Self-Learning Topic: DeadLock prevention and detection		
5	**Database programming**		4
	After completion of the module the student will be able to. 1. Establish database connectivity using JDBC. 2. Perform CRUD operations through database applications. 3. Develop simple database-driven applications using Java and MySQL.		
	5.1	Database driver and connection string, Cursor, RecordSets, Default query and Parameterized query.	

	5.2	Database programming with JDBC – CRUD Operations with Java and MySQL	
	Self-learning Topics: Domain specific database and UI design		
6	**Introduction to Schema-less Database**		**3**
	After completion of the module the student will be able to : 1. Compare schema-based and schema-less database systems. 2. Explain the features, benefits, and limitations of NoSQL databases. 3. Perform basic data operations using MongoDB and understand its application scenarios.		
	6.1	Schema vs. Schema-less databases. Advantages and disadvantages	
	6.2	Schema-less database (NoSQL) features, SQL vs NoSQL, and MongoDB Basic Operations	
	Self-Learning Topic: Installation and query processing with Cassendra		
		Total	**45**

Suggested List of Experiments:

Expt No.	List of Experiments
1	**Draw an ER diagram and convert it into relational schema for a real life problem** **Objective:** To understand conceptual database design by modeling real-world entities, relationships, and constraints using ER diagrams and converting them into relational schemas. **Outcome:** Students will be able to design ER diagrams for real-world applications and transform them into well-structured relational database schemas.
2	**Perform SQL DDL commands with various Constraints** **Objective:** To learn the use of SQL Data Definition Language (DDL) commands and integrity constraints for creating and managing database objects. **Outcome:** Students will be able to create, modify, and manage database structures while enforcing data integrity using appropriate constraints.
3	**Perform SQL DML commands with Where clause and various Operators**

	Objective: To develop proficiency in manipulating database records using SQL DML statements and filtering data using conditions and operators. **Outcome:** Students will be able to insert, update, delete, and retrieve data efficiently by applying suitable filtering conditions and operators.
4	**Execute Aggregate functions, Group By and Subqueries.** **Objective:** To understand advanced query formulation techniques for data summarization, grouping, and nested query execution. **Outcome:** Students will be able to analyze and summarize database information using aggregate functions, grouping mechanisms, and subqueries.
5	**Perform various types of SQL JOINs** **Objective:** To understand methods for retrieving related data from multiple tables using different SQL join operations. **Outcome:** Students will be able to combine and query data from multiple relations using appropriate join techniques.
6	**Create SQL VIEWs and Indexing** **Objective:** To explore database abstraction and query optimization techniques through views and indexing mechanisms. **Outcome:** Students will be able to create virtual tables and implement indexing strategies to enhance database security and query performance.
7	**Create SQL Procedure and Trigger for the given scenario** **Objective:** To understand procedural extensions of SQL and automate database operations using stored procedures and triggers. **Outcome:** Students will be able to design and implement stored procedures and triggers to support business logic and automated database actions.
8	**Perform CRUD operations using JDBC API** **Objective:** To learn database connectivity and application-level interaction using the JDBC API for

	executing SQL operations. **Outcome:** Students will be able to develop Java-based applications that perform Create, Read, Update, and Delete (CRUD) operations on relational databases.
9	**Simulate transaction control using TCL (Commit, Rollback, Savepoint).** **Objective:** To understand transaction management concepts and control mechanisms that ensure consistency and reliability in database operations. **Outcome:** Students will be able to implement transaction control commands to maintain database integrity during concurrent and multi-step operations.
10	**Perform DCL - New Users, Roles, and Privileges (Revoke and Grant)** **Objective:** To learn database security concepts by managing user access, privileges, and roles using Data Control Language (DCL) commands. **Outcome:** Students will be able to administer database security by creating users, assigning roles, and controlling access permissions effectively.

Assessment Methodology:

Theory	**CA-20**	Certification: NPTEL (20 Marks) (Approved by instructor) OR Any two Pedagogies (10 marks each) 4. MCQ /Class Test 5. Case study/Assignment 6. GATE based Assignment 7. Certification: Udemy/Coursera (Approved by instructor) 8. Open Book Test 9. Working model / simulation of a course-based concept.
Theory	**MSE**	Question Paper Pattern is as follows: All Questions are compulsory. ● Q1 A or B - 10 marks ● Q2 A or B - 10 marks ● Q3 A or B - 10 marks ● For each question, A and B should be based on the same CO. ● MSE should be based on 50% syllabus. ● Time: 90 minutes (1 hour 30 minutes) ● Total Marks: 30

Theory	**ESE**	Question Paper Pattern is as follows: All Questions are compulsory. ● Q1 A or B - 10 marks ● Q2 A or B - 10 marks ● Q3 A or B - 10 marks ● Q4 A or B - 10 marks ● Q5 A or B - 10 marks 7. For each question, A and B should be based on the same CO. 8. ESE should be based on 30% syllabus of MSE and 70% syllabus after MSE. 9. Time: 120 minutes (2 hours) 10. Total Marks: 50
Course - Laboratory	**TW- 25**	● Active Participation (Lab) = 5 marks ● Laboratory Report = 10 marks ● Laboratory performance = 10 marks Based on the performance and satisfactory completion of assigned laboratory work

Textbooks:

1. Database System Concepts – Abraham Silberschatz, Henry Korth, and S. Sudarshan (7th Edition, McGraw Hill)
2. Fundamentals of Database Systems – RamezElmasri and ShamkantNavathe (6th Edition, Pearson)

References:

1. SQL For Dummies – Allen G. Taylor (Wiley).
2. SQL in 10 Minutes, Sams Teach Yourself by Ben Forta.
3. Principles of Distributed Database Systems – M. Tamer Özsu and Patrick Valduriez.
4. NoSQL Distilled – Pramod J. Sadalage, Martin Fowler (Addison-Wesley).

Online Platforms:

1. MIT OpenCourseWare - https://ocw.mit.edu/courses/6-830-database-systems-fall-2010/pages/readings/.
2. NPTEL - https://onlinecourses.nptel.ac.in/noc22_cs91/preview.
3. CoursEra - https://www.coursera.org/learn/database-management.
4. IBM/Edx - https://www.edx.org/certificates/professional-certificate/ibm-sql-nosql-and-relational-database-fundamentals.

Course Code	Course Name	Teaching Scheme (Hrs. / Week)			Credits Assigned			
25ET5MDM01	IoT System Design	L	P	T	L	P	T	Total
		3	2	-	3	1	-	4
		Examination Scheme						
			CA	MSE	ESE	TW	OR	PR	Total
		Theory	20	30	50	-	-	-	100
		Lab	-	-	-	25	-	-	25
		Total	125						

Pre-Requisite Courses:	25FE1BSC01: Engineering Physics
	25FE1VSEC01: C Programming
	25FE1ESC02: Basic Electrical and Digital Electronics
	25ET4MDM01: Embedded Systems

Course Overview

This course gives students a complete guide to building smart, connected systems using the Internet of Things (IoT). It is highly useful for Computer Science (Comp) and Information Technology (IT) students because it teaches them how to collect big data, use cloud platforms like ThingSpeak, and manage network protocols like MQTT. These students will learn to write code that safely connects hardware to web servers and handles data visualization. Mechanical (Mech) students will benefit greatly because modern machinery, factories, and vehicles rely heavily on smart sensors and automatic actuators to run smoothly. They will learn how to read physical values like temperature or motion and turn them into digital data to automate tasks. By bringing these fields together, the course allows Comp and IT students to understand physical machine parts, while Mech students learn how to make their mechanical designs smart and connected. Everyone will gain hands-on skills in prototyping with Arduino and Raspberry Pi, which are highly valued in smart manufacturing and automation jobs. Ultimately, this course helps students from different fields work together to design, build, and secure real-world smart systems.

Module 1 introduces the fundamental concepts of the Internet of Things (IoT), including its architecture, components, and characteristics. It also explains the differences between M2M and IoT systems and introduces Cloud, Fog, and Edge computing paradigms used in IoT applications.

Module 2 focuses on popular IoT development platforms such as Arduino, NodeMCU, and Raspberry Pi. Students learn to interface sensors and actuators with microcontrollers and develop basic programs for data acquisition and device control.

Module 3 covers common wireless communication technologies used in IoT, including Wi-Fi, Bluetooth, ZigBee, and LoRaWAN. It also introduces IoT communication protocols such as MQTT and CoAP and compares them with traditional HTTP communication.

Module 4 explains the role of cloud computing in IoT systems and demonstrates how sensor data can be transmitted, stored, and visualized using cloud platforms such as ThingSpeak. Students also learn to create dashboards and configure simple automated alerts.

Module 5 introduces the security challenges associated with IoT devices and networks. It covers fundamental security practices such as authentication, encryption, password management, and firmware updates to protect IoT systems from common threats.

Module 6 presents the complete workflow of an IoT system, from data collection to cloud-based analytics and decision-making. It explores real-world applications of IoT in smart homes, healthcare monitoring, and intelligent transportation systems through practical case studies.

This course introduces students to the design and development of smart, connected IoT systems by integrating sensors, communication technologies, cloud platforms, and security mechanisms. It provides hands-on experience with IoT hardware and cloud-based data analytics to build real-world applications. The course promotes interdisciplinary learning, enabling students from Computing and Mechanical Engineering backgrounds to collaboratively develop intelligent and automated solutions for modern industries.

	After successful completion of the course, the students will be able to:	
Course Outcomes	CO1	State the basic differences between standard Machine-to-Machine (M2M) communication and cloud, fog, or edge computing.
	CO2	Describe the pin outs and features of development boards like Arduino Uno, NodeMCU, and Raspberry Pi.
	CO3	Construct simple Arduino IDE sketches to read data from sensors and trigger physical actuators like relays.
	CO4	Differentiate between HTTP and lightweight protocols like MQTT to see which is best for small devices.

	CO5	Assess cloud logging data charts on ThingSpeak to judge if a system requires an automatic IFTTT alert.
	CO6	Build a step-by-step IoT case study application that connects physical sensors to a functional web dashboard.

Syllabus:

Module No.	Unit No.	Topics	Hours
1		**Introduction to IoT Basics**	07
		After completing this module, students will be able to. 1. Define what IoT is and identify its basic structural building blocks. 2. Explain the main differences between standard Machine-to-Machine (M2M) communication and IoT. 3. Understand the basic concepts of Cloud, Fog, and Edge computing.	
	1.1	Definition, characteristics, and basic components of an IoT system. Physical and logical design of IoT.	
	1.2	Difference between M2M (Machine to Machine) and IoT. Introduction to Cloud computing, Fog computing, and Edge computing in IoT.	
		Self-Learning Topics: Basic real-world examples of IoT like Smart Homes and Smart Agriculture.	
2		**IoT Hardware and Prototyping (Arduino & Raspberry Pi)**	08
		After completing this module, students will be able to. 1. Differentiate between common IoT development boards like Arduino Uno, NodeMCU, and Raspberry Pi. 2. Connect common sensors (Temperature, Motion, and Gas) and actuators (Relays, Motors) to a microcontroller. 3. Write simple Arduino IDE sketches to read sensor data and control outputs.	
	2.1	Introduction to popular IoT boards: Arduino Uno, NodeMCU (ESP8266/ESP32), and Raspberry Pi pin outs and features.	
	2.2	Interfacing Sensors: DHT11 (Temperature & Humidity), PIR (Motion Sensor), Ultrasonic (Distance), and MQ-2 (Gas/Smoke Sensor).	
	2.3	Interfacing Actuators: Relays (for switching AC home appliances), LEDs, Buzzers, and Servo Motors.	

	2.4	Programming Basics: Writing, compiling, and uploading basic C/C++ code using the Arduino IDE.	
	Self-Learning Topics: Understanding the "Blink" and "Serial Monitor" examples in Arduino IDE.		
3	**Wireless Technologies and IoT Protocols**		08
	After completing this module, students will be able to. 1. Identify common wireless technologies used in IoT and their basic ranges (Wi-Fi, Bluetooth, ZigBee, LoRa). 2. Explain how the MQTT protocol works using the Publish-Subscribe model. 3. Compare HTTP protocol with lightweight IoT protocols like MQTT and CoAP.		
	3.1	Overview of IoT connectivity: Wi-Fi, Bluetooth/BLE, ZigBee, and LoRaWAN (Basic features, range, and use cases).	
	3.2	Introduction to IoT Application Protocols: MQTT (Publish/Subscribe, Broker, Client concept) and CoAP (Request/Response).	
	3.3	Comparison between HTTP and MQTT protocols for small devices.	
	3.4	Overview of IoT connectivity: Wi-Fi, Bluetooth/BLE, ZigBee, and LoRaWAN (Basic features, range, and use cases).	
	Self-Learning Topics: Study how a wireless Wi-Fi router connects IoT devices to the internet.		
4	**Cloud Platforms for IoT**		08
	After completing this module, students will be able to. 1. Send sensor data from an IoT board to a free cloud platform over Wi-Fi. 2. Create online dashboards with graphs to view live data remotely on a phone or laptop. 3. Set up simple automatic alerts (like an email notification when temperature crosses a limit).		
	4.1	Introduction to IoT Cloud platforms: Why do we need the cloud? Overview of free platforms like ThingSpeak.	
	4.2	Data Logging: Sending sensor data to ThingSpeak channels using API keys.	
	4.3	Data Visualization: Creating online charts, gauges, and graphs. Setting up simple triggers (e.g., using IFTTT for email alerts).	

	Self-Learning Topics: Creating a free account on ThingSpeak and exploring its public channels.		
5	**Basic IoT Security**		08
	After completing this module, students will be able to. 1. Identify the common security risks and attack points in a simple IoT setup. 2. Understand the basic importance of passwords, encryption, and secure updates for smart devices.		
	5.1	Introduction to IoT security challenges: Privacy issues, weak passwords, and physical tampering of devices.	
	5.2	Basic security practices: Device authentication, data encryption basics, and the importance of regular firmware updates.	
	Self-Learning Topics: Reading about famous IoT cyberattacks (like the Mirai Botnet) in plain language.		
6	**IoT Applications and Case Studies**		07
	After completing this module, students will be able to. 1. Explain the step-by-step working mechanism of a complete IoT system. 2. Discuss how IoT is transforming industries like healthcare, home automation, and city planning.		
	6.1	Step-by-step design of an IoT project: From sensor data collection to cloud storage.	
	6.2	Real-world Applications: Smart Home Automation (Lights/Fans control).	
	6.3	Smart Health Monitoring (Pulse/Heart rate tracking), and Smart Traffic Management.	
	Self-Learning Topics: Finding a simple DIY IoT project tutorial on Hackster.io and summarizing it. Writing report on various case studies.		
		Total	**45**

Suggested List of Experiments:

Experiment No.	Title of the Experiment
1	**Network Architecture Simulation: Cloud vs. Edge Computing**

	Objective: To understand the differences between Cloud, Fog, and Edge processing models by analyzing data latency and bandwidth usage. **Outcome:** Simulate network topologies to evaluate the structural building blocks and communication delays of an IoT system.
2	**M2M vs. IoT Communication Framework Analysis** **Objective:** To study the distinct architectural differences between point-to-point Machine-to-Machine (M2M) communication and internet-connected IoT structures. **Outcome:** Model and distinguish data flows for isolated M2M setups versus internet-routed cloud IoT networks.
3	**Sensor Interfacing and Serial Monitoring** **Objective:** To interface environmental sensors like the DHT11 (Temperature & Humidity) or MQ-2 (Gas) with an Arduino Uno/NodeMCU board using the Arduino IDE. **Outcome**: Write, compile, and upload sketches to read analog/digital sensor inputs and view real-time data on the Serial Monitor.
4	**Actuator Control and Relay Interfacing** **Objective:** To interface high-power actuators like AC home appliances via Relays, LEDs, and Buzzers with an IoT development board. **Outcome**: Develop an Arduino IDE program to trigger physical actuators based on specific sensory threshold inputs.
5	**Wireless Connectivity Setup and Wi-Fi Configuration** **Objective:** To configure a NodeMCU (ESP8266/ESP32) board to scan and connect to local wireless networks using local Wi-Fi router parameters. **Outcome:** Establish a stable local wireless link and verify the device network configurations and IP assignment.
6	**MQTT Publish-Subscribe Communication Modeling** **Objective:** To implement lightweight application protocol communication using the MQTT Publish-Subscribe model via an external MQTT Broker. **Outcome:** Create an MQTT client to publish sensor payloads and subscribe to control topics, comparing it against heavy HTTP transactions.
7	**Cloud Data Logging on ThingSpeak** **Objective: T**o upload real-time sensor data packets from an internet-connected

	NodeMCU board to a free cloud platform over Wi-Fi using Write API keys. **Outcome:** Configure remote IoT data streaming channels on ThingSpeak to systematically store long-term sensor logs.
8	**Cloud Visualization Dashboards and Web Alerts** **Objective:** To construct online charts, gauges, and visualization widgets on an IoT platform and set up automated email/SMS alerts via IFTTT triggers. **Outcome:** Create a functional web dashboard for remote monitoring and trigger emergency notifications when a sensor value breaches safety limits.
9	**Device Authentication and Firmware Protection** **Objective:** To study the implementation of basic security practices including strong device passwords, authentication keys, and encrypted payloads. **Outcome:** Build a secure connection script that rejects unauthenticated communication requests and protects the node from physical tampering vulnerabilities.
10	**Secure Firmware Updates and Vulnerability Assessment** **Objective:** To understand the importance of regular firmware updates and analyze common security attack points like the Mirai Botnet framework. **Outcome:** Perform a simulated remote secure firmware update (OTA) and identify open vulnerabilities in an unpatched IoT device node.
11	**Local Storage Data Logging** **Objective:** To log sensor data locally on an SPI-based MicroSD card module during network connection or internet dropouts. **Outcome:** Implement a backup storage architecture that prevents data loss when the primary Wi-Fi cloud connection fails.
12	**Bluetooth Low Energy (BLE) Beacon Advertisement** **Objective**: To configure an ESP32 micro-controller as a Bluetooth Low Energy (BLE) beacon to broadcast custom data packets. **Outcome:** Transmit environment parameters over BLE and read them successfully using a mobile phone application.
13	**Ultrasonic Distance Measurement and Buzzer Alert System** **Objective:** To interface an HC-SR04 ultrasonic distance sensor to calculate object proximity and trigger physical audio alerts.

	Outcome: Write an algorithmic loop to map distance values into changing frequency alarm outputs using microcontrollers.
14	**Edge-Based Data Filtering and Threshold Processing** **Objective:** To filter noisy raw sensor data directly at the edge node using a simple moving average filters before transmission. **Outcome:** Reduce network bandwidth usage by transmitting data only when significant parameter variations breach pre-set limits.
15	**CoAP Protocol Request-Response Implementation** **Objective:** To study the lightweight Constrained Application Protocol (CoAP) using a request-response communication model on a local server. **Outcome:** Create a CoAP client node to read resources from an IoT sensor device with low network overhead.

Suggested Course Project Areas

1. Smart Home Automation and Appliance Control System
2. IoT-Based Smart Agriculture and Soil Moisture Monitor
3. Remote Patient Health Telemetry Station (Pulse & Temperature)
4. Smart Weather Station with Cloud Logging (ThingSpeak)
5. IoT Smoke & Gas Leakage Detector with IFTTT Email Alerts
6. Smart Waste Management and Garbage Bin Level Tracker
7. Intruder Detection System with PIR Sensor and MQTT Alerts

Assessment Methodology:

Type of Assessment	Assessment Tools
Continuous Assessment (CA) (20 Marks)	Certification: NPTEL (20 Marks) (Approved by instructor) **OR** Any 02 Pedagogies (10 marks each) ● MCQ /Class Test ● Case study/Assignment ● GATE based Assignment ● Certification Udemy/Coursera (Approved by instructor) ● Open Book Test ● Working model / Simulation of a course-based concept.

Mid Semester Examination (MSE) (30 Marks)	Question Paper Pattern is as follows: All Questions are compulsory. 11. Q1 A or B - 10 marks 12. Q2 A or B - 10 marks 13. Q3 A or B - 10 marks 14. For each question, A and B should be based on the same CO. 15. MSE should be based on 50% syllabus. 16. Time: 90 minutes (1 hour 30 minutes)
End Semester Examination (ESE) (50 Marks)	Question Paper Pattern is as follows: All Questions are compulsory. ● Q1 A or B - 10 marks ● Q2 A or B - 10 marks ● Q3 A or B - 10 marks ● Q4 A or B - 10 marks ● Q5 A or B - 10 marks ● For each question, A and B should be based on the same CO. ● ESE should be based on 30% syllabus of MSE and 70% syllabus after MSE. ● Time: 120 minutes (02 hours) Total Marks: 50
Term Work (25 Marks)	10. Active Participation (Lab) = 05 marks 11. Laboratory Report / Journal = 10 marks 12. Laboratory Performance = 10 marks Based on the performance & satisfactory completion of minimum 8 experiments.

Textbooks

1. Internet of Things (IoT), S.K. Kataria & Sons, Dr. Rajiv Chopra, 1st Edition (2022).
2. The Internet of Things: Enabling Technologies, Platforms, and Use Cases, CRC Press, Pethuru Raj and Anupama C. Raman, 1st Edition (2022).
3. Internet of Things, Oxford University Press India, Dr. Surya Durbha and Dr. Jyoti Joglekar, 1st Edition (2019).
4. Internet of Things (IoT): Principles, Paradigms and Applications of IoT, BPB Publications, Dr. Kamlesh Lakhwani, Dr. Hemant Kumar Gianey, Joseph Kofi Wireko, and Kamal Kant Hiran, 1st Edition (2020).

Reference Books

- Internet of Things A to Z: Technologies and Applications, Wiley-IEEE Press, Qusay F. Hassan, 2nd Edition (2026).
- The Internet of Things, MIT Press, Samuel Greengard, Revised and Updated Edition (2021).
- Wireless Networks and Industrial IoT: Applications, Challenges and Enablers, Springer International Publishing, Nurul Huda Mahmood, Nikolaj Marchenko, Mikael Gidlund, and Petar Popovski, 1st Edition (2020).
- Industry 4.0 and Industrial IoT, Amazon Digital Services, Dr. Ashutosh Kumar Dubey and Prof. (Dr.) S. B. Choraddi, 1st Edition (2023).

Course Code	Course Name	Teaching Scheme (Hrs. / Week)			Credits Assigned			
25ET5MDM02	Data Acquisition and Industrial Monitoring	L	P	T	L	P	T	Total
		3	2	-	3	1	-	4
		Examination Scheme						
			CA	MSE	ESE	TW	OR	PR	Total
		Theory	20	30	50	-	-	-	100
		Lab	-	-	-	25	-	25	50
		Total	125						

Pre-Requisite Courses:	
	25FE1BSC01: Engineering Physics
	25FE1VSEC01: C Programming
	25FE1ESC02: Basic Electrical and Digital Electronics
	25ET4MDM02: Sensor Technology

Course Overview

This course builds upon Sensor Technology and focuses on acquisition, conditioning, processing, visualization and interfacing of sensor data in modern engineering systems. The course is designed for Mechanical, Computer and Information Technology students and emphasizes interdisciplinary applications in Industry 4.0, robotics, automation, smart manufacturing, predictive maintenance and intelligent monitoring systems.

Module 1: This module introduces the fundamentals of measurement systems, instrumentation, and sensor technologies used in engineering applications. It covers the operating principles of conventional and smart sensors, along with their static and dynamic performance characteristics. Students will learn calibration techniques, error analysis methods, and the role of sensors in mechanical, IT, and computer-based systems. The module also provides an introduction to virtual instrumentation for modern measurement and monitoring applications.

Module 2: This module focuses on the acquisition and conditioning of analog and digital signals for accurate measurement and processing. It covers instrumentation amplifiers, Wheatstone bridge circuits, filtering techniques, and methods for reducing noise and interference. Students will understand how signal conditioning improves data quality and reliability in engineering systems. The module also introduces programmable gain amplifiers used in modern instrumentation applications.

Module 3: This module introduces the principles and architecture of data acquisition systems used for collecting and converting real-world signals into digital data. Topics include sampling theory,

sample-and-hold circuits, ADCs, DACs, and data logging systems. Students will learn how data is acquired, processed, and stored in modern measurement systems. An overview of industrial SCADA systems is also provided for self-learning.

Module 4: This module explores the integration of sensors and data acquisition systems with embedded platforms. It covers microcontroller-based acquisition systems and widely used communication protocols such as UART, I²C, SPI, and CAN bus. Students will gain knowledge of reliable data exchange between embedded devices and peripheral systems. The module also introduces the Modbus protocol used in industrial automation.

Module 5: This module focuses on processing, analyzing, and visualizing sensor-generated data for meaningful interpretation. It includes data preprocessing, statistical analysis, filtering methods, and feature extraction techniques. Students will learn how to transform raw sensor data into useful information and present it effectively using Python-based visualization tools. The module also provides a foundation for applying machine learning techniques to sensor data.

Module 6: This module introduces intelligent engineering systems and emerging Industry 4.0 technologies. It covers Cyber-Physical Systems, Digital Twin concepts, predictive maintenance, machine condition monitoring, and smart manufacturing practices. Students will understand how advanced sensing, data analytics, and automation contribute to intelligent industrial operations. The module also explores future trends and emerging intelligent engineering systems through self-learning.

This course focuses on the acquisition, conditioning, processing, visualization, and interfacing of sensor data for modern engineering applications. It covers measurement systems, embedded data acquisition, communication protocols, signal analysis, and Python-based data visualization techniques. The course also introduces Industry 4.0 concepts such as Cyber-Physical Systems, Digital Twins, predictive maintenance, and intelligent monitoring for smart manufacturing environments.

	After successful completion of the course, the students will be able to	
Course Outcomes	CO1	Define measurement systems, sensors and signal acquisition principles.
	CO2	CO2 Explain signal conditioning circuits, data acquisition architectures and conversion techniques.
	CO3	CO3 Implement sensor interfacing and communication protocols using embedded platforms.
	CO4	CO4 Analyze acquired sensor data using processing and visualization techniques.
	CO5	CO5 Evaluate intelligent monitoring solutions for engineering applications.
	CO6	CO6 Design integrated sensor-based intelligent systems for real-world applications.

Syllabus:

Module No.	Unit No.	Topics	Hours
1		**Sensors and Measurement Systems**	05
		After completing this module, students will be able to. 1. Explain the fundamentals of measurement systems, instrumentation, and sensor technologies. 2. Differentiate between conventional sensors and smart sensors based on their characteristics and applications. 3. Analyze the static and dynamic performance characteristics of sensors and measurement systems. 4. Perform calibration and error analysis for measurement systems and identify their applications in mechanical, IT, and computer-based engineering systems.	
	1.1	Measurement systems and instrumentation	
	1.2	Sensor review and smart sensors	
	1.3	Static and dynamic characteristics	
	1.4	Calibration and error analysis	
	1.5	Applications in mechanical, IT and computer syst	
		Self-Learning Topics: Virtual instrumentation.	
2		**Signal Acquisition and Conditioning**	05
		After completing this module, students will be able to. 1. Differentiate between analog and digital signals used in measurement systems. 2. Analyze the operation of instrumentation amplifiers and Wheatstone bridge circuits. 3. Design and apply active and passive filters for signal conditioning. 4. Implement noise reduction and shielding techniques to improve signal quality.	
	2.1	Analog and digital signals	
	2.2	Instrumentation amplifiers	
	2.3	Wheatstone bridge circuits	

	2.4	Active and passive filters	
	2.5	Noise reduction and shielding	
	Self-Learning Topics: Programmable gain amplifiers		
3	**Sensor Data Acquisition System**		05
	After completing this module, students will be able to. 1. Explain the principles of signal sampling and data acquisition. 2. Compare different types of ADCs and DACs based on their characteristics and applications. 3. Analyze the role of sample-and-hold circuits in data acquisition systems. 4. Evaluate the architecture and operation of data loggers and DAQ systems.		
	3.1	Sampling theorem	
	3.2	Sample and hold circuits	
	3.3	ADC fundamentals and types	
	3.4	DAC fundamentals and applications	
	3.5	Data loggers and DAQ architectures	
	Self-Learning Topics: Industrial SCADA overview		
4	**Embedded Interfacing and Communication**		05
	After completing this module, students will be able to. 1. Develop basic microcontroller-based data acquisition and interfacing systems. 2. Explain the working principles of UART, I²C, and SPI communication protocols. 3. Analyze CAN bus communication for industrial and automotive applications. 4. Select appropriate communication protocols for embedded system applications.		
	4.1	Microcontroller-based acquisition systems	
	4.2	UART communication	
	4.3	I2C communication	
	4.4	SPI communication	

	4.5	CAN bus fundamental	
	Self-Learning Topics: Modbus protocol		
5	**Sensor Data Processing and Visualization**		05
	After completing this module, students will be able to: 1.Apply preprocessing techniques to improve the quality of sensor data. 2.Perform statistical analysis and filtering of acquired data. 3.Extract meaningful features from sensor signals for further analysis. 4.Create effective data visualizations using Python-based tools.		
	5.1	Data preprocessing	
	5.2	Statistical analysis	
	5.3	5.3 Filtering techniques	
	5.4	Feature extraction	
	5.5	Data visualization using Python tool	
	Self-Learning Topics: Introduction to machine learning for sensor data		
6	**Intelligent Systems and Industry 4.0 Applications**		05
	After completing this module, students will be able to: 1. Explain the concepts and components of Cyber-Physical Systems and Digital Twins. 2. Analyze predictive maintenance strategies using sensor data. 3. Evaluate machine condition monitoring techniques in industrial environments. 4. Describe the role of smart manufacturing technologies in Industry 4.0 ecosystems.		
	6.1	Cyber-Physical Systems	
	6.2	Digital Twin concepts	
	6.3	Predictive maintenance	
	6.4	Machine condition monitoring	
	6.5	Smart manufacturing and Industry 4.0	

	Self-Learning Topics: Emerging intelligent engineering systems	
	Total	**30**

Suggested List of Experiments:

Task No.	Task / Learning Objective / Learning Outcome
1	Sensor Calibration **Objective:** To understand sensor calibration procedures and evaluate measurement accuracy. **Outcome:** Calibrate a sensor and determine measurement errors and accuracy.
2	Temperature Sensor Interfacing **Objective:** To study the interfacing of temperature sensors with a data acquisition system. **Outcome:** Acquire and interpret temperature data using suitable interfacing techniques.
3	**Pressure Sensor Interfacing** **Objective:** To understand pressure measurement and sensor characteristics. **Outcome:** Interface a pressure sensor and acquire pressure data for analysis.
4	**Instrumentation Amplifier Design** **Objective:** To design signal conditioning circuits for low-level sensor signals. **Outcome:** Design and test instrumentation amplifier circuits for sensor applications.
5	**Filter Implementation** **Objective:** To implement filtering techniques for signal conditioning. **Outcome:** Apply analog and digital filters to improve signal quality.
6	**ADC Interfacing** **Objective:** To understand analog-to-digital conversion and data acquisition. **Outcome:** Interface ADC modules and acquire digital sensor data.
7	**DAC Interfacing** **Objective:** To study digital-to-analog conversion techniques. **Outcome:** Generate analog outputs using DAC interfacing methods.
8	**UART Communication** **Objective:** To establish serial communication between embedded devices. **Outcome:** Implement UART communication for reliable data transfer.
9	**I2C Communication** **Objective:** To study multi-device communication using I2C protocol. **Outcome:** Interface sensors and peripherals using I2C communication.
10	**SPI Communication** **Objective:** To understand high-speed serial communication using SPI protocol.

	Outcome: Implement SPI communication between embedded systems.
11	**Data Logger Development** **Objective:** To develop a system for recording and storing sensor data. **Outcome:** Design and implement a basic sensor data logger.
12	**Sensor Data Acquisition using ESP32** **Objective:** To acquire real-time sensor data using ESP32. **Outcome:** Develop an ESP32-based sensor acquisition system.
13	**Sensor Data Visualization using Python** **Objective:** To visualize acquired sensor data using Python tools. **Outcome:** Generate plots and dashboards for sensor data analysis.
14	**Condition Monitoring System** **Objective:** To monitor machine or process conditions using sensor data. **Outcome:** Analyze sensor data for condition monitoring applications.
15	**Mini Project Demonstration** **Objective:** To integrate course concepts into a practical engineering application. **Outcome:** Develop and demonstrate a sensor-based intelligent monitoring system.

Suggested Mini Project Areas:

1. Machine Health Monitoring System
2. Industrial Data Logger
3. Vibration Monitoring System
4. Environmental Monitoring Station
5. Smart Energy Monitoring System
6. Predictive Maintenance Prototype
7. Multi-Sensor Acquisition Platform

Assessment Methodology:

Type of Assessment	Assessment Tools
Continuous Assessment (CA) (20 Marks)	Certification: NPTEL (20 Marks) (Approved by instructor) **OR** Any 02 Pedagogies (10 marks each) 2. MCQ /Class Test 3. Case study/Assignment 4. GATE based Assignment 5. Certification Udemy/Coursera (Approved by instructor) 6. Open Book Test 7. Working model / Simulation of a course-based concept.

Mid Semester Examination (MSE) (30 Marks)	Question Paper Pattern is as follows: All Questions are compulsory. 2. Q1 A or B - 10 marks 3. Q2 A or B - 10 marks 4. Q3 A or B - 10 marks 5. For each question, A and B should be based on the same CO. 6. MSE should be based on 50% syllabus. 7. Time: 90 minutes (1 hour 30 minutes)
End Semester Examination (ESE) (50 Marks)	Question Paper Pattern is as follows: All Questions are compulsory. 17. Q1 A or B - 10 marks 18. Q2 A or B - 10 marks 19. Q3 A or B - 10 marks 20. Q4 A or B - 10 marks 21. Q5 A or B - 10 marks 22. For each question, A and B should be based on the same CO. 23. ESE should be based on 30% syllabus of MSE and 70% syllabus after MSE. 24. Time: 120 minutes (02 hours) Total Marks: 50
Term Work (25 Marks)	● Active Participation (Lab) = 05 marks ● Laboratory Report / Journal = 10 marks ● Laboratory Performance = 10 marks Based on the performance & satisfactory completion of minimum 8 experiments.
Practical & Oral (25 Marks)	Practical examination will be based on the experiments performed by the students during laboratory sessions.

Text Books:

1. Ramon Pallas-Areny, Sensors and Signal Conditioning.
2. D.V.S. Murthy, Transducers and Instrumentation.
3. H.S. Kalsi, Electronic Instrumentation.
4. A.K. Sawhney, Electrical and Electronic Measurements.

Reference Books:

1. John G. Webster, Measurement, Instrumentation and Sensors Handbook.
2. Nathan Ida, Sensors, Actuators and their Interfaces.
3. Robert H. Bishop, Mechatronics Handbook.
4. Behzad Razavi, Fundamentals of Microelectronics.

Course Code	Course Name	Teaching Scheme (Hrs. / Week)			Credits Assigned			
25ME5MDM01	Quality Engineering	L	P	T	L	P	T	Total
		3	2	-	2	1	-	3
		Examination Scheme						
			CA	MSE	ESE	TW	OR	PR	Total
		Theory	20	30	50	-	-	-	100
		Lab/Tut	-	-	-	25	-		25
		Total	125						

Pre-Requisite Course	25ME4MDM01 - Logistics & SCM

Course Overview

Quality Engineering introduces the principles, tools, and techniques used to achieve and sustain quality in products, processes, and services. The course covers Total Quality Management (TQM), Statistical Quality Control (SQC), Reliability Engineering, Quality Function Deployment (QFD), Lean Six Sigma, and Quality Management Systems. It emphasizes continuous improvement, customer satisfaction, process optimization, and reliability in manufacturing, software, IT, and communication systems.

Module 1: Total Quality Management (TQM)

This module introduces the concepts, principles, and philosophies of Total Quality Management. It focuses on customer satisfaction, continuous improvement, quality costs, and the contributions of leading quality gurus in achieving organizational excellence.

Module 2: Statistical Quality Control and Process Capability

This module covers statistical techniques used for monitoring and controlling process quality. It includes control charts, process capability analysis, and applications of statistical quality control in manufacturing, software, and electronic systems.

Module 3: Acceptance Sampling and Reliability Engineering

This module introduces acceptance sampling methods and reliability concepts used to evaluate product quality and system performance. It also covers reliability analysis of engineering systems and their applications across different domains.

Module 4: Quality Improvement Tools and Techniques

This module focuses on the practical tools used for quality improvement and problem-solving. Students learn the seven basic quality tools, root cause analysis, Kaizen, 5S, and Poka-Yoke techniques for continuous process improvement.

Module 5: Design for Quality and Quality Function Deployment

This module emphasizes customer-focused product and service development. It covers Voice of Customer (VOC), Quality Function Deployment (QFD), House of Quality, and design approaches that improve quality during product development.

Module 6: Lean Six Sigma and Quality Management Systems

This module introduces modern quality improvement methodologies such as Lean Manufacturing and Six Sigma. It also covers ISO 9001 Quality Management Systems, quality audits, and best practices for achieving operational excellence and continuous improvement.

This course provides a comprehensive understanding of quality engineering principles, methodologies, and tools used to ensure excellence in products, processes, and services. It covers Total Quality Management, Statistical Quality Control, Reliability Engineering, Quality Function Deployment, Lean Six Sigma, and Quality Management Systems, emphasizing customer satisfaction, continuous improvement, process optimization, and operational excellence across manufacturing, software, IT, and communication domains.

	After successful completion of this course the students will be able to	
Course Outcomes	**CO1**	Recall the fundamental concepts, philosophies, and terminology associated with quality engineering and TQM.
	CO2	Explain quality management principles, statistical quality control techniques, and quality improvement methodologies.
	CO3	Apply quality engineering tools for process monitoring, process capability analysis, reliability assessment, and quality improvement.
	CO4	Analyze process performance, quality problems, customer requirements, and system reliability using appropriate quality techniques.
	CO5	Evaluate manufacturing, software, IT service, and communication system processes using quality engineering principles and improvement methodologies.
	CO6	Develop quality-focused solutions using TQM, QFD, Lean Six Sigma, reliability engineering, and quality management systems.

Syllabus:

Module No.	Unit No.	Topics	Hours
1	**Total Quality Management**		8
		After completion of this course the students will be able to. 1. Explain the fundamental concepts, principles, and philosophies of Total Quality Management and the contributions of quality gurus. 2. Describe the role of customer satisfaction, quality costs, and continuous improvement in achieving organizational excellence.	
	1.1	Evolution and importance of quality, Dimensions of quality in products and services.,Contributions of Deming, Juran, Crosby, Ishikawa, and Taguchi,Principles of Total Quality Management.	
	1.2	Customer focus and customer satisfaction, Employee involvement and teamwork, Continuous improvement philosophy.	
	1.3	Cost of Quality, Applications of TQM in manufacturing, software industries, IT services, and telecom sectors.	
	Self-Learning Topics: Study of Quality Management Practices in a Leading Organization (e.g., Toyota, Tata Motors, Infosys, TCS, Samsung, etc.)		
2	**Statistical Quality Control and Process Capability**		8
		After completion of this course the students will be able to" 1. Explain process variation and the application of control charts for monitoring process performance. (CO2, CO3) 2. Calculate and interpret process capability indices for assessing process quality and performance. (CO3, CO4)	
	2.1	Process variation and quality characteristics, Statistical concepts for quality control.	
	2.2	Control Charts,$\bar{X}$ Chart, R Chart, p Chart, np Chart, c Chart, u Chart Process Capability Analysis, Process Capability Indices (Cp and Cpk).	
	2.3	Applications of SPC in manufacturing, software defect monitoring, network performance monitoring, and electronics manufacturing.	
	Self-Learning Topics: Applications of Statistical Process Control (SPC) in Manufacturing, Software Development, and Network Performance Monitoring		

3	**Acceptance Sampling and Reliability Engineering**		**7**
		After completion of this course the students will be able to. 1. Apply acceptance sampling concepts and OC curves for quality inspection and decision-making. (CO3) 2. Analyze the reliability characteristics of engineering systems using reliability concepts and models. (CO3, CO4)	
	3.1	Fundamentals of acceptance sampling. Single sampling plans. Producer's Risk and Consumer's Risk. Operating Characteristic (OC) Curve.	
	3.2	Reliability concepts and terminology. Failure rate and reliability function. Bath Tub Curve. Reliability of series and parallel systems. Reliability applications in mechanical systems, computer hardware, communication systems, and embedded systems.	
	Self-Learning Topics: Reliability Analysis of Consumer Products, Computer Hardware, and Communication Systems		
4	**Quality Improvement Tools and Techniques**		**8**
		After completion of this course the students will be able to. 1. Utilize basic quality tools to identify and investigate quality-related problems. 2. Analyze process improvement opportunities using root cause analysis, Kaizen, 5S, and Poka-Yoke techniques.	
	4.1	Seven Basic Quality Tools: Check Sheet, Histogram, Pareto Chart Cause-and-Effect Diagram, Scatter Diagram, Control Chart,Flow Chart	
	4.2	Root Cause Analysis, Kaizen, 5S Methodology. Poka-Yoke (Mistake Proofing),Applications in manufacturing, software quality improvement, IT services, and electronics industries.	
	Self-Learning Topics: Case Study on the Implementation of Kaizen, 5S, or Poka-Yoke in Industrial or Service Organizations		
5	**Design for Quality and Quality Function Deployment**		**7**
		After completion of this course the students will be able to. 1. Explain customer requirements and the methodology of Quality Function Deployment (QFD). 2. Analyze customer needs and translate them into engineering specifications using the House of Quality.	

	5.1	Voice of Customer (VOC), Customer requirements and expectations, Quality Function Deployment (QFD),House of Quality.	
	5.2	Design for Manufacturability (DFM), Design for Assembly (DFA), Taguchi Quality Loss Function, Applications in product design, software development, consumer electronics, and communication devices.	
	Self-Learning Topics: Development of a House of Quality (QFD) for a Product, Mobile Application, or Service System		
6	**Lean Six Sigma and Quality Management Systems**		**7**
		After completion of this course the students will be able to. 1. Analyze equipment performance using reliability, availability, maintainability, MTBF, and MTTR concepts. (CO4) 2. Examine modern maintenance management practices such as TPM, CMMS, Industry 4.0, and predictive maintenance. (CO4)	
	6.1	Lean Manufacturing concepts, Waste identification and elimination, Introduction to Six Sigma, DMAIC Methodology. Quality Management Systems (QMS).	
	6.2	ISO 9001 requirements and documentation, Quality Audits. Quality practices in manufacturing, software quality assurance, IT services, and telecom industries.	
	Self-Learning Topics: Case Study on Lean Six Sigma and ISO 9001 Implementation in Manufacturing, IT, or Telecom Industries		
		TOTAL	**45**

Suggested List of Experiments:

Expt. No.	List of Experiments
1.	**Study of Quality Dimensions and Quality Costs Using Industrial Case Studies** **Objective:** To understand the dimensions of quality and the impact of quality costs on organizational performance. **Outcome:** Students will be able to identify quality dimensions and classify quality costs in industrial applications.
2.	**Analysis of Quality Philosophies of Deming, Juran, Crosby, Ishikawa, and Taguchi**

	Objective: To study the contributions of quality gurus and their role in quality management practices. **Outcome:** Students will be able to explain and compare various quality philosophies for organizational excellence.
3.	**Construction and Interpretation of $\bar{X}$ and R Control Charts** **Objective:** To learn the application of variable control charts for monitoring process performance. **Outcome:** Students will be able to construct and interpret $\bar{X}$ and R control charts for process control.
4.	**Machinery Servicing (Greasing, Oiling, Cleaning, Inspection)** **Objective:**To understand routine servicing procedures required for efficient machine operation. **Outcome:**Students will be able to perform basic machine servicing activities and assess their importance in preventive maintenance.
5.	**Construction of Attribute Control Charts (p, np, c and u Charts)** **Objective:** To understand the use of attribute control charts for quality monitoring and analysis. **Outcome:** Students will be able to select and apply appropriate attribute control charts for different quality characteristics.
6.	**Design of Acceptance Sampling Plans and OC Curve Analysis** **Objective:** To study acceptance sampling techniques used in quality inspection and decision-making. **Outcome:** Students will be able to design sampling plans and analyze OC curves for quality acceptance decisions.
7.	**Reliability Analysis of Series and Parallel Systems** **Objective:** To understand reliability concepts and methods for evaluating system reliability. **Outcome:** Students will be able to determine the reliability of simple engineering systems and configurations.
8.	**Application of Pareto Analysis and Cause-and-Effect Diagrams for Problem Solving** **Objective:** To identify and analyze major causes of quality problems using basic quality tools.

	Outcome: Students will be able to perform root cause analysis and prioritize corrective actions.
9.	**Implementation Study of 5S and Kaizen in Manufacturing or Service Systems** **Objective:** To understand the principles of workplace organization and continuous improvement. **Outcome:** Students will be able to apply 5S and Kaizen methodologies for process and workplace improvement.
10.	**Development of a House of Quality (QFD) for a Selected Product or Service** **Objective:** To translate customer requirements into engineering specifications using QFD. **Outcome:** Students will be able to develop a House of Quality for customer-focused product or service design.
11.	**Lean Waste Identification and Six Sigma DMAIC Case Study** **Objective:** To study Lean and Six Sigma methodologies for quality and process improvement. **Outcome:** Students will be able to identify process waste and apply DMAIC methodology to solve quality-related problems.
12.	**Mini Project on Quality Improvement for a Manufacturing Process, Software Application, IT Service, or Communication System** **Objective:** To integrate quality engineering tools and techniques for solving real-world quality challenges. **Outcome:** Students will be able to develop and present a quality improvement solution using appropriate quality engineering methodologies.

Assessment Methodology:

Assessment Tools	Marks Distribution

Continuous Assessment (CA) (20 Marks)	Certification: NPTEL (20 Marks) (Approved by Instructor) **OR** Any two Pedagogies (10 marks each) ● MCQ /Class Test ● Case study/Assignment ● GATE based Tutorial ● MOOCs Certification (Approved by Instructor) ● Open Book Test ● Working model / Simulation of a course-based concept.
Mid Semester Examination (MSE) (30 Marks)	Question Paper Pattern is as follows: All Questions are compulsory. ● Q1 A or B - 10 marks ● Q2 A or B - 10 marks ● Q3 A or B - 10 marks ● For each question, A and B should be based on the same CO. ● MSE should be based on 50% syllabus. ● Time: 90 minutes (1 hour 30 minutes) ● Total Marks: 30
End Semester Examination (ESE) (50 Marks)	Question Paper Pattern is as follows: All Questions are compulsory. ● Q1 A or B -10 marks ● Q2 A or B -10 marks ● Q3 A or B -10 marks ● Q4 A or B -10 marks ● Q5 A or B - 10 marks ● For each question, A and B should be based on the same CO. ● ESE should be based on 30% syllabus of MSE and 70% syllabus after MSE. ● Time: 120 minutes (02 hours) Total Marks: 50
Term Work (25 Marks)	● Laboratory Performance = 10 marks ● Assignments = 10 marks ● Attendance = 5 marks Based on the performance and satisfactory completion of assigned laboratory work.

Text Books:

1. Total Quality Management – N. V. R. Naidu and K. M. Babu, New Age International Publishers, Latest Edition.
2. Statistical Quality Control – M. Mahajan, Dhanpat Rai Publications, Latest Edition.

3. Quality Planning and Analysis – J. M. Juran and F. M. Gryna, McGraw Hill Education, 5th Edition.

Reference Books:

1. Total Quality Management – Dale H. Besterfield, Pearson Education, 4th Edition.
2. Introduction to Statistical Quality Control – Douglas C. Montgomery, Wiley India, 8th Edition.
3. Quality Engineering Using Robust Design – Madhav S. Phadke, Pearson Education, Latest Edition.
4. Managing for Total Quality – N. V. R. Naidu and K. M. Babu, New Age International Publishers, Latest Edition.
5. Reliability Engineering – E. Balagurusamy, Tata McGraw Hill, Latest Edition.
6. The Certified Quality Engineer Handbook – Connie M. Borror, ASQ Quality Press, 5th Edition.

Online Links

1. Total Quality Management by IIT Kharagpur
2. Statistical Quality Control by IIT Roorkee
3. Six Sigma by IIT Kharagpur
4. Reliability Engineering by IIT Kharagpur

Course Code	Course Name	Teaching Scheme (Hrs. / Week)			Credits Assigned			
25IT5PEC01	Data Mining and Business Intelligence	L	P	T	L	P	T	Total
		2	2	-	2	1	-	3
		Examination Scheme						
			CA	MSE	ESE	TW	OR	PR	Total
		Theory	20	30	50	-	-	-	100
		Lab	-	-	-	25	25		50
		Total	150						

Pre-Requisite Courses:	**25IT3PC02 -** Database Management Systems (DBMS)
	25IT4PCC03 - Advanced Database Management Systems (ADBMS)

Course Overview

Data Mining and Business Intelligence (DMBI) is a comprehensive elective course that focuses on extracting meaningful knowledge and actionable insights from large volumes of data to support informed business decision-making. The course covers fundamental concepts of Data Mining, Data Preprocessing, Classification, Clustering, Frequent Pattern Mining, and Business Intelligence. Students learn analytical techniques and tools used for discovering hidden patterns, trends, and relationships in data. The course also introduces modern Business Intelligence architectures and AI-enabled analytics for solving real-world business problems.

Module I – Introduction to Data Mining: Introduces the fundamental concepts of Data Mining, the Knowledge Discovery in Databases (KDD) process, data mining functionalities, and real-world applications. Covers the evolution of data mining, architecture of data mining systems, and challenges in extracting knowledge from large datasets.

Module II – Data Exploration and Data Preprocessing: Focuses on understanding data characteristics, statistical descriptions, similarity and dissimilarity measures, and preprocessing techniques including data cleaning, integration, reduction (attribute subset selection, histograms, clustering, sampling, data cube aggregation), transformation, normalization, binning, and discretization.

Module III – Classification: Covers supervised learning techniques including Decision Tree Induction (Information Gain, Gain Ratio, Gini Index), Bayesian Classification (Naive Bayes), Simple Linear Regression, Model Evaluation metrics (Accuracy, Precision, Recall, F-Measure), Validation Methods (Holdout, Cross-Validation, Bootstrap), and Ensemble Methods (Bagging, Boosting, AdaBoost, Random Forest).

Module IV – Clustering and Outlier Detection: Introduces unsupervised learning techniques: Partitioning Methods (K-Means, K-Medoids), Hierarchical Methods (Agglomerative, Divisive), Density-Based Methods (DBSCAN), along with outlier types, challenges, and detection methods (Supervised, Semi-Supervised, Unsupervised, Proximity-based, Clustering-based).

Module V – Frequent Pattern Mining: Covers association analysis including Market Basket Analysis, frequent itemsets, closed itemsets, and association rules. Topics include the Apriori Algorithm, candidate generation, pattern growth approaches, and advanced patterns: Multilevel and Multidimensional Association Rule Mining.

Module VI – Business Intelligence: Focuses on Business Intelligence concepts, architectures, and Decision Support Systems. Introduces AI-Driven BI including Predictive Analytics, Conversational BI using Generative AI, Automated Data Preparation, and AI-Assisted Decision Making. Covers development of AI-enabled BI systems and BI tools (Power BI, Tableau) with case studies on Fraud Detection and Recommendation Systems.

This course equips students with the knowledge and skills required to analyze large datasets, discover meaningful patterns, and support data-driven decision-making using Data Mining and Business Intelligence techniques. By integrating modern analytical tools and AI-enabled BI applications, learners are prepared to address real-world business challenges and contribute effectively to data-centric organizations.

After successful completion of the course, the students will be able to ..		
Course Outcomes	**CO1**	Describe the concepts, architecture, and characteristics of data mining and business intelligence systems. (Remembering)
	CO2	Explain data exploration, similarity measures, and preprocessing techniques required for data mining applications. (Understanding)
	CO3	Apply statistical analysis, preprocessing methods, and visualization techniques to organize and prepare datasets for mining tasks. (Apply)
	CO4	Analyze datasets using classification, clustering, outlier detection, and association mining techniques. (Analyse)
	CO5	Evaluate the performance of data mining models using appropriate validation and error metrics. (Evaluate)
	CO6	Design Business Intelligence solutions integrating data mining and AI-enabled analytics for enterprise decision support. (Design)

Syllabus:

Module No.	Unit No.	Topics	Hours
	Introduction to Data Mining		

1	**After completion of the module the student will be able to:** 1. Describe the fundamental concepts, functionalities, and applications of Data Mining. 2. Explain the stages involved in the Knowledge Discovery in Databases (KDD) process. 3. Explain the characteristics and architecture of a Data Mining system. 4. Analyze the challenges, issues, and real-world applications of Data Mining in various domains.		3
	1.1	Introduction to Data Mining: What is Data Mining? Evolution of Data Mining, Data Mining vs Traditional Data Analysis	
	1.2	KDD Process: Stages of KDD – Data Cleaning, Integration, Selection, Transformation, Data Mining, Pattern Evaluation, Knowledge Presentation	
	1.3	Data Mining Systems and Applications: Types of Data Mining Systems, Classification of Data Mining Tasks, Applications in various domains	
	Self-Learning Topics / Tools: Major Issues in Data Mining.Tools: Pentaho		
2	**Data Exploration and Data Preprocessing**		5
	After completion of the module the student will be able to: 1. Analyze datasets using statistical measures and similarity/dissimilarity techniques. 2. Identify data quality issues and select appropriate preprocessing methods. 3. Apply data cleaning, integration, transformation, and reduction techniques. 4. Prepare and visualize datasets suitable for data mining applications.		
	2.1	Data Exploration: Types of Attributes, Statistical Description of Data (Mean, Median, Mode, Variance, Skewness), Measuring Similarity and Dissimilarity	
	2.2	Data Preprocessing: Need for Preprocessing, Data Cleaning, Data Integration, Data Reduction (Attribute Subset Selection, Histograms, Clustering, Sampling, Data Cube Aggregation), Data Transformation and Data Discretization: Normalization, Binning, Histogram Analysis	
	Self-Learning Topics / Tools: Data Visualization		
3	**Classification**		6
	After completion of the module the student will be able to: 1. Apply classification techniques for predictive data analysis. 2. Develop classification models using Decision Tree and Naive Bayes algorithms.		

	3. Evaluate classification models using suitable performance measures and validation methods. 4. Explain ensemble learning techniques for improving prediction accuracy.			
	3.1	Classification Fundamentals: Basic concepts and applications of classification in data mining		
	3.2	Classification Methods: (3.2.1) Decision Tree Induction – Attribute Selection Measures: Information Gain, Gain Ratio, Gini Index; Tree Pruning. (3.2.2) Bayesian Classification: Naive Bayes Classifier. (3.3) Prediction: Structure of Regression Models, Simple Linear Regression		
	3.3	Model Evaluation and Ensemble Methods: Accuracy, Error Measures, Precision, Recall, F-Measure; Validation Methods: Holdout, Random Sampling, Cross Validation, Bootstrap; Ensemble Methods: Bagging, Boosting, AdaBoost, Random Forest		
	Self-Learning Topics / Tools: Multiple Linear Regression, Logistic Regression, Support Vector Machines			
4	**Clustering and Outlier Detection**			6
	After completion of the module the student will be able to: 1. Explain the concepts and applications of clustering techniques. 2. Implement partitioning, hierarchical, and density-based clustering algorithms. 3. Analyze clusters to identify hidden patterns and relationships in data. 4. Apply appropriate outlier detection methods for anomaly identification.			
	4.1	Cluster Analysis: Basic Concepts and applications in data mining		
	4.2	Clustering Methods:		
		4.2.1	Partitioning Methods: K-Means, K-Medoids.	
		4.2.2	Hierarchical Methods: Agglomerative, Divisive.	
		4.2.3	Density-Based Methods: DBSCAN	
	4.3	Outlier Detection: What are outliers? Types, Challenges; Outlier Detection Methods: Supervised, Semi-Supervised, Unsupervised, Proximity-based, Clustering-based		
	Self-Learning Topics / Tools: Hierarchical methods (BIRCH, Chameleon), Density-based methods (OPTICS), Grid-based methods (STING)			
	Frequent Pattern Mining			
	After completion of the module the student will be able to: 1. Explain the concepts of association rules and frequent pattern mining.			

5	2. Apply the Apriori algorithm to discover frequent itemsets. 3. Generate and interpret association rules for business decision-making. 4. Analyze advanced pattern mining techniques for multidimensional datasets.		5
	5.1	Basic Concepts: Market Basket Analysis, Frequent Itemset, Closed Itemset, and Association Rules	
	5.2	Frequent Itemset Mining Methods: (5.2.1) The Apriori Algorithm: Finding Frequent Itemsets using Candidate Generation, Generating Association Rules from Frequent Itemsets, Improving Efficiency of Apriori. (5.2.2) A Pattern Growth Approach for Mining Frequent Itemsets	
	5.3	Advanced Pattern Mining: Mining Multilevel Association Rules and Multidimensional Association Rules	
	Self-Learning Topics / Tools: Topics: Mining Frequent Itemsets using vertical data formats, Association Mining to Correlation Analysis, lift, Constraint-Based Association Mining.		
6	**Business Intelligence**		5
	After completion of the module the student will be able to: 1. Explain the concepts of Business Intelligence and Decision Support Systems. 2. Analyze business problems using data-driven decision support approaches. 3. Utilize modern BI tools for visualization, dashboard creation, and intelligent decision-making. 4. Apply AI techniques such as predictive analytics, recommendation systems, and sentiment analysis to business problems. 5. Develop AI-enabled BI solutions using data mining techniques.		
	6.1	Introduction to Business Intelligence: Definition and Importance of BI, Business Intelligence Architecture, Decision Support System (DSS): Definition and Applications	
	6.2	AI-Enabled BI: Introduction to AI in BI, Predictive Analytics using Machine Learning, Conversational BI using Generative AI, and AI-Assisted Decision Making.	
	6.3	Development of AI-Enabled BI Systems and BI Tools: Steps in Developing a BI System, Integration of Data Mining and AI Techniques in BI, Business Applications (Fraud Detection and Recommendation Systems), AI Features in Modern BI Tools	

	Self-Learning Topics / Tools: Fake News and Cyberbullying Detection Tool: Power BI	
	Total	**30**

Suggested List of Experiments:

Expt. No.	List of Experiments
1	**Introduction to Open-Source Data Mining Tool – WEKA** **Objective:** To familiarize students with the WEKA tool and understand its capabilities for preprocessing, classification, clustering, and association rule mining. **Outcome:** Students can launch and navigate WEKA, load ARFF/CSV datasets, run built-in classifiers, and interpret evaluation metrics such as accuracy and confusion matrix.
2	**Data Exploration using Python / R / Java / any tool** **Objective:** To explore and understand datasets by computing statistical summaries (mean, median, variance, skewness), identifying missing values and outliers, and visualizing distributions and correlations. **Outcome:** Students perform Exploratory Data Analysis (EDA), generate descriptive statistics, produce histograms, box plots, and heat maps, and document initial data quality findings.
3	**Data Preprocessing using Python / R / Java / any tool** **Objective:** To apply data preprocessing techniques including cleaning (handling missing values, noise removal), integration, transformation (normalization, binning), and reduction on real-world datasets. **Outcome:** Students are able to preprocess a raw dataset using Python (Pandas/NumPy) or R and document each preprocessing step applied to improve data quality for mining.
4	**Implementation of Classification using WEKA Tool** **Objective:** To perform and evaluate classification algorithms Decision Tree Induction using the WEKA GUI environment and compare results with programming-based implementations. **Outcome:** Students apply WEKA built-in algorithms for all three mining tasks, interpret results from the WEKA Explorer, and produce a comparative analysis report.
5	**Classification Algorithms using Python / R / Java** **Objective:** To implement and evaluate classification algorithms including, Naive Bayes Classifier, and assess performance using accuracy, precision, recall, and F-measure.

	Outcome: Students implement classification algorithms, build models on standard datasets, evaluate results using confusion matrices and performance metrics, and compare algorithm effectiveness.
6	**Implementation of Frequent Pattern Mining Algorithms using Python / R / Java** **Objective:** To implement the Apriori Algorithm to discover frequent itemsets, generate association rules, and evaluate rules using support, confidence, and lift measures. **Outcome:** Students apply the Apriori algorithm on transactional datasets (e.g., market basket data), generate association rules, and interpret results for business decision-making.
7	**Implementation of Clustering Algorithms using Python / R / Java** **Objective:** To implement partitioning (K-Means, K-Medoids), hierarchical (Agglomerative), and density-based (DBSCAN) clustering algorithms and analyze cluster quality. **Outcome:** Students implement clustering algorithms on real datasets, visualize cluster results, and evaluate clustering quality using appropriate metrics such as silhouette score.
8	**Frequent Pattern Mining using WEKA Tool** **Objective:** To perform and evaluate association rule mining algorithm using the WEKA GUI environment and compare results with programming-based implementations. **Outcome:** Students apply WEKA built-in algorithms for all three mining tasks, interpret results from the WEKA Explorer, and produce a comparative analysis report.
9	**Comparative Case Study of Business Intelligence Tools** **Objective:** To analyze and compare at least three BI tools such as Pentaho, Tableau, Power BI, and QlikView based on features, usability, visualization capabilities, and enterprise integration support. **Outcome:** Students prepare a structured comparative report highlighting strengths, limitations, pricing models, and appropriate use cases of different BI tools for organizational decision-making.
10	**Business Intelligence Mini Project – Data Mining and Power BI Dashboard** **Objective:** To design and implement a complete BI solution: define a business problem, select a standard dataset from repositories (Kaggle, UCI, WEKA), implement suitable data mining algorithms using WEKA/RapidMiner/Python, and create interactive dashboards using Microsoft Power BI. **Outcome:** Students implement an end-to-end BI mini project with data preprocessing, data mining algorithm execution, result visualization through Power BI dashboards, and derive actionable business insights with recommendations.
Self- Study: RapidMiner Studio – Exploration and Workflow Design	

Assessment Methodology:

CA-20	Certification: NPTEL (20 Marks) (Approved by instructor) OR Any two Pedagogies (10 marks each): • MCQ / Class Test • Case Study / Assignment • GATE Based Assignment • Certification: Udemy / Coursera / NPTEL (Approved by instructor) • Open Book Test • Working model / AI-assisted prototype demonstrating a course-based concept
MSE	Question Paper Pattern is as follows – All Questions are compulsory: • Q1 A or B – 10 marks • Q2 A or B – 10 marks • Q3 A or B – 10 marks • A and B for each question should be from the same CO. • MSE should be based on 50% of the syllabus. Time: 90 minutes. Total Marks: 30.
ESE	Question Paper Pattern is as follows – All Questions are compulsory: • Q1 A or B – 10 marks • Q2 A or B – 10 marks • Q3 A or B – 10 marks • Q4 A or B – 10 marks • Q5 A or B – 10 marks • A and B for each question should be from the same CO. • ESE covers 30% of MSE syllabus and 70% of syllabus after MSE. Time: 120 minutes. Total Marks: 50.
TW-25	• Active Participation (Lab) = 5 marks • Mini-Project (BI Dashboard) = 5 marks • Laboratory Report / Journal = 5 marks • Laboratory Performance = 10 marks (based on satisfactory completion of assigned laboratory work)
OR-25	Oral examination based on the experiments and term work performed by the students. Examiner may ask questions on any tool used: WEKA, RapidMiner, Python, Power BI, or any data mining algorithm covered in the syllabus.

Textbooks:

1. Han, Kamber, "Data Mining Concepts and Techniques", Morgan Kaufmann 3nd Edition.
2. P. N. Tan, M. Steinbach, Vipin Kumar, “Introduction to Data Mining”, Pearson Education.

3. Business Intelligence: Data Mining and Optimization for Decision Making by Carlo Vercellis, Wiley India
4. Publications.
5. G. Shmueli, N.R. Patel, P.C. Bruce, “Data Mining for Business Intelligence: Concepts, Techniques, and Applications in Microsoft Office Excel with XLMiner”, 2nd Edition, Wiley India.

References:

1. Provost, Foster and Fawcett, Tom, Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking, O'Reilly Media, First Edition, 2013.
2. Michael Berry and Gordon Linoff “Data Mining Techniques”, 2nd Edition Wiley Publications.
3. Michael Berry and Gordon Linoff “Mastering Data Mining- Art & science of CRM”, Wiley Student Edition.
4. Vikram Pudi & Radha Krishna, “Data Mining”, Oxford Higher Education.

Online Platforms and Useful Links:

1. https://onlinecourses.nptel.ac.in/noc21_cs06/preview (NPTEL – Data Mining)
2. https://ml.cms.waikato.ac.nz/weka/courses.html (Data Mining with WEKA – University of Waikato)
3. https://academy.rapidminer.com/ (RapidMiner Academy – Free Courses)
4. https://pentaho.com/getting-started-with-pentaho/ (Pentaho Getting Started)
5. https://www.kaggle.com/learn/overview (Kaggle Learn – Data Science and ML)
6. https://onlinecourses.nptel.ac.in/noc21_cs33/preview (NPTEL – Python for Data Science)
7. https://learn.microsoft.com/en-us/power-bi/ (Microsoft Power BI Documentation)
8. https://archive.ics.uci.edu/ (UCI Machine Learning Repository – Standard Datasets)

Course Code	Course Name	Teaching Scheme (Hrs. / Week)			Credits Assigned			
25IL5OEXX	Open Elective Course #	L	P	T	L	P	T	Total
		2	–	–	2	--	–	2
		Examination Scheme						
			CA	ESE	TW	OR	PR	Total
		Theory	20	30	--	–	–	50
		Total	50					

List Of Open Electives (OE) Courses # (02 Credits)

- Students are required to **enroll in any one of the NPTEL Open Elective courses from the list of approved courses circulated by the Institute**. Enrolment in courses outside this approved list will not be considered for the Open Elective requirement.

Serial No.	Course Code	Name of Open Elective Courses (OE)
01	25IL5OE01	*Energy Economics and Policy
02	25IL5OE02	*Ethics in Engineering Practice
03	25IL5OE03	*Intellectual Property, Rights and Competition Law
04	25IL5OE04	*Principle of Human Resource Management
05	25IL5OE05	*Patent Search for Engineers and Lawyers
06	25IL5OE06	*Developing Soft Skills and Personality
07	25IL5OE07	*Mastering Speaking and Presentation Skills

***** For Open Elective Courses, online NPTEL courses approved by Institute

Assessment Methodology:

The assessment for the Open Elective course shall be carried out as follows:

1. **EContinuous Assessment (CA):**
    - Students shall complete and submit all the weekly N**PTEL assignments** as per the schedule prescribed by NPTEL.
    - The marks obtained in the NPTEL assignments shall be considered for the Continuous Assessment component.

- The assignment marks shall be **scaled to 20 marks** for the purpose of awarding the CA marks.

2. **End Semester Examination (ESE)**
   - **S**tudents shall register for and appear in the **NPTEL certification examination** for the Open Elective course in which they have enrolled.
   - The marks obtained in the NPTEL certification examination shall be treated as the End Semester Examination marks for the course scaled down to 30 marks.
3. **Grade Incentive**
   - **S**tudents who **successfully qualify in the NPTEL certification examination** shall be awarded **one grade higher** than the grade earned based on their overall performance in the course, subject to the Institute's grading regulations.

Total Marks Obtained (%)	**DBIT Grade**	**Incentive Grade for NPTEL Certification**	**Performance**
>=90	O	O	Outstanding
>=80 and <90	A+	O	Excellent
>=70 and <80	A	A+	Very Good
>=60 and <70	B+	A	Good
>=55 and <60	B	B+	Fair
>=50 and <55	C	B	Average
>=40 and <50	D	C	Pass
<40	F	No incentive grade	Fail

4. **Students Those Who are Not Appearing for the NPTEL Examination**

- Students who **do not appear** for the NPTEL certification examination shall be required to appear for the **college-conducted End Semester Examination** for the same Open Elective course.
  - In such cases, the student **shall not be eligible for the one-grade incentive**, and the final grade shall be awarded based on the marks obtained in the college End Semester Examination together with the Continuous Assessment marks.

Course Code	Course Name	Teaching Scheme (Hrs. / Week)			Credits Assigned			
25IT5PEC02	Foundation of Extended Reality (ER)	L	P	T	L	P	T	Total
		2	2	-	2	1	-	3
		Examination Scheme						
			CA	MSE	ESE	TW	OR	PR	Total
		Theory	20	30	50	-	-	-	100
		Lab	-	-	-	25	25		50
		Total	150						

Pre-Requisite Courses:	25FE1VSEC02 Basic Programming Knowledge C / Python Fundamentals
	Any Image Editing Tool or Software

Course Outcomes	After successful completion of the course, the students will be able to ..	
	CO1	Describe the concepts, components, and ecosystem of Extended Reality technologies.
	CO2	Explain human perception, interaction mechanisms, and usability considerations in immersive environments.
	CO3	Apply principles of XR hardware, tracking technologies, and display systems for immersive applications.
	CO4	Analyze 3D content creation workflows and rendering techniques used in XR environments.
	CO5	Evaluate XR applications and user experiences across various industrial domains.
	CO6	Examine emerging trends such as Spatial Computing, AI-enabled XR, and Metaverse ecosystems for future immersive systems.

Course Overview

Foundations of Extended Reality (XR) introduces the basic concepts, technologies, and applications of immersive computing, including Virtual Reality (VR), Augmented Reality (AR), Mixed Reality (MR), and Extended Reality (XR). The course covers human perception, interaction techniques, XR hardware, tracking systems, and 3D content creation fundamentals. Students explore the use of XR in healthcare, education, manufacturing, entertainment, and smart environments, along with emerging topics such as spatial computing, digital twins, artificial intelligence, and metaverse technologies.

Through theoretical concepts and case studies, learners gain the knowledge required to understand, analyze, and evaluate immersive experiences and prepare for advanced XR development and applications.

Module I – Introduction to Extended Reality

Introduces the evolution and taxonomy of immersive technologies including Virtual Reality, Augmented Reality, Mixed Reality, and Extended Reality. Students understand the XR ecosystem, application domains, technological advancements, and emerging industry trends.

Module II – Human Perception and Interaction in XR

Focuses on human visual perception, spatial cognition, ergonomics, presence, immersion, and interaction paradigms. Students study multimodal interfaces and user experience considerations for immersive environments.

Module III – XR Hardware and Tracking Technologies

Examines XR devices, display systems, sensors, tracking mechanisms, input controllers, and computer vision techniques enabling immersive interactions and environment understanding.

Module IV – 3D Graphics and Content Pipeline for XR

Introduces 3D modeling concepts, coordinate systems, rendering techniques, animation principles, asset optimization, and workflows used in immersive content development.

Module V – Applications of XR

Explores real-world applications of XR in healthcare, education, manufacturing, retail, tourism, defense, entertainment, and Industry 5.0 environments through contemporary case studies.

Module VI – Emerging Trends in XR

Examines recent developments including Spatial Computing, Digital Twins, Artificial Intelligence in XR, Cloud XR, Edge Computing, and Metaverse technologies shaping the future of immersive systems.

This course equips students with the knowledge and skills required to understand immersive technologies, analyze XR systems, evaluate interaction techniques, and explore emerging applications of spatial computing and intelligent immersive environments.

Syllabus:

Module No.	Unit No.	Topics	Hours
	Introduction to Extended Reality		
1			5

	After completion of the module the student will be able to: 1. Describe the evolution of immersive technologies. 2. Differentiate VR, AR, MR, and XR. 3. Explain the XR ecosystem and applications. **4.** Identify opportunities and challenges in XR adoption.		
	1.1	**Introduction to XR- Definition, History and Evolution.**	
	1.2	**Taxonomy of Immersive Technologies - VR, AR, MR, XR continuum**	
	1.3	**1.3 XR Ecosystem - Devices, software platforms, development environments, market trends.**	
	Self-Learning Topics: OpenXR Standard, XR Industry Reports.		
2	**Human Perception and Interaction in XR**		5
	After completion of the module the student will be able to. 1. Explain visual and spatial perception concepts. 2. Analyze factors affecting immersion. 3. Describe multimodal interaction methods. 4. Evaluate user experience in XR.		
	2.1	**Human Perception - Depth perception, stereopsis, field of view.**	
	2.2	Presence and Immersion - Presence, embodiment, cybersickness.	
	2.3	Interaction Paradigms - Gesture recognition, eye tracking, voice interaction, haptics.	
	Self-Learning Topics : Accessibility in XR, Eye Tracking SDKs.		
3	**XR Hardware and Tracking Technologies**		5
	After completion of the module the student will be able to. 1. Explain XR display technologies. 2. Compare tracking mechanisms. 3. Analyze sensor integration methods. 4. Describe computer vision applications.		
	3.1	**XR Hardware - Head Mounted Displays, Smart Glasses.**	
	3.2	Sensors - IMU, LiDAR, GPS, RGB-D cameras.	
	3.3	Tracking Systems - Marker-based, markerless, SLAM, Inside-out tracking.	
	Self-Learning Topics : Simultaneous Localization and Mapping, ARCore, ARKit.		
	3D Graphics and Content Pipeline for XR		
4			5

	After completion of the module the student will be able to. 1. Explain 3D graphics concepts. 2. Analyze rendering workflows. 3. Apply asset optimization techniques. 4. Describe animation principles.		
	4.1	**Fundamentals of 3D Graphics - Coordinate systems, transformations.**	
	4.2	Rendering Techniques- Real-time rendering, shaders, lighting.	
	4.3	Content Creation Pipeline- Modeling, texturing, rigging, optimization.	
	Self-Learning Topics: Procedural Content Generation., Blender.		
5	**Applications of XR**		5
	After completion of the module the student will be able to. 1. Explain XR applications in various domains. 2. Analyze industry case studies. 3. Evaluate benefits and challenges. 4. Explore opportunities in Industry 5.0.		
	5.1	Healthcare and Education -	
	5.2	Manufacturing and Smart Factories	
	5.3	Retail, Tourism, Entertainment and Defense	
	Self-Learning Topics : XR for Sustainable Development, Google Earth VR.		
6	**Emerging Trends in XR**		5
	After completion of the module the student will be able to. 1. Describe spatial computing concepts. 2. Explain digital twins and AI integration. 3. Analyze cloud-based XR architectures. 4. Examine future immersive ecosystems		
	6.1	Spatial Computing Fundamentals	
	6.2	Digital Twins and AI-enabled XR	
	6.3	Cloud XR, Edge XR and Metaverse Technologies	
	Self-Learning Topics: WebXR and Decentralized Metaverse Platforms,NVIDIA Omniverse, Microsoft Mesh.		
		Total	**30**

Suggested List of Experiments:

Expt. No.	Experiment Details
1	**Students often find it difficult to distinguish among VR, AR, MR, and XR technologies and their application domains.** Objective: To study the evolution of XR technologies, compare VR, AR, MR, and XR systems, and analyze their applications across various industries. Outcome: Students will be able to differentiate immersive technologies and identify suitable XR solutions for specific application domains.
2	**Designing immersive experiences requires understanding how human perception and cognition influence user engagement.** Objective: To investigate visual perception, depth cues, field of view, spatial cognition, and factors affecting immersion and presence in XR environments. Outcome: Students will be able to analyze perceptual factors influencing immersion and evaluate user experiences in immersive environments.
3	**Selection of appropriate XR hardware and tracking mechanisms is essential for developing accurate and responsive immersive systems.** Objective: To study XR devices, sensors, controllers, and compare marker-based, markerless, SLAM, and inside-out tracking technologies through demonstrations and case studies. Outcome: Students will be able to explain the working principles of XR hardware components and evaluate tracking technologies for different applications.
4	**Creation of immersive experiences requires an understanding of 3D graphics concepts and content development workflows.** Objective: To explore 3D coordinate systems, transformations, rendering techniques, animation concepts, and content optimization using tools such as Blender or equivalent software. Outcome: Students will be able to understand the XR content development pipeline and prepare optimized 3D assets for immersive applications.
5	**Organizations are increasingly adopting XR technologies, but selecting appropriate solutions requires evaluating practical use cases.** Objective: To analyze real-world case studies of XR applications in healthcare, education, manufacturing, retail, entertainment, and smart industries and assess their benefits and limitations. Outcome: Students will be able to evaluate XR implementations and recommend suitable immersive solutions for domain-specific problems.
6	**Emerging technologies such as Spatial Computing, Digital Twins, AI-enabled XR, and Metaverse platforms are reshaping immersive ecosystems and require systematic study.** Objective: To investigate recent advancements in XR, including Spatial Computing, Cloud XR, AI-assisted immersive systems, Digital Twins, and Metaverse technologies through literature review and technology demonstrations. Outcome: Students will be able to examine future trends in immersive computing and assess their potential impact on next-generation XR applications.

Suggested Hardware and Simulation Tools

Expt. No.	Experiment	Suggested Hardware	Simulation/Software Tools
1	Study of VR, AR, MR, and XR Technologies	Smartphone, VR Box/Cardboard Viewer (Optional)	Meta Quest Demo Videos, OpenXR SDK, WebXR Samples, YouTube XR Experiences
2	Human Perception and Immersion Analysis	VR Headset (Optional), Webcam	Unity XR Simulator, Mozilla Hubs, Google VR SDK, Figma for UX Prototyping
3	XR Hardware and Tracking Technologies	Smartphone with Camera and IMU Sensors, Meta Quest (Optional)	ARCore, ARKit, Vuforia Engine, OpenCV, SLAM Demonstrations
4	3D Graphics and Content Pipeline	Standard PC/Laptop	Blender, Sketchfab, Mixamo, Unity Personal Edition, Autodesk Viewer
5	XR Application Case Studies	Standard PC/Laptop	Google Earth VR, Microsoft Mesh, Spatial.io, NVIDIA Omniverse Viewer, Mozilla Hubs
6	Emerging Trends in XR and Metaverse	Standard PC/Laptop	NVIDIA Omniverse, Unity XR Toolkit, Meta Horizon Worlds, Decentraland, Unreal Engine View

Assessment Methodology:

CA-20	Certification: NPTEL (20 Marks) (Approved by instructor) OR Any two Pedagogies (10 marks each): • MCQ / Class Test • Case Study / Assignment • GATE Based Assignment • Certification: Udemy / Coursera / NPTEL (Approved by instructor) • Open Book Test • Working model / AI-assisted prototype demonstrating a course-based concept

MSE	Question Paper Pattern is as follows – All Questions are compulsory: • Q1 A or B – 10 marks • Q2 A or B – 10 marks • Q3 A or B – 10 marks • A and B for each question should be from the same CO. • MSE should be based on 50% of the syllabus. Time: 90 minutes. Total Marks: 30.
ESE	Question Paper Pattern is as follows – All Questions are compulsory: • Q1 A or B – 10 marks • Q2 A or B – 10 marks • Q3 A or B – 10 marks • Q4 A or B – 10 marks • Q5 A or B – 10 marks • A and B for each question should be from the same CO. • ESE covers 30% of MSE syllabus and 70% of syllabus after MSE. Time: 120 minutes. Total Marks: 50.
TW-25	• Active Participation (Lab) = 5 marks • Mini-Project (BI Dashboard) = 5 marks • Laboratory Report / Journal = 5 marks • Laboratory Performance = 10 marks (based on satisfactory completion of assigned laboratory work)
OR-25	Oral examination based on the experiments and term work performed by the students. Examiner may ask questions on any tool used: WEKA, RapidMiner, Python, Power BI, or any data mining algorithm covered in the syllabus.

Textbooks

1. Alan B. Craig, *Understanding Augmented Reality: Concepts and Applications*, Morgan Kaufmann Publishers, 1st Edition, 2013.
2. Dieter Schmalstieg and Tobias Hollerer, *Augmented Reality: Principles and Practice*, Addison-Wesley Professional, 1st Edition, 2016.
3. Steven M. LaValle, *Virtual Reality*, Cambridge University Press, 1st Edition, 2017.
4. Tony Parisi, *Learning Virtual Reality: Developing Immersive Experiences and Applications for Desktop, Web, and Mobile*, O'Reilly Media, 1st Edition, 2015.
5. Jason Jerald, *The VR Book: Human-Centered Design for Virtual Reality*, Association for Computing Machinery (ACM), 1st Edition, 2016.

Reference Books

1. Burdea, G. and Coiffet, P., *Virtual Reality Technology*, Wiley-IEEE Press, 2nd Edition, 2003.
2. Oliver Bimber and Ramesh Raskar, *Spatial Augmented Reality: Merging Real and Virtual Worlds*, CRC Press, 1st Edition, 2005.
3. Charles River Editors, *Introduction to Augmented Reality and Virtual Reality*, Charles River Publishers, 2021.
4. Anind K. Dey, Mark Billinghurst, Robert W. Lindeman, J. Edward Swan II, *User Interface Design and Evaluation for Virtual and Augmented Reality*, Springer, 2018.
5. Ralf Doerner, Wolfgang Broll, Paul Grimm, Bernhard Jung, *Virtual and Augmented Reality (VR/AR): Foundations and Methods of Extended Realities*, Springer, 2022.
6. Cathy Hackl, Dirk Lueth, Tommaso Di Bartolo, *Navigating the Metaverse: A Guide to Limitless Possibilities in a Web 3.0 World*, Wiley, 2022.

Online Platforms and Useful Links

1. NPTEL – Introduction to Virtual Reality
    - https://nptel.ac.in/
2. Unity Learn – XR Development Pathway
    - https://learn.unity.com/
3. OpenXR Specification and Documentation
    - https://www.openxr.org/
4. Google ARCore Developer Documentation
    - https://developers.google.com/ar
5. Apple ARKit Documentation
    - https://developer.apple.com/augmented-reality/
6. Blender Official Tutorials
    - https://www.blender.org/support/tutorials/
7. NVIDIA Omniverse Learning Resources
    - https://developer.nvidia.com/omniverse
8. Mozilla WebXR Documentation
    - https://developer.mozilla.org/en-US/docs/Web/API/WebXR_Device_API
9. Microsoft Mixed Reality Documentation
    - https://learn.microsoft.com/en-us/windows/mixed-reality/
10. Meta Developer Portal for XR
    - https://developers.meta.com/

Recommended NPTEL/SWAYAM Courses

1. *Virtual Reality* – IIT Madras / IIT Delhi (subject to availability).

2. *Computer Graphics* – IIT Delhi.
3. *Human-Computer Interaction* – IIT Guwahati.
4. *Introduction to Computer Vision* – IIT Kharagpur**.**

<table>
<tr><th rowspan="2">Course Code</th><th rowspan="2">Course Name</th><th colspan="3">Teaching Scheme (Hrs. / Week)</th><th colspan="4">Credits Assigned</th></tr>
<tr><th>L</th><th>P</th><th>T</th><th>L</th><th>P</th><th>T</th><th>Total</th></tr>
<tr><td rowspan="8">25IT5PEC03</td><td rowspan="8">Blockchain and Cryptocurrency</td><td>2</td><td>2</td><td>–</td><td>2</td><td>1</td><td>–</td><td>3</td></tr>
<tr><td colspan="7">Examination Schemez</td></tr>
<tr><td colspan="3" rowspan="2">Theory</td><td>CA</td><td>MSE</td><td>ESE</td><td>Total</td></tr>
<tr><td>20</td><td>30</td><td>50</td><td>100</td></tr>
<tr><td colspan="3" rowspan="2">Tutorial</td><td>TW</td><td>OR</td><td>PR</td><td>Total</td></tr>
<tr><td>25</td><td>25</td><td>–</td><td>–</td></tr>
<tr><td colspan="3">Total</td><td colspan="4">150</td></tr>
</table>

Pre-Requisite Courses:	25IT4PCC02 Computer Network Design
	25FE1VSEC02 Problem Solving using C- Programming

Course Overview:

This course provides a comprehensive exploration of blockchain architecture, cryptography, and network operations through the lens of Bitcoin and institutional applications. Students will analyze cryptographic primitives, decentralized consensus models, and the mechanics of peer-to-peer networks. By exploring transaction scripts, wallet structures, and real-world fintech case studies, learners gain the technical and analytical engineering foundations necessary to evaluate, design, and deploy secure distributed ledger technologies within modern digital economies.

Module 1: Introduction to Blockchain This module introduces the structural and cryptographic foundations of distributed ledgers. Students explore block anatomy, identifiers, and block-linking mechanics. It covers cryptographic hash functions and Merkle trees for data integrity, concluding with node hierarchies and Simplified Payment Verification protocols to analyze how lightweight clients verify ledger inclusion.

Module 2: Consensus and Mining

This module examines decentralized coordination and data persistence in trustless networks. Students analyze the Byzantine Generals Problem, independent transaction verification rules, and mining node operational mechanics. It details block construction, Proof-of-Work puzzles, chain selection, validation pipelines, and the governance implications of blockchain network forks.

Module 3: Introduction to Bitcoin

This module traces the historical evolution and practical lifecycle of the pioneer cryptocurrency, Bitcoin. Students examine market pricing mechanisms, wallet interactions, and fundamental sending and receiving protocols. The curriculum details serialized transaction structures, network fee dynamics, user privacy strategies, and the global transactional lifecycle.

Module 4: Concepts of Bitcoin

This module dives deep into the cryptographic data structures and programming logic of Bitcoin. Students study elliptic curve keys, Base58Check encoding, and hierarchical deterministic wallet paths. It extensively covers stack-based, Turing-incomplete Bitcoin Script execution across standard locking mechanisms from legacy P2PKH to modern P2TR Taproot scripts.

Module 5: Bitcoin Networks

This module analyzes the decentralized peer-to-peer architecture driving global network routing. Students investigate diverse node roles, discovery protocols, inventory exchanges, and relay networks. Special emphasis is placed on the network communication optimization of SPV clients, including data privacy enhancement via probabilistic Bloom filters.

Module 6: Applications & Case Studies

This module bridges core engineering principles with macroeconomic and institutional implementations. Students perform critical structural analyses of sovereign cryptocurrency adoption using El Salvador's Lightning Network implementation. It further evaluates corporate permissioned networks through JPMorgan's Onyx architecture, JPM Coin, and atomic cross-border settlement systems.

This course introduces the fundamental concepts of blockchain technology, including cryptography, consensus mechanisms, Bitcoin architecture, and peer-to-peer networks. Students explore transactions, wallets, scripting, mining, and network operations while analyzing real-world blockchain applications and case studies. The course equips learners with the knowledge required to evaluate, design, and deploy secure distributed ledger solutions for modern digital economies.

Course Outcomes	**After successful completion of the course, the students will be able to**	
	CO1	Identify the core architectural components of a standard block header, cryptographic key formats, and peer-to-peer node roles within the Bitcoin network ecosystem.
	CO2	Explain the mechanical execution of decentralized consensus models, the step-by-step resolution of the Byzantine Generals Problem, and the structural purpose of Merkle Tree hierarchies.
	CO3	Compute the validation path for a Simplified Payment Verification (SPV) block using sibling hashes and track stack manipulation during a standard P2PKH transaction script execution.

	CO4	Analyze the algorithmic factors influencing mempool transaction prioritization, fee market dynamics, and the system-wide security implications of blockchain forks.
	CO5	Evaluate the structural and privacy trade-offs between utilizing full nodes versus SPV nodes integrated with probabilistic Bloom filters for network query obfuscation.
	CO6	Design a functional deployment specification for tokenized institutional asset transfers using permissioned atomic settlement smart contracts based on the Onyx clearing architecture.

Syllabus:

Module No.	Unit No.	Topics	Hours
1	**Introduction to Block chain**		5
	After completion of the module the student will be able to . 1. Describe the foundational components of a blockchain, including the block header, genesis block, and block identifiers. 2. Explain the structural mechanics and purpose of cryptographic hash functions and Merkle Trees in maintaining immutable data integrity. 3. Illustrate the node hierarchy and the verification protocol used by Simplified Payment Verification (SPV) clients to check block inclusion.		
	1.1	Structure of a Block, Block Header, Block Identifiers: Block Header Hash and Block Height, The Genesis Block, Linking Blocks in the Block chain	
	1.2	Cryptographics Hash Functions, Structure of a Merkle Tree, Use of Merkle tree in Blockchain.	
	1.3	Node Hierarchy, Structure of (Simplified Payment Verification) SPV Blocks , SPV Verification Protocol	
	Self Study Topics: Blockchain Explorer Tool		
2	**Consensus and Mining**		4
	After completion of the module the student will be able to. 1. Summarize the Byzantine General's Problem and its relevance to achieving trustless, decentralized consensus.		

	2. Detail the step-by-step operational mechanics of how mining nodes aggregate transactions, construct headers, and successfully mine a new block. 3. Analyze the process of chain selection, block validation rules, and the network conditions that result in blockchain forks.		
	2.1	Decentralized Consensus, Byzantine General's Problem, Independent Verification of Transactions	
	2.2	Mining Nodes: Aggregating Transactions into Blocks, Constructing the Block header, Successfully Mining the Block, Validating a New Block, Assembling and Selecting Chains of Blocks, Block chain Forks	
	Self Study Topics: Nothing-at-Stake Problem, Comparison with Byzantine General's Problem		
3	**Introduction to Bit coin**		4
	After completion of the module the student will be able to. 1. Outline the history of Bitcoin and the basic mechanics of acquiring, sending, and receiving cryptocurrency. 2. Map the complete lifecycle of a Bitcoin transaction from initial wallet broadcast to mempool entry and final block confirmation. 3. Analyze how transaction fees are calculated based on data size and the strategies users employ to maintain privacy on a public ledger.		
	3.1	What is Bit coin and the history of Bit coin, Getting the first bit coin, finding the current price of bit coin and sending and receiving bit coin	
	3.2	Bit coin transaction, Transaction Life cycle, transaction Fees, Privacy in Transaction	
	Self Study Topics: Cryptocurrency Markets and Exchanges		
4	**Concepts of Bit coin**		9
	After completion of the module the student will be able to. 1. Differentiate between various wallet architectures, specifically focusing on the generation of Hierarchical Deterministic (HD) wallets and Base58Check encoding. 2. Demonstrate the execution of standard transaction scripts by structurally tracking the lock and unlock validation stack (e.g., P2PKH). 3. Compare the storage and privacy enhancements of advanced script formats, moving from legacy P2SH to modern SegWit (P2WPKH) and Taproot (P2TR).		

	4.1	Keys and Address: Public Key Cryptography and Crypto currency, Private and Public Keys, Bitcoin Addresses, Base58 and Base58Check Encoding	
	4.2	Wallets : Nondeterministic (Random) Wallets, Deterministic (Seeded) Wallets, HD Wallets (BIP-32/BIP-44), Wallet Best Practices, Using a Bitcoin Wallets,	
	4.3	Transaction: Transaction Outputs and Inputs, Transaction Fees, Transaction Scripts and Script Language	
	4.4	Transaction Scripts and Script Language, Turing Incompleteness, Stateless Verification, Script Construction (Lock + Unlock), Pay-to-Public-Key-Hash (P2PKH), Pay-to-Public-Key (P2PK), Pay-to-Script-Hash (P2SH), Pay-to-Witness-Public-Key-Hash (P2WPKH), Pay-to-Witness-Script-Hash (P2WSH), Pay-to-Taproot (P2TR)	
	Self Study Topics: Blockchain Platforms for other cryptocurrency		
5	**Bitcoin Networks**		6
	After completion of the module the student will be able to : 1. Identify the various node types, their specific data storage roles, and the incentive-based engineering governing the peer-to-peer network. 2. Explain the protocols for network discovery, inventory exchange, and how transaction pools are synchronized across full nodes. 3. Evaluate how SPV nodes utilize probabilistic Bloom filters to balance bandwidth efficiency with transactional privacy.		
	5.1	Peer-to-Peer Network Architecture, Node Types and Roles, Incentive based Engineering The Extended Bitcoin Network, Bitcoin Relay Networks	
	5.2	Network Discovery, Full Nodes, Exchanging “Inventory”, Simplified Payment Verification (SPV) Nodes, Bloom Filters	
	5.3	SPV Nodes and Privacy, Encrypted and Authenticated Connections, Transaction Pools	
	Self Study Topics: Blockchain Security and attacks on Blockchain		
6	**Applications & case studies**		4
	After completion of the module the student will be able to : 1. Examine the macroeconomic and infrastructure challenges of deploying cryptocurrency as legal tender, utilizing the El Salvador case study.		

	2. Assess the architectural efficiency and risk mitigation of using a permissioned blockchain network (JPMorgan Onyx) for wholesale cross-border clearing. 3. Propose a conceptual deployment model for atomic settlement and tokenized asset transfers in other enterprise environments based on institutional case studies.		
	6.1	El Salvador's Bitcoin Adoption Case study	
	6.2	JPMorgan's Onyx & JPM Coin case study - cross border clearing.	
	Self Study Topics: Case study of Blockchain in Health sector		
		Total	**30**

Suggested List of Experiments:

Expt.. No	List of Experiments
1	**Cryptographic Hashing and Block Header Construction** **Objective:** To write a Python script that takes a dictionary of transaction data, serializes it using json.dumps(), and applies hashlib.sha256() to generate a hash. Extend this to construct a dictionary representing a block header. **Outcome:** Students will be able to programmatically assemble a block data structure in Python and demonstrate the avalanche effect by altering a single character in the data.
2	**Merkle Tree Construction and Root Calculation** **Objective:** To write a Python script that accepts a list of strings (transactions), hashes each string, and iteratively pairs and concatenates the hashes (hashlib.sha256(left + right)) until a single Merkle Root string remains. **Outcome:** Students will be able to implement data aggregation algorithms using Python lists to compute a cryptographic root.
3	**Proof-of-Work (PoW) Mining Simulation** **Objective:** To create a Python mining loop that appends an integer (nonce) to a string (block data), hashes the combined string, and checks if the .hexdigest() starts with a target number of zeros (e.g., hash.startswith('0000')). **Outcome:** Students will be able to simulate decentralized consensus by measuring the computational execution time required to "mine" a valid hash in Python.
4	**Analyzing Transactions via API (Replaces Web Navigation)** **Objective**: To write a Python script that fetches real-time block and transaction data from a public blockchain API (e.g., requests.get('https://mempool.space/api/blocks/tip/height')) and parses the returned JSON payload. **Outcome:** Students will be able to programmatically extract, parse, and analyze real-time

	on-chain metrics (inputs, outputs, fees) directly within a Python environment.
5	**Asymmetric Key Generation and Base58Check** **Objective**: To generate an SECP256k1 private key using Python's ecdsa library, derive the public key, and apply consecutive hashlib.sha256() and hashlib.new('ripemd160') functions, finally encoding the result with base58.b58encode_check(). **Outcome:** Students will be able to execute the exact cryptographic pipeline required to map a raw private key to a standard cryptocurrency address.
6	**Hierarchical Deterministic (HD) Wallet Derivation** **Objective:** To write a script that generates a 12-word BIP-39 seed phrase using the mnemonic library, and then derives multiple child addresses (Index 0, 1, 2) from that single master seed. **Outcome:** Students will be able to programmatically generate deterministic key trees, demonstrating how modern wallets manage backup and recovery.
7	**Stack-Based Bitcoin Script Execution Simulator** **Objective:** To build a Python script that simulates a P2PKH transaction by parsing a list of string commands (e.g., ['<Sig>', '<PubKey>', 'OP_DUP', 'OP_HASH160']), popping elements, performing basic logic (like equality checks), and pushing results back to the list. **Outcome**: Students will be able to model stateless, Turing-incomplete execution environments using standard Python data structures.
8	**SPV Verification and Merkle Proofs** **Objective**: To write a Python function verify_proof(target_hash, proof_list, root) that takes a transaction hash and a list of sibling hashes, iteratively concatenates and hashes them, and returns True if the final output matches the provided Merkle Root. **Outcome:** Students will be able to validate cryptographic inclusion paths for lightweight node verification.
9	**Implementing Bloom Filters for Network Privacy** **Objective:** To initialize a Python bit array of a fixed size, apply multiple hash functions to a list of transaction IDs to set bits to 1, and then write a function to check if a new transaction ID *might* be in the set or is *definitely not* in the set. **Outcome**: Students will be able to code probabilistic data structures in Python to demonstrate bandwidth and privacy optimizations.
10	**Simulating Atomic Settlement (Pure Python Implementation)** **Objective:** To define two Python classes: BankNode (holding digital cash and tokenized assets) and SmartContractEscrow. Write an execute_swap(bank_a, bank_b) method that strictly requires both parties to possess sufficient balances before updating their state dictionaries simultaneously (Delivery-versus-Payment). **Outcome:** Students will be able to design object-oriented escrow logic that eliminates counterparty settlement risk, modeling institutional blockchain clearing architecture natively in Python.

Assessment Methodology:

CA-20	Certification: NPTEL (20 Marks) (Approved by instructor) OR Any two Pedagogies (10 marks each) * MCQ / Class Test * Case Study / Assignment * GATE Based Assignment * Certification: Udemy / Coursera / NPTEL (Approved by instructor) * Open Book Test * Working model / AI-assisted prototype demonstrating a course-based concept.
MSE	All Questions are compulsory. * Q1 A or B – 10 marks * Q2 A or B – 10 marks * Q3 A or B – 10 marks * A and B for each question should be from the same CO. * MSE should be based on 50% syllabus. Time: 90 minutes. Total Marks: 30.
ESE	All Questions are compulsory. * Q1–Q5 A or B – 10 marks each * A and B for each question should be from the same CO. * ESE based on 30% MSE syllabus + 70% post-MSE syllabus. * Time: 120 minutes. Total Marks: 50.
TW-25	• Active Participation (Lab) = 5 marks • Laboratory Report = 10 marks • Laboratory performance = 10 marks Based on the performance and satisfactory completion of assigned laboratory work
OR-25	Oral examination will be based on the entire syllabus.

Textbooks:

1. "Mastering Bitcoin, PROGRAMMING THE OPEN BLOCKCHAIN", 2nd Edition by Andreas M. Antonopoulos, June 2017, O'Reilly Media, Inc. ISBN: 9781491954386.
2. "Blockchain Applications: A Hands-On Approach", by ArshdeepBahga, Vijay Madisetti, Paperback – 31 January 2017.
3. "Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction", July 19, 2016, by Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, Steven Goldfeder, Princeton University Press.

References:

1. "Mastering Blockchain", by Imran Bashir, Third Edition, Packt Publishing
2. "Mastering Ethereum: Building Smart Contracts and Dapps Paperback" byAndreas Antonopoulos, Gavin Wood, Publisher(s): O'Reilly Media

3. “Blockchain revolution: how the technology behind bitcoin is changing money, business and the world don tapscott and alex tapscot, portfolio penguin, 856157449

Useful Links:

Sr. No	Resource Name	URL Link
1	Bitcoin Developer Guide: P2P Network	https://developer.bitcoin.org/reference/p2p_networking.html
2	BIP 37: Bloom Filtering Specification	https://github.com/bitcoin/bips/blob/master/bip-0037.mediawiki
3	Bitcoin Wiki: Node Typologies	https://en.bitcoin.it/wiki/Node_types
4	JPMorgan Onyx Documentation	https://www.jpmorgan.com/onyx/index
5	NBER Academic Paper: El Salvador	https://www.nber.org/papers/w29968
6	Mempool.space Network Metrics	https://mempool.space/

Course Code	Course Name	Teaching Scheme (Hrs. / Week)			Credits Assigned			
25IT5ELC	Research Methodology	L	P	T	L	P	T	Total
		2	-	-	2	-	-	2
		Examination Scheme						
			CA	MSE	ESE	TW	OR	PR	Total
		Theory	-	-	-	50	-	-	50
		Lab	-	-	-	-	-	-	-
		Total	50						

Pre-Requisite	25FE1BSE01 Engineering Mathematics 01

Course Overview:

This course introduces engineering students to the principles and practices of scientific research. It covers problem identification, literature review, research design, experimental methods, data analysis, technical writing, research ethics, and intellectual property rights. The course equips students with the skills required to undertake research projects, publish technical papers, and pursue innovation in engineering.

Module 1: Introduction to Engineering Research Concepts and Types of Research

Introduces the fundamentals of engineering research, including its purpose, significance, and role in technological innovation. It covers the research process, scientific inquiry, and various types of research such as basic, applied, experimental, and computational research. Students will gain an understanding of how research contributes to solving engineering problems and advancing technology.

Module 2: Literature Review and Research Problem Formulation

Focuses on identifying research problems and conducting effective literature reviews. It introduces techniques for searching, evaluating, and synthesizing scholarly literature to identify research gaps and formulate research objectives, questions, and hypotheses. Students will develop the ability to define meaningful engineering research problems based on existing knowledge and emerging challenges.

Module 3: Research Design and Methodology

Introduces the principles of research design and methodology used in engineering research. It covers the selection of appropriate research approaches, methods, and procedures for conducting systematic investigations. Students will learn to design research studies, develop experimental plans, select data collection methods, and ensure the reliability and validity of research outcomes..

Module 4: Data Collection and Statistical Analysis

Focuses on methods of data collection, organization, and analysis in engineering research. It introduces primary and secondary data sources, sampling techniques, and statistical tools used for data interpretation. Students will learn to apply descriptive and inferential statistics to analyze research data, draw meaningful conclusions, and support evidence-based decision-making.

Module 5: Research Ethics, IPR, and Technical Publications

Introduces the ethical, legal, and professional aspects of engineering research. It covers research integrity, plagiarism, intellectual property rights (IPR), patents, copyrights, and scholarly publishing practices. Students will gain an understanding of ethical research conduct, protection of research outcomes, and the process of publishing technical papers in journals and conferences.

Module 6: Technical Writing and Research Presentation

Focuses on the effective communication of research findings through technical writing and professional presentations. It covers the preparation of research papers, technical reports, theses, and research proposals, along with citation and referencing practices.

This course introduces students to the fundamentals of engineering research, covering research concepts, literature review, problem formulation, research methodology, data collection and statistical analysis. It also emphasizes research ethics, intellectual property rights, technical writing, publication, and research presentation, enabling students to systematically conduct, document, and communicate engineering research.

	After successful completion of the course, the students will be able to	
Course Outcomes	CO1	Explain the fundamental concepts, types, and processes of research in engineering and technology.
	CO2	Identify research problems, formulate objectives and hypotheses, and conduct a comprehensive literature review using scientific databases.
	CO3	Design appropriate research methodologies, experiments, and data collection procedures for engineering investigations.
	CO4	Apply statistical and analytical techniques to collect, process, analyze, and interpret research data.
	CO5	Demonstrate awareness of research ethics, plagiarism, intellectual property rights (IPR), and publication practices.
	CO6	Prepare and present research proposals, technical reports, dissertations, and research papers in a professional format.

Syllabus:

Module No.	Unit No.	Topics	Hours
1		**Introduction to Engineering Research Concepts and Types of Research**	4
		After completing this module, students will be able to: 1. Students will understand the fundamentals and scope of engineering research.	
	1.1	Meaning, objectives, and significance of research; Types of research: Basic research, Applied research, Experimental research, Computational research, Action research; Characteristics of good research, Research process and scientific method, Engineering research challenges and opportunities	
		Self-Learning Topics: Emerging Research Paradigms, Research Impact Assessment, Global Research Trends, AI in Engineering Research, Technology Transfer.	
2		**Literature Review and Research Problem Formulation**	4
		After completing this module, students will be able to: 1. Students will be able to formulate research problems and conduct literature reviews.	
	2.1	Identification of research problems, Selection of research topics, Research objectives and questions, Hypothesis formulation, Research gap identification, Sources of technical literature, Literature survey techniques, Citation and referencing methods, Use of digital databases and search engines.	
		Self-Learning Topics: Research Question Development Frameworks, AI-Assisted Literature Reviews, Critical Analysis of Research Papers	
3		**Research Design and Methodology**	8
		After completing this module, students will be able to: 1. Students will be able to design suitable methodologies for engineering research.	
	3.1	Research design concepts, Quantitative and qualitative research methods, Experimental research design, Survey methods, Case study approach, Modeling and simulation-based research, Variables and measurement scales, Reliability and validity.	
		Self-Learning Topics: Bibliometric Analysis, AI-Assisted Research	
4		**Data Collection and Statistical Analysis**	6

	After completing this module, students will be able to: 1. Students will be able to collect and analyze engineering data.		
	4.1	Sources of data, Primary and secondary data, Sampling methods, Questionnaire design, Data coding and classification, Descriptive statistics, Probability distributions, Hypothesis testing, Correlation and regression analysis, Analysis of Variance (ANOVA)	
	Self-Learning Topics: Data Ethics and Privacy, Open Data Repositories, Uncertainty Analysis, and Perform exploratory analysis on an open dataset.		
5	**Research Ethics, IPR, and Technical Publications**		4
	After completing this module, students will be able to: 5.Students will understand ethical and legal aspects of research.		
	5.1	Research ethics, Plagiarism and academic integrity, Ethical issues in engineering research, Intellectual Property Rights (IPR), Patents, copyrights, trademarks, Patent search and filing basics, Journal publications and conferences, Peer-review process.	
	Self-Learning Topics: Responsible Conduct of Research, Ethics in Emerging Technologies, Open Science Practices, Predatory Publishing, Research Visibility and Scholarly Identity		
6	**Technical Writing and Research Presentation**		4
	After completing this module, students will be able to: 1. Students will be able to communicate research findings effectively.		
	6.1	Scientific Writing, Research proposal writing and Funding Agencies, Thesis and dissertation preparation, writing journal papers, Preparation of tables, figures, and references IEEE and APA, MLA, Chicago Style and Harvard, Vancouver referencing styles, Use of Reference Management Tools (Zotero, Mendeley, EndNote), Responding to Reviewers' Comments and Manuscript Revision, Research Communication through Blogs and social media.	
	Self-Learning Topics: Graphical Abstracts and Infographics, Digital Tools for Academic Writing, Peer Review and Manuscript Revision		
		Total	**30**

Assessment Methodology:

Type of Assessment	Assessment Tools

Term Work (50 Marks)	Any two pedagogy (10 marks each) for an individual student. 8. Literature Review Assignment 9. Statistical Analysis Using Software 10. Patent Search Activity The following technical Group activity/Tasks of 10 Marks each to be completed during the semester. ● Technical Paper Review and Presentation ● Research Problem Identification Exercise ● Research Proposal Preparation So Total 50 Marks are assigned as a part of term work in the course.

Text Books:

1. Research Design: Qualitative, Quantitative, and Mixed Methods Approaches, John W. Creswell, J. David Creswell 6th Edition, SAGE Publications, 2023.
2. Research Methodology: Methods and Techniques, C. R. Kothari, Gaurav Garg, 5th Edition, New Age International Publishers, 2024
3. Research Methodology for Engineers, by R. Ganesan, MJP Publishers, 2024

Useful Links:

3. https://onlinecourses.nptel.ac.in/e-learning/preview/noc25_ge66
4. https://onlinecourses.nptel.ac.in/e-learning/preview/noc26_ge79
5. https://ocw.mit.edu/courses/esd-932-engineering-ethics-spring-2006/
6. https://ocw.mit.edu/courses/15-348-doctoral-seminar-in-research-methods-ii-spring-2004/

Course Code	Course Name	Teaching Scheme (Hrs. / Week)			Credits Assigned			
		L	P	T	L	P	T	Total
25ILAEC02	Professional Communication Skills	*2+2	-	-	2	-	-	4
		Examination Scheme						
			CA	MSE	ESE	TW	OR	PR	Total
		Theory	-	-	-	-	-	-	00
		Lab	-	-	-	50	-	-	50
		Total	50						

Pre-Requisite Courses	25FEAEC01 Effective Communication Skills
	Theory of Communication, Communication cycle, effective use of body language and barriers to communication

Pre-Requisites	25FEAEC01 Effective Communication Skills
	Theory of Communication, Communication cycle, effective use of body language and barriers to communication

Course Overview

This course is designed to develop professional communication competencies required in academic, technical, corporate, and global workplace environments. The course focuses on enhancing verbal, written, interpersonal, digital, and presentation communication skills essential for engineering graduates and professionals. Students will develop the ability to communicate effectively in professional settings through technical writing, presentations, group discussions, workplace interactions, and digital communication platforms.

The course also emphasizes leadership communication, critical thinking, emotional intelligence, corporate etiquette, teamwork, and interview readiness. Practical activities such as mock interviews, presentations, group discussions, report writing, and role play simulations provide experiential learning opportunities to prepare students for placements, internships, entrepreneurship, and professional careers.

Module 1: Advanced Professional Communication

Develops advanced professional communication competencies required in workplace and corporate environments. It focuses on persuasive communication, negotiation skills, emotional intelligence, workplace ethics, and cross-cultural communication. Students will learn to communicate strategically, handle professional interactions effectively, and resolve workplace conflicts through appropriate

communication practices.

Module 2: Technical and Business Writing

Emphasizes professional writing skills required for technical and corporate communication. Students will learn technical report writing, proposal drafting, executive summaries, research documentation, professional emails, meeting documentation, and AI-assisted writing practices used in industry and academic environments

Module 3: Corporate Speaking & Group Communication

Develops professional speaking and interpersonal communication abilities. It focuses on presentations, group discussions, panel discussions, interview communication, storytelling techniques, and corporate meeting simulations. Students will strengthen public speaking confidence, teamwork communication, and leadership interaction skills.

Module 4: Digital Communication & AI Tool

Introduces modern digital communication platforms and AI-based communication tools used in professional environments. Students will learn professional networking, digital etiquette, prompt engineering basics, AI-assisted content generation, online collaboration, and personal branding strategies for digital workplace communication.

Module 5: Placement and Career Readiness

Prepares students for internships, placements, and professional careers through resume building, portfolio development, interview preparation, corporate professionalism, and workplace communication practices. Students will gain confidence in handling HR, technical, behavioral, and stress interviews effectively.

Module 6: Leadership, Innovation & Global Workplace Skills

Focuses on leadership communication, innovation pitching, entrepreneurship communication, design thinking, crisis communication, and multicultural workplace interaction. Students will develop the communication competencies required for leadership roles, global collaboration, and future workplace environments

This course develops professional communication skills required in workplace, technical, digital, and global professional environments by strengthening students' verbal, written, interpersonal, and presentation abilities. It equips students with business writing, report writing, professional documentation, group communication, and corporate speaking skills for effective workplace interaction. The syllabus also focuses on digital communication, AI tools, professional networking, career readiness, resume building, and interview preparation to enhance employability. Finally, it prepares students for leadership roles through innovation pitching, crisis communication, and future workplace skills.

	After successful completion of the course, the students will be able to	
Course Outcomes (COs)	CO1	Define the fundamentals of professional, technical, digital, and workplace communication.
	CO2	Explain communication models, professional ethics, corporate communication practices, and workplace interaction strategies.

	CO3	Apply verbal, written, presentation, and digital communication skills in academic, technical, and professional contexts.
	CO4	Analyze workplace communication situations, group discussions, negotiation scenarios, and professional challenges to identify suitable communication strategies.
	CO5	Evaluate leadership communication, ethical communication practices, and cross-cultural workplace interactions in professional environments.
	CO6	Create technical reports, business documents, presentations, resumes, portfolios, and AI-assisted communication content for industry readiness.

Syllabus:

Module No.	Unit No.	Topics	Hours
1	**Introduction to Professional Communication**		6
	After completing this module, students will be able to: 1. Define advanced workplace communication concepts and communication models. 2. Analyze workplace communication scenarios and conflict situations. 3. Demonstrate emotional intelligence and ethical communication practices. 4. Communicate effectively in multicultural and global workplace settings.		
	1.1	Fundamentals of Professional Communication: Types of professional communication and principles of effective communication.	
	1.2	Workplace Communication Skills: Formal and informal workplace communication, Internal and external communication, Corporate communication structure, Professional interaction techniques	
	1.3	Persuasive Communication and Influencing Skills: Persuasion techniques, Negotiation communication, Influencing strategies Assertive communication	
	1.4	Emotional Intelligence, Interpersonal Skills and Professional Etiquettes: Ethical communication practices, Professional behavior and etiquette, corporate responsibility in communication and Confidentiality and professionalism.	
	Self-Learning Topics: Watch corporate communication videos and prepare observation reports.		
	Technical and Business Writing		

2	**After completing this module, students will be able to.** 1. Draft technical reports, proposals, and executive summaries professionally. 2. Prepare professional emails, meeting agendas, and minutes. 3. Apply concise and structured business writing principles.		8
	2.1	Technical Report Writing: Structure of technical reports, Formal writing style, Data presentation techniques, Report formatting standards.	
	2.2	Business Correspondence: Professional email writing, Escalation and response mails, and corporate communication drafting.	
	2.3	Technical proposal Writing: Proposal writing structure, Writing executive summaries, Parts a technical proposal documentation standard.	
	2.4	Business Meeting Documentation: Notice and agenda preparation, writing meeting minute, Documentation practices, Official communication records, conducting business meeting and AI tools for professional writing Content refinement techniques, Grammar and style checking tools, Ethical use of AI in documentation	
	Self-Learning Topics: Create documentation using AI tools.		
3	**Corporate Speaking & Group Communication**		6
	After completing this module, students will be able to. 1. Deliver professional and impactful presentations confidently. 2. Participate effectively in group discussions, panel discussions, and meetings. 3. Demonstrate leadership and teamwork communication skills. 4. Apply storytelling and audience engagement techniques in presentations. 5. Handle interview communication professionally.		
	3.1	Professional Presentation Skills**:** Presentation structure Visual aids and slide design, Voice modulation and body language, Audience engagement techniques.	
	3.2	Group Discussions**:** GD strategies and participation, Analytical and critical discussion skills, Leadership during discussions Professional interaction techniques.	
	3.3	Team Communication and Collaboration: Team interaction methods, Collaborative communication, Workplace coordination, Communication in projects, Storytelling techniques, Persuasive presentation methods, corporate storytelling and Influencing audience perception	
	3.4	Interview Communication Mastery: HR interview communication, technical interview interaction, Behavioral interview handling and Mock interview practice	

	Self-Learning Topics: Record and analyze personal presentation videos.		
4	**Digital Communication & AI Tool**		6
	After completing this module, students will be able to: 1. Use professional digital communication platforms effectively. 2. Build professional online profiles and digital presence. 3. Apply AI tools for presentations, reports, and communication tasks. 4. Demonstrate digital etiquette and cyber professionalism. 5. Develop personal branding and online reputation management strategies.		
	4.1	Digital Communication Platforms: Professional communication tools, Email and virtual communication, Online collaboration platforms and Digital workplace communication	
	4.2	LinkedIn and Professional Networking: LinkedIn profile optimization, Professional networking strategies, Building online professional identity and Networking etiquette	
	4.3	AI Tools for Communication: AI tools for presentations, AI-assisted report writing, Communication automation tools and Productivity enhancement using AI	
	4.4	Digital Etiquette and Cyber Professionalism: Online professional behavior, Cyber ethics and safety, Responsible digital communication and Professional conduct in virtual environments, Personal branding strategies, Online presence management, Digital reputation building and Professional portfolio creation	
	Self-Learning Topics: Build personal online portfolio		
5	**Placement and Career Readiness**		6
	After completing this module, students will be able to. 1. Prepare ATS-friendly resumes and professional portfolios. 2. Demonstrate effective communication during HR and technical interviews. 3. Apply the STAR method while answering interview questions. 4. Exhibit corporate professionalism and workplace readiness. 5. Communicate confidently during placement and internship processes.		
	5.1	Resume and Portfolio Development: Resume writing techniques, ATS-friendly resume preparation, cover letters, Portfolio building methods and Career documentation standards	
	5.2	Placement Communication Skills: Internship communication, Placement interaction strategies, corporate communication readiness and Professional self-introduction.	

	5.3	HR and Technical Interviews: HR interview preparation, technical interview handling, Stress interview techniques, Behavioral interview strategies, Structured interview responses, Situation-based communication and Answer organization techniques	
	5.4	Corporate Professionalism and Workplace Readiness: Workplace expectations Professional ethics and conduct, Office communication etiquette and Time and task communication, Career planning communication, Workplace adaptability, Professional relationship management, and Industry communication practices.	
	Self-Learning Topics: Watch placement interview sessions and prepare reflection reports.		
6	**Leadership, Innovation & Global Workplace Skills**		4
	After completing this module, students will be able to. 1. Demonstrate leadership and innovation communication skills. 2. Present entrepreneurial and technical ideas effectively. 3. Apply design thinking and workplace problem-solving communication. 4. Handle crisis communication situations professionally. 5. Work effectively in multicultural and global professional environments.		
	6.1	Leadership Communication: Leadership communication styles, Motivational communication, Decision-making communication and Team leadership interaction.	
	6.2	Innovation and Entrepreneurial Communication: Startup pitching techniques, Entrepreneurial communication and Innovation presentations and Business idea communication	
	6.3	Crisis Communication and Media Handling: Crisis management communication, Media interaction basics, Reputation management and Emergency communication strategies.	
	6.4	Future Workplace Skills and Industry 5.0 Communication: Future communication trends, Human-AI workplace collaboration, Smart workplace communication and Industry 5.0 professional competencies.	
	Self-Learning Topics: Study successful startup presentations.		
		Total	**30**

List of Assignments

1. Case studies on ethical communication practices.
2. Draft a technical proposal on a given topic.
3. Write a **1-minute self-introduction** for an interview.
4. Short note on AI Tools for communication.
5. Prepare a cover letter with resume for the given position

6. Write a detailed note on industry 5.0 professional competencies.

Suggested List of Activities:

Experiment No.	Title of the Experiment
1	**Mock GD 1 (Group activity : The group of students will participate in group discussion on general and current affair topics)** **Objective:** To develop students' ability to participate actively and effectively in group discussions. **Outcome:** Students will be able to present opinions logically, listen actively to others, and contribute meaningful ideas during group discussions.
2	**Business Meeting (Group activity : Students will conduct business meetings on given situations or topics. They will play role of chairperson and participants)** **Objective:** To train students in professional communication and collaboration during formal business meetings. **Outcome:** Students will be able to participate professionally in meetings, communicate ideas clearly, and demonstrate teamwork and meeting etiquette.
3	**Mock GD 2 (Group activity : The group of students will participate in group discussion on given abstract topics)** **Objective:** To enhance students' confidence and communication skills in group discussion settings. **Outcome:** Students will be able to articulate viewpoints confidently, support arguments with relevant points, and engage constructively in discussions.
4	**Presentation on Interpersonal Skills (Individual Activity : Students will deliver presentations on interpersonal skills)** **Objective:** To develop students' presentation and interpersonal communication skills. **Outcome:** Students will be able to organize and deliver structured presentations while demonstrating effective verbal and non-verbal communication skills.
5	**Mock Interviews (Individual Activity : Students will be interviewed by panel or jury)** **Objective:** To prepare students for professional interviews through practice and self-assessment. **Outcome:** Students will be able to answer interview questions confidently, exhibit professional behavior, and communicate their strengths effectively.
6	**Creation of LinkedIn profile (Individual Activity : Students will create LinkedIn profile and present it)** **Objective:** To guide students in building a professional online presence for career opportunities. **Outcome:** Students will be able to create a professional LinkedIn profile showcasing

	their qualifications, achievements, skills, and career interests effectively.
7	**Debate on emerging technologies (Group Activity : Students will debate on emerging technologies and current issues)** **Objective:** To encourage critical thinking and effective public speaking through debate activities. **Outcome:** Students will be able to express arguments logically, defend viewpoints confidently, and participate actively in formal debates.
8	**Pitching a startup idea / Pitching idea presentation (Individual activity / group activity : Students individually or as a group present their startup idea)** **Objective:** To enable students to present innovative business ideas persuasively and professionally. **Outcome:** Students will be able to explain a startup idea clearly, justify its feasibility and value, and deliver an effective pitch presentation.
9	**Final Group Discussion (Group activity : The group of students will discuss on givtopics)** **Objective:** To assess a students' overall group discussion and communication skills. **Outcome:** Students will be able to participate confidently in discussions, present relevant ideas effectively, and demonstrate collaborative communication skills.
10	**Final presentation on Project (Group presentation : Students will present the section of the group project that they contribute)** **Objective:** To develop students' ability to present project work in a structured and professional manner. **Outcome:** Students will be able to prepare and deliver organized project presentations with clarity, confidence, and effective audience engagement.

Assessment Methodology:

Course - Laboratory	**TW-50**	• Active Participation (Lab) = 05 marks • Assignments = 10 marks • Activities = 10 marks Total = 25 marks Based on the performance & satisfactory completion of all activities ● Book Report = 05 marks\ ● Presentation + 10 Marks ● Final Group Discussion =10 Marks ● Total = 25 Marks

		(A group of minimum 6-7 students will draft a group project book report on given topics) ● GD, Business Meetings and Mock Interviews will be evaluated by the respective departments' faculty or jury

Text Books:

1. Business Communication Today – Bovee & Thill
2. Technical Communication – Meenakshi Raman

Reference Books:

1. Business Communication Today – Bovee & Thill
2. Technical Communication – Meenakshi Raman
3. HBR Guide to Persuasive Presentations
4. Crucial Conversations – Patterson et al.
5. The Quick and Easy Way to Effective Speaking – Dale Carnegie

2. Rubrics for Assignments:

Sr No	Criteria	10 Marks
1	Content Quality	03
2	Understanding of Topic	02
3	Structure & Organization	02
4	Grammar & Language	02
5	Submission & Formatting	01

3. Activity Rubrics

Activity No	Activity Rubrics
1	Mock GD 1 (10 Marks) ● Content knowledge 03 ● Team interaction 02 ● Body language 02 ● Confidence 02 ● Active participation 01
2	Business Meetings (10 Marks) ● Participation 02

	• Communication clarity 03 • Professionalism 03 • Team work 02
3	Mock GD 1 (10 Marks) • Content knowledge 03 • Team interaction 02 • Body language 02 • Confidence 02 • Active participation 01
4	Presentation on Interpersonal Skill (10 Marks) • Content quality 03 • Delivery 03 • Confidence 02 • Q&A Handling 02
5	Mock Interviews (10 Marks) • Communication 03 • Confidence 03 • Body language 02 • Answer quality 02
6	Creation of LinkedIn profile (10 Marks) • Profile completeness 03 • Professional Presentation 03 • Content Quality 02 • Profile visibility 02
7	Debate (10 Marks) • Content knowledge 03 • Communication 03 • Confidence 02 • Body language 02
8	Pitching a startup idea / Pitching idea presentation (10 Marks) • Content quality 03 • Delivery 03 • Confidence 02 • Q&A Handling 02

9	Final Group Discussion (10 Marks) ● Content knowledge 03 ● Team interaction 02 ● Body language 02 ● Confidence 02 ● Active participation 01
10	Final presentation on Project (10 Marks) ● Content quality 03 ● Delivery 03 ● Confidence 02 ● Q&A Handling 02

Course Code	Course Name	Teaching Scheme (Hrs. / Week)			Credits Assigned			
25ITIL3CEP03	Prototyping Of Systems & Solutions	L	P	T	L	P	T	Total
		–	2	–	–	1	–	1
		Examination Scheme						
		Theory		CA	MSE	ESE	Total	
				–	–	–	–	
		Lab/Tut.		TW	OR	PR	Total	
		25	–	25	–	–		
		Total		50				

Pre-Requisite Courses:	25ITIL3PCC02 – Web Technology (WT)
	25FE2PCC04 – Data Structures and Algorithms
	Operating System, Linux Administration, and Software Engineering Fundamentals

Course Overview: The Prototyping Of Systems & Solutions using DevOps is a Community Engagement Project (CEP) which is practice-oriented, lab-based course that bridges DevOps principles with real-world community problem solving. Students form teams of 3–4, identify a socially relevant problem, and develop an automated, cloud-deployable solution using a modern DevOps toolchain. The project spans the full DevOps lifecycle — from version control and CI/CD pipelines to containerisation, configuration management, security scanning, and monitoring — while documenting societal impact aligned with UN Sustainable Development Goals (SDGs).

Module 1 – DevOps Foundations and Project Scoping: Students study core DevOps principles, the software delivery lifecycle, and Agile frameworks (Scrum, Kanban). They conduct community surveys or field visits to identify a real need, define a focused problem statement, and map it to at least one SDG. A Gantt/PERT/CPM implementation plan is prepared and approved by the guide.

Module 2 – Version Control and Collaborative Development: Teams set up Git repositories (GitHub/GitLab), apply branching strategies, perform code reviews via pull requests, and establish contribution guidelines. All project source code (frontend and backend) is maintained under version control with meaningful commit history.

Module 3 – Continuous Integration with Jenkins: Students configure Jenkins CI pipelines using Maven/Gradle/Ant to automate builds, run unit tests, and generate build reports on every commit. Jenkins master–slave architecture is used to distribute builds for the community project application.

Module 4 – Continuous Testing with Selenium: Automated test suites are created using Selenium WebDriver and integrated with Jenkins pipelines. Test cases validate critical UI workflows of the community application, and test results are reported automatically on each build trigger.

Module 5 – Containerisation with Docker: The community application is containerised using Docker. Students write Dockerfiles for the frontend and backend, publish images to Docker Hub, and orchestrate multi-service deployment using Docker Compose. Container lifecycle management is practiced in the context of the project.

Module 6 – Configuration Management and Cloud Deployment: Infrastructure provisioning is automated using Puppet or Ansible. Students deploy the application stack on a cloud platform (AWS/GCP/Azure) and optionally implement Kubernetes orchestration, Terraform IaC, and a CI/CD pipeline ending in cloud deployment. Continuous monitoring using Nagios or equivalent tools completes the DevOps loop.

This CEP equips students with hands-on DevOps competencies embedded within a socially impactful project context. The course builds professional readiness through iterative development, peer collaboration, logbook documentation, and formal review-based assessment aligned with the DBIT DB25-V1 scheme.

Course Outcomes	After successful completion of the course, the students will be able to ..	
	CO1	Recall and describe fundamental DevOps principles, SDLC phases, Agile frameworks, and community engagement concepts relevant to technology-driven social projects. (Remembering)
	CO2	Explain the socio-economic and environmental context of a chosen community problem and outline a DevOps-powered IT intervention mapped to at least one UN SDG. (Understanding)
	CO3	Apply DevOps tools and frameworks — Git, Jenkins, Docker, Selenium, Puppet/Ansible — to build and deliver a CI/CD pipeline for a community-focused mini-project. (Apply)
	CO4	Analyse community needs, stakeholder expectations, and infrastructure constraints to design an automated, scalable, and sustainable DevOps solution. (Analyse)
	CO5	Evaluate the proposed DevOps solution in terms of deployment reliability, pipeline efficiency, security posture, societal impact, and alignment with Sustainable Development Goals (SDGs). (Evaluate)
	CO6	Create and demonstrate a fully containerised, CI/CD-enabled community project application with documentation of the DevOps lifecycle, societal impact, and future scope. (Design)

Hardware and Software Requirements:

Hardware Requirements	Software Requirements	Other Requirements
PC / Laptop with: • Intel i5 Core or above	• Linux (Ubuntu 24.04 LTS) / Windows 10+ • VirtualBox / VMware Workstation	• Active internet connection for package installation • GitHub / GitLab account (free)

• 8 GB RAM or above (recommended) • 500 GB HDD or SSD • Network Interface Card	• Git, Jenkins (LTS), Maven / Gradle • Docker Engine and Docker Compose • Puppet Enterprise / Open Source • VS Code / IntelliJ IDEA • SonarQube (optional)	• Docker Hub account (free) • AWS Free Tier account (for cloud experiments) • Postman (API testing, free) • Nagios or equivalent monitoring tool

Detailed Lab Syllabus:

Module No.	Unit No.	Topics / Lab Content	Hours.
1	**DevOps Foundations and Project Scoping**		4
	After completion of this module the student will be able to: 1. Define DevOps, its principles, phases of the SDLC, and the role of a DevOps engineer. 2. Identify Agile methodologies (Scrum, Kanban) and their relation to continuous delivery. 3. Conduct community surveys or field visits and formulate a focused problem statement. 4. Map the identified community problem to at least one UN Sustainable Development Goal (SDG) and prepare a Gantt/PERT project plan.		
	1.1	DevOps Fundamentals: Principles and Culture, SDLC Phases, Agile vs DevOps, Minimum Viable Product (MVP), Cross-functional Teams, DevOps Engineer Roles and Responsibilities	
	1.2	Community Problem Identification: Conducting Surveys / Interviews / Field Visits, Formulating a Problem Statement, Mapping to UN SDGs, Stakeholder Analysis	
	1.3	Project Planning: Gantt Chart / PERT Chart / CPM Chart preparation, Weekly Activity Scheduling, Logbook setup and Maintenance, Multiple Solution Proposal and Feasibility Analysis	
	Self-Learning Topics: Scrum ceremonies and artefacts (Sprint Planning, Backlog Grooming, Retrospective); Kanban boards and WIP limits in digital tools (Trello, Jira)		
	Version Control and Collaborative Development		
	After completion of this module the student will be able to: 1. Install Git and create a GitHub/GitLab account for project version control. 2. Perform core Git operations: clone, branch, merge, rebase, and conflict resolution.		

2	3. Implement Git workflow strategies (Feature Branch, GitFlow) for team collaboration. 4. Maintain a clean project repository with meaningful commits, pull requests, and code reviews.		4
	2.1	Git Installation and Setup: Version Control Concepts, Git vs SVN, git init/clone/add/commit/push/pull, Working with Remote Repositories, .gitignore, SSH Keys, GitHub/GitLab Account Setup	
	2.2	Branching and Merging: Creating and Switching Branches, Merging Strategies (Fast-forward, 3-way), Rebasing, Resolving Merge Conflicts, Git Stash, Git Cheat Sheet	
	2.3	Git Workflow and Collaboration: Feature Branch Workflow, GitFlow Strategy, Forking and Pull Requests, Code Review Process, Git Operations in IDE (VS Code / IntelliJ), Project Repository Structure for Frontend and Backend	
	Self-Learning Topics: AWS CodeCommit as a managed Git service; Mercurial and Subversion as alternative VCS tools; Bitbucket Pipelines for integrated CI/CD		
3	**Continuous Integration using Jenkins**		**4**
	After completion of this module the student will be able to: 1. Install and configure Jenkins with Maven/Gradle/Ant for automated build management. 2. Create Jenkins freestyle and pipeline jobs for the community project application. 3. Set up Jenkins Master–Slave architecture to distribute build workloads. 4. Automate build, test, and deployment stages using Jenkins Pipeline scripts (Declarative/Scripted).		
	3.1	Jenkins Introduction: CI/CD Concepts, Jenkins Architecture (Master–Agent), Installation and Initial Setup, Jenkins Dashboard, Plugin Management, Global Tool Configuration (JDK, Maven, Gradle)	
	3.2	Build Automation: Creating Freestyle Jobs, Maven/Gradle/Ant Build Integration, Build Triggers (Poll SCM, Webhooks), Build History and Console Output, Email Notifications on Build Failure	
	3.3	Jenkins Pipelines: Declarative vs Scripted Pipeline, Jenkinsfile Syntax, Stages and Steps, Parallel Stages, Adding Slave Nodes, Pipeline to Build–Test–Deploy Application on Tomcat Server, Pipeline for Community Project	
	Self-Learning Topics: Travis CI and GitHub Actions as cloud-native CI alternatives; Bamboo and GitLab CI/CD for enterprise pipelines; AWS CodePipeline integration		

4	**Continuous Testing with Selenium**		4
	After completion of this module the student will be able to: 1. Install Selenium WebDriver and configure it for automated browser testing. 2. Write and execute Selenium test cases for key UI workflows of the community application. 3. Integrate Selenium test suite with Jenkins pipelines using Maven for automated test execution. 4. Generate test reports and interpret test results within the CI pipeline.		
	4.1	Selenium Introduction: Test Automation Concepts, Selenium Architecture, WebDriver Installation (ChromeDriver/GeckoDriver), Browser Setup, First Selenium Script, Locating Elements (id, name, XPath, CSS Selector)	
	4.2	Test Case Development: Selenium WebDriver API, Handling Alerts and Iframes, Waits (Implicit, Explicit, Fluent), TestNG Framework Integration, Data-Driven Testing, Writing Test Cases for Community Application UI	
	4.3	Selenium–Jenkins Integration: Running Selenium Tests in Jenkins using Maven, Configuring Maven Surefire Plugin, Viewing Test Reports in Jenkins (TestNG/JUnit Report Plugin), Debugging Failed Tests in CI Pipeline	
	Self-Learning Topics: Cucumber BDD framework for behaviour-driven test scenarios; JUnit 5 features for unit testing in Java; Playwright as a modern cross-browser automation alternative		
5	**Continuous Deployment: Containerisation with Docker**		5
	After completion of this module the student will be able to: 1. Explain Docker architecture, container lifecycle, and the difference between images and containers. 2. Write Dockerfiles to containerise the frontend and backend of the community project application. 3. Manage Docker images and containers using Docker CLI commands. 4. Compose multi-container deployments using Docker Compose and publish images to Docker Hub.		
	5.1	Docker Fundamentals: Virtualisation vs Containerisation, Docker Architecture (Client–Daemon–Registry), Docker Installation, Core Commands (run, ps, stop, rm, exec, logs), Container Lifecycle, Docker Hub	
	5.2	Building Images: Dockerfile Instructions (FROM, RUN, COPY, EXPOSE, CMD, ENV, WORKDIR, VOLUME), Building Custom Images, Image Tagging and Versioning, Multi-stage Builds for Production Optimisation, Publishing Image to Docker Hub	

	5.3	Multi-Container Deployment: Docker Compose (docker-compose.yml structure, services, networks, volumes), Deploying Full-Stack Community Application with Compose (Frontend + Backend + Database), Container Health Checks, Docker Networking Basics	
	Self-Learning Topics: Docker Swarm for native container clustering and service orchestration; Docker Compose v3 advanced features (secrets, configs, deploy); Podman as a daemonless Docker alternative		
6	**Configuration Management and Cloud Deployment**		5
	After completion of this module the student will be able to: 1. Explain Infrastructure as Code (IaC) concepts and the role of Puppet/Ansible in configuration management. 2. Install and configure Puppet Master–Agent or Ansible Control–Node setup on Linux machines. 3. Write Puppet Manifests or Ansible Playbooks to provision and configure the community project environment. 4. Deploy the community project application on a cloud platform (AWS/GCP) with optional Kubernetes orchestration and Nagios-based continuous monitoring.		
	6.1	Configuration Management with Puppet: Puppet Architecture (Master–Agent), Installation on Linux, Puppet DSL (Manifests, Modules, Classes, Functions), Master–Agent Communication (SSL Certificates), Provisioning LAMP/MEAN Stack using Puppet Manifest	
	6.2	Cloud Deployment and IaC: Introduction to AWS Cloud Services (EC2, S3, RDS, IAM), DevOps on Cloud (CodeBuild, CodeDeploy, CodePipeline), Terraform Basics (Init, Plan, Apply, Destroy), Deploying Community Application Infrastructure on AWS using Terraform	
	6.3	Monitoring and Capstone Integration: Introduction to Continuous Monitoring, Nagios Core Installation and Configuration, Nagios Plugins (NRPE), Server and Service Monitoring, Complete DevOps Pipeline for Community Project (Git → Jenkins CI → Docker → Cloud Deploy → Nagios Monitor), Final Review and Documentation	
	Self-Learning Topics: Ansible as an agentless alternative to Puppet for configuration management; Kubernetes (kubectl, Deployments, Services) for container orchestration at scale; Splunk and Prometheus+Grafana for advanced monitoring and observability		
		Total	**26**

Guidelines for DevOps Community Engagement Mini Project:

Guidelines for DevOps Community Engagement Mini Project	
1. Group Formation	
•	Each group shall consist of 3–4 students.
•	Groups should collectively cover skills in DevOps tooling, scripting (Python/Bash), frontend, backend, database administration, and technical documentation.
2. Problem Identification and Need Assessment	
•	Conduct community surveys, interviews, or field visits to identify a real societal need or problem.
•	Convert the identified need into a clear, concise problem statement in consultation with the project guide.
•	Map the identified problem to at least one UN Sustainable Development Goal (SDG).
3. Planning and Documentation	
•	Prepare an implementation plan in the form of a Gantt chart / PERT chart / CPM chart covering weekly activities for the semester.
•	Maintain a logbook to record weekly progress, with guide's verification and feedback notes.
•	Propose multiple possible DevOps solutions and select the most feasible one based on stakeholder discussion.
4. Technical Scope and Expectations	
•	Minimum Pipeline Scale: At least 3–4 stages (Source → Build → Test → Deploy).
•	Suggested Technology Stack: Frontend: React.js or equivalent with API integration; Backend: Java Spring Boot / Node.js REST API; Database: MySQL / MongoDB; CI/CD: Jenkins with Maven/Gradle; Containerisation: Docker (Dockerfile + Docker Compose); Configuration Management: Puppet or Ansible; Cloud: AWS / GCP / Azure (optional); Monitoring: Nagios or equivalent.
•	Integration: End-to-end DevOps pipeline from source commit to deployed application.
•	Security: JWT-based authentication, SAST scanning using SonarQube/GitLab (if implemented).
•	Testing: Automated Selenium test suite integrated into Jenkins pipeline.
5. Development and Validation	
•	Convert the selected solution into a working containerised prototype with an automated CI/CD pipeline.
•	Validate the solution with real users/stakeholders and collect feedback.
•	Ensure the application and pipeline meet reliability, scalability, security, and accessibility standards.
•	Compile a final report in standard academic format: introduction, methodology, pipeline design, testing, results, societal impact, challenges, and future scope.

6. Project Deliverables	
•	Project Proposal: Problem statement, objectives, scope, target community, mapped SDG, and chosen DevOps toolchain.
•	Design Documentation: System architecture (MVC + API + CI/CD pipeline flow), Database ER diagram, UI wireframes/mockups, Docker architecture diagram.
•	Implementation: Source code repositories (GitHub/GitLab) for frontend and backend; Jenkinsfile (CI/CD pipeline script); Dockerfiles and docker-compose.yml; Puppet Manifests or Ansible Playbooks.
•	Testing: Selenium test cases and results (UI validation); API test results (Postman); Jenkins build and test reports.
•	Final Report: Problem background, DevOps pipeline design, implementation details, testing summary, deployment evidence, societal impact analysis, challenges faced, and future scope.
•	Presentation and Demo: 8–10 minute live demo including DevOps pipeline walkthrough, application demonstration, architecture explanation, and Q&A.
7. Assessment Process – Progress Monitoring	
•	At least two formal guide reviews per semester must be conducted:
•	Review 1: Problem finalization, SDG mapping, repository setup, and proposed DevOps toolchain.
•	Review 2: CI/CD pipeline implementation, Docker containerisation, testing results, and deployment demonstration.
8. Assessment Criteria	
•	Quality of community survey, need identification, and SDG mapping.
•	Clarity in problem definition, scope, and DevOps solution approach.
•	Innovativeness and novelty in the proposed DevOps-powered community solution.
•	Feasibility, completeness, and correctness of the CI/CD pipeline implementation.
•	Quality of containerisation (Dockerfile, Compose) and configuration management artefacts.
•	Societal impact, sustainability, and alignment with SDGs.
•	Quality of automated test coverage and pipeline integration.
•	Effective use of DevOps tools and standard engineering practices.
•	Individual contribution, collaboration, and leadership within the team.
•	Clarity in written documentation (logbook, report) and oral presentation during the final demo.

List of Experiments:

Sr. No.	Experiment Title and Details

1	**DevOps Orientation and Project Kick-off** **Objective:** To understand DevOps principles, SDLC, and the DevOps Engineer role. Conduct a community survey/field visit, define the problem statement, map to SDG, and set up a project logbook. **Outcome:** Students will be able to articulate the DevOps philosophy, identify a real community problem, and produce a structured problem statement with SDG mapping.
2	**Git Installation, GitHub Account Setup, and Repository Creation** **Objective:** To install Git, create a GitHub/GitLab account, and initialise the community project repository with proper structure and README. **Outcome:** Students will be able to set up a version-controlled project repository with meaningful initial commit, .gitignore, and project folder structure.
3	**Git Operations on Local and Remote Repositories** **Objective:** To practice core Git operations including branching, merging, rebasing, and conflict resolution using a Git Cheat Sheet on the community project codebase. **Outcome:** Students will be able to manage collaborative development using branches, resolve merge conflicts, and maintain a clean commit history.
4	**Jenkins Installation and CI Job Setup with Maven/Gradle/Ant** **Objective:** To install Jenkins, configure build tools (Maven/Gradle/Ant), and create a CI job that builds the community project application automatically on code commits. **Outcome:** Students will be able to configure Jenkins, create a build job linked to the project GitHub repository, and automate the build process.
5	**Jenkins Pipeline – Build, Test, and Deploy to Tomcat Server** **Objective:** To create a Jenkins Declarative Pipeline (Jenkinsfile) with stages for checkout, build, test, and deployment of the community application on a Tomcat server. **Outcome:** Students will be able to write and execute a multi-stage Jenkinsfile that automates the complete CI/CD cycle for the project.
6	**Jenkins Master–Slave Architecture and Distributed Builds** **Objective:** To configure a Jenkins Master–Slave (Agent) setup and distribute community project builds across slave nodes for parallel execution. **Outcome:** Students will be able to add and configure Jenkins agent nodes and run distributed pipeline jobs for scalable CI.
7	**Selenium Test Automation and Jenkins Integration** **Objective:** To create Selenium WebDriver test cases for critical UI workflows of the community application and integrate them into the Jenkins CI pipeline using Maven. **Outcome:** Students will be able to write and run automated UI tests in the CI pipeline and generate test reports automatically on every build.
8	**Docker Installation, Image Management, and Container Operations** **Objective:** To install Docker, explore Docker architecture, and perform container lifecycle operations (run, stop, exec, rm, logs) using CLI commands on sample applications.

	Outcome: Students will be able to manage Docker images and containers using CLI and understand the container lifecycle in the context of application deployment.
9	**Dockerising the Community Project Application (Dockerfile + Docker Compose)** **Objective:** To write Dockerfiles for the frontend and backend of the community project, build images, and orchestrate a full-stack deployment using Docker Compose. **Outcome:** Students will be able to containerise a multi-service application and deploy it using Docker Compose with defined networks and volumes.
10	**Puppet Manifests and Modules for Community Project Environment** **Objective:** To write Puppet Manifests using DSL constructs (Classes, Modules, Functions) to automate provisioning of the community project's server environment (web server, database, application dependencies). **Outcome:** Students will be able to author and apply Puppet Manifests that provision a complete application environment automatically.
11	**Capstone – End-to-End DevOps Pipeline and Project Demonstration** **Objective:** To integrate all DevOps tools into a complete, automated pipeline: Git → Jenkins CI → Docker Build → Container Deploy → Monitoring. Present the community project with a live pipeline execution, application demo, and societal impact discussion. **Outcome:** Students will be able to demonstrate a fully functional DevOps pipeline delivering the community project, and articulate the societal impact, SDG alignment, and future scope of the solution.

Assessment Methodology:

TW – 25	• Active Participation = 05 marks • Project Report = 10 marks (Introduction, Methodology, Pipeline Design, Testing, Results, SDG Impact, Conclusion) • Progress Presentations (minimum 02) and Demonstration = 10 marks (Review 1: Problem + Toolchain; Review 2: Pipeline + Demo)
PR – 25	• Practical and Oral Examination based on the DevOps pipeline, Dockerised application, and CI/CD demonstration. • Examiners may ask questions on any tool used in the project: Git, Jenkins, Docker, Selenium, Puppet/Ansible, Cloud deployment. • Live execution of at least one pipeline stage or Dockerfile build during the practical examination.
Term Work Journal	• Minimum 12 practicals completed and documented in the lab journal (Experiments 1–12). • At least 2 written assignments: (i) Case Study on Real-World DevOps Implementation in a company; (ii) Self-learning topic assignment from the syllabus. • Logbook with weekly progress entries, guide verification signatures, and feedback notes.

Textbooks:

1. DevOps Bootcamp, Sybex Learning.
2. Karl Matthias and Sean P. Kane, Docker: Up and Running, O'Reilly Publication.
3. Len Bass, Ingo Weber and Liming Zhu, DevOps: A Software Architect's Perspective, Addison-Wesley Pearson Publication.
4. John Ferguson Smart, Jenkins: The Definitive Guide, O'Reilly Publication.
5. Ryan Russell-Yates, Mastering Puppet 5: Optimise Enterprise-Grade Environment Performance with Puppet, Packt Publishing.
6. Yevgeniy Brikman, Terraform: Up and Running – Writing Infrastructure as Code, 2nd Edition, O'Reilly Media.
7. Brendan Burns, Kubernetes: Up and Running – Dive into the Future of Infrastructure, 2nd Edition, O'Reilly Media.

References:

1. Sanjeev Sharma and Bernie Coyne, DevOps for Dummies, Wiley Publication.
2. Httermann, Michael, DevOps for Developers, Apress Publication.
3. Joakim Verona, Practical DevOps, Packt Publication.
4. Martin Alfke, Puppet 5 Essentials – A Fast-Paced Guide to Automating Your Infrastructure, 3rd Revised Edition, Packt Publishing.
5. Amit Shah and Aurobindo Sarkar, Learning AWS – Design, Build, and Deploy Responsive Applications Using AWS, 2nd Edition.
6. Ajay Pherwani, Going Serverless with AWS Lambda: Leveraging the Latest Services from the AWS Cloud, Shroff/X-Team.

Useful Links:

1. https://docs.docker.com/ (Docker Official Documentation)
2. https://www.jenkins.io/doc/ (Jenkins Official Documentation)
3. https://git-scm.com/doc (Git Official Documentation and Book)
4. https://www.selenium.dev/documentation/ (Selenium Official Documentation)
5. https://puppet.com/docs/ (Puppet Official Documentation)
6. https://developer.hashicorp.com/terraform/docs (Terraform by HashiCorp Documentation)
7. https://kubernetes.io/docs/ (Kubernetes Official Documentation)
8. https://aws.amazon.com/training/ (AWS Free Training and Tutorials)
9. https://nptel.ac.in/courses/106/105/106105261/ (NPTEL – DevOps and CI/CD)
10. https://sdgs.un.org/goals (United Nations Sustainable Development Goals – Reference for SDG Mapping)