PERIYAR UNIVERSITY
PERIYAR PALKALAI NAGAR
SALEM 636 011

MASTER OF COMPUTER APPLICATIONS (M.C.A)

SEMESTER PATTERN

Under Choice Based Credit System

REGULATIONS AND SYLLABUS

FOR AFFILIATED COLLEGES

(Effective from the Academic year 2026-2027 onwards)

PERIYAR UNIVERSITY, SALEM – 636 011
MASTER OF COMPUTER APPLICATIONS (M.C.A.)
REGULATIONS AND SYLLABUS – 2026 - 2027

(Affiliated Colleges)
Effective from the academic year 2026 - 2027

MASTER OF COMPUTER APPLICATIONS (M.C.A)

The Master of Computer Applications (M.C.A.) is a postgraduate programme designed to develop skilled professionals capable of meeting the growing demands of the software industry through a strong foundation in computer applications, programming, and system design. The programme provides an in-depth understanding of key areas such as programming languages, data structures, database management systems, software engineering, computer networks, and web and mobile application development. It is structured to bridge the gap between theoretical computer science concepts and their practical implementation in real-world software solutions.

The programme places significant emphasis on application-oriented learning, encouraging students to gain hands-on experience through programming labs, mini-projects, and a major project or internship component. Students are also introduced to contemporary trends such as cloud computing, artificial intelligence, and enterprise application development, ensuring their readiness for the evolving IT landscape. Graduates of the M.C.A. programme are equipped with strong technical, analytical, and problem-solving abilities, enabling them to pursue careers as software developers, systems analysts, and IT consultants across various industry domains.

K) PROGRAMME OUTCOMES (POs)

PO1: Computational Knowledge: Apply mathematical foundations, data structures, algorithms, and core CS principles to analyse and solve complex real-world computing problems.

PO2: Software Engineering: Design, develop, test, and deploy software applications and systems using established software engineering principles and methodologies.

PO3: Modern Tools and Technologies: Select and effectively use modern computing tools, frameworks, cloud platforms, and IDEs to build industry-relevant applications.

PO4: Data and Intelligent Systems: Collect, store, process, and analyse data using database systems, big data tools, machine learning, and AI to build intelligent applications.

PO5: Professional Ethics and Security: Practise professional conduct, adhere to data privacy laws, intellectual property rights, cyber security principles, and ethical responsibilities.

PO6: Lifelong Learning and Innovation: Commit to continuous learning, adapt to evolving technologies, collaborate in teams, and apply innovative thinking for societal impact.

L) PROGRAMME SPECIFIC OUTCOMES (PSOs)

PSO1: Industry Readiness and Placement: Equip students with technical skills, problem-solving abilities, and professional competencies for employment in software companies and IT enterprises.

PSO2: Software Engineering Proficiency: Develop expertise in full-stack development, system design, agile methodologies, DevOps pipelines, and cloud-native architectures for enterprise-scale software.

PSO3: Research, Innovation and Development: Foster research aptitude by applying advanced algorithms and emerging technologies (AI, IoT, Cloud) to solve open-ended computing challenges.

PSO4: Data Science and Intelligent Systems: Apply machine learning, deep learning, generative AI, big data analytics, and intelligent agents for data-driven applications and decision-support systems.

PSO5: Entrepreneurship and Social Impact: Inspire students to launch technology start-ups and develop solutions addressing societal needs, promoting inclusion and sustainable development.

A) OBJECTIVE OF THE PROGRAMME

Enable the students to pursue lifelong multidisciplinary learning, function effectively on teams to accomplish a common goal and become innovative through technical advancement.

To meet dynamic global needs, the syllabus is focused on current industry-relevant technical concepts, to enrich the knowledge of students and enhance their readiness for employment, research and entrepreneurship.

B) CONDITIONS FOR ADMISSION

Candidates who have passed in any one of the following or equivalent are eligible to apply:

i) Bachelor's degree (under 10+2+3/4) in any subject with Mathematics at +2 level.

ii) Bachelor's degree (under 10+2+3/4-or-10+3-year Diploma + 3 year later entry BE) in any subject with Mathematics / Business Mathematics / Statistics as one of the subjects.

C) DURATION OF THE PROGRAMME

The duration of the degree of Master of Computer Applications shall consist of two academic years divided into four semesters.

D) EXAMINATIONS

The examination shall be of three hours duration for each course at the end of each semester. The candidate failing in any subject(s) will be permitted to appear in the subsequent examination.

The practical / project should be an individual work. The University examination for practical / project work will be conducted by the internal and external examiners jointly at the end of each semester.

D) EXAMINATIONS (Marks Distribution)

THEORY:

EVALUATION OF CONTINUOUS INTERNAL ASSESSMENT

Test	10 Marks
Seminar	05 Marks
Assignment	05 Marks
Attendance	05 Marks
Total	25 Marks

(No passing minimum)

EVALUATION OF EXTERNAL ASSESSMENT
QUESTION PAPER PATTERN

Time: 3 Hours Max. Marks: 75

PART- A: 15x1 = 15 marks
Answer all the questions
Three questions from each unit (Multiple Choice Questions)

PART- B: 2x5 = 10 marks
Answer any TWO questions
One question from each unit

PART- C: 5x10 = 50 marks
Answer all the questions
One question from each unit (either or type)

The Passing minimum shall be 50% out of 75 marks (38 marks)

PRACTICAL / MINI PROJECT:

EVALUATION OF CONTINUOUS INTERNAL ASSESSMENT

Test 1	15 Marks
Test 2	15 Marks
Record	10 Marks
Total	40 Marks (No passing minimum)

EVALUATION OF EXTERNAL ASSESSMENT

I) PRACTICAL

QUESTION PAPER PATTERN

Time: 3 Hours Max. Marks: 60

There will be two questions with or without subsections to be asked for the practical examination. Every question should be chosen from the question bank prepared by the examiner(s). Every sixth student should get a new question i.e. each question may be used for at most five students.

Distribution of Marks

Each question	30 Marks
Problem Understanding	05 Marks
Program writing	10 Marks
Debugging	10 Marks
For Correct Results	05 Marks

II) INDUSTRY LITERACY MINI PROJECT DEVELOPMENT

Viva-voce (Jointly)	30 Marks
Modification	30 Marks

Students should write about their Mini Project briefly.

i. Aim
ii. Features
iii. Modules
iv. Modification

III) PROJECT WORK

Continuous Internal Assessment	100 Marks
Evaluation & Viva-Voce (External)	300 Marks

E) REGULATIONS OF PROJECT WORK

- Students should do their Project work in a MNC Company / Research Institution during the IV semester.
- The Candidate should submit the filled in format as given in Annexure-I to the department for approval during the 1st week of December.
- Periodically the project should be reviewed.
- The student should submit three copies of their Project Report.
- A Sample format is enclosed in Annexure-II.
- Format of the Title page and Certificate are enclosed in Annexure III.
- The students may use power point presentation during their viva voce examination.

F) PASSING MINIMUM

The candidate shall be declared to have passed in the Theory / Practical / Project Work examination, if the candidate secures not less than 50% marks in EA and also in Total of the prescribed marks. However, submission of a record notebook is a must.

G) CLASSIFICATION OF SUCCESSFUL CANDIDATES

Candidates who obtain 75% and above in the aggregate shall be deemed to have passed the examination in First Class with Distinction provided they pass all the examinations prescribed for the programme at the first appearance. Candidates, other than the above, who secure not less than 60% of the aggregate marks in the whole examination, shall be declared to have passed the examination in First Class. The remaining successful candidates shall be declared to have passed in Second Class.

Candidates who pass all the examinations prescribed for the programme in first instance and within a period of two academic years from the year of admission are only eligible for University Ranking.

H) MAXIMUM DURATION FOR THE COMPLETION OF THE PROGRAMME

The maximum duration to complete the programme shall be three academic years after normal completion of the programme.

I) COMMENCEMENT OF THIS REGULATION

These regulations shall take effect from the academic year 2026-27, that is, for students who are admitted to the first year of the programme during the academic year 2026-27 and thereafter.

J) TRANSITORY PROVISION

Candidates who were admitted to the MCA programme of study before 2026-2027 shall be permitted to appear for the examinations under those regulations for a period of three years after completion of the programme. Thereafter, they will be permitted to appear for the examination only under the regulations then in force.

ANNEXURE - I

PERIYAR UNIVERSITY

Name of the College :

Programme :

Name of the Student :

Register Number :

Title of the Project Work :

Address of Organization / Institution :

Name of the External Guide :

Designation :

Place :
Date : Signature of External Guide

(with seal)

Name of the Internal Guide :

Qualification :

Teaching Experience :

Place :
Date : Signature of Internal Guide

Principal

ANNEXURE II

PROJECT DOCUMENTATION CURRICULUM FRAMEWORK

Chapter No.	Title	Page No.
	College Bonafide Certificate	i
	Company Attendance / Certificate	ii
	Declaration	iii
	Acknowledgement	iv
	Synopsis	v
	List of Figures	vi
	List of Tables	vii
1.	**INTRODUCTION**	
	1.1 Project Overview	
	1.2 Organization / Company Profile	
	1.3 System Specifications	
	1.3.1 Hardware Configuration	
	1.3.2 Software Specification	
2.	**SYSTEM ANALYSIS**	
	2.1 Existing System Description	
	2.2 Drawbacks and Limitations	
	2.3 Proposed System Description	
	2.4 Core Features & System Advantages	
3.	**SYSTEM DESIGN**	
	3.1 Input Interface Design	
	3.2 Output Interface Design / Report Formats	
	3.3 Code Design & Methodology	
	3.4 Logical Database Design (ER Diagrams)	
4.	**SYSTEM DEVELOPMENT & TESTING**	
	4.1 Description of Functional Modules	
	4.2 Testing Strategies (Unit & Integration Testing)	
5.	**CONCLUSION & FUTURE ENHANCEMENTS**	
	5.1 Conclusion	
	5.2 Scope for Future Enhancements	
6.	**BIBLIOGRAPHY / REFERENCES**	
	APPENDICES	
	A. Data Flow Diagrams (DFD) / UML Diagrams	
	B. Database Table Structures & Field Specs	
	C. Sample Source Code Snippets	
	D. Sample Input Screens	
	E. Sample Output Reports	

DEFINITIONS & LOGICAL SYLLABUS BREAKDOWN

Preliminary Pages (i to vii)

- **College Bonafide Certificate:** Official institutional document signed by the Project Supervisor, Head of the Department (HOD), and the Principal, formally certifying the originality of the project work completed by the candidate to fulfil degree regulations.
- **Company Attendance / Internship Certificate:** Mandatory corporate document issued on official industry letterhead validating the student's external technical training execution, attendance timelines, and project module completion.
- **Declaration:** Formal statement signed directly by the student affirming that the software application and corresponding documentation are entirely original works and have not been submitted for any other academic degree.
- **Acknowledgement:** Dedicated section expressing formal gratitude to the institutional administration, faculty guides, industry supervisors, and supporting individuals who provided necessary infrastructure and academic counselling.
- **Synopsis:** Comprehensive executive summary of the technical problem statement, the software development stack, structural architecture components, and definitive project outcomes.
- **List of Figures:** Systematic index mapping every graphic element, architecture block diagram, UML chart, and user interface snapshot alongside the exact page allocation.
- **List of Tables:** Structured index referencing data frameworks, technical specification grids, database schemas, field parameters, and system testing evaluation logs with cross-referenced page entries.

Chapter 1: Introduction

- **1.1 Project Overview:** High-level introduction clarifying the problem domain boundaries, intended target user audience, real-world utility, and overarching system design goals.
- **1.2 Organization / Company Profile:** Background details of the host organization or department, its field of operation, commercial domain, and the operational setting associated with the internship deployment.
- **1.3 System Specifications:** The engineering constraint envelope dictating hardware capability and developer dependency environments.
- **1.3.1 Hardware Configuration:** Infrastructure requirements including memory allocation, processor throughput, and storage subsystem requirements.
- **1.3.2 Software Specification:** Compilation variables including operating environment, runtime platform, build tools, and the database engine used.

Chapter 2: System Analysis

- **2.1 Existing System Description:** Documents the precise logic of current manual or legacy workflows in place before implementation.
- **2.2 Drawbacks and Limitations:** Critical diagnosis of data transmission issues, processing latency, storage redundancy, and access control gaps in the current system.
- **2.3 Proposed System Description:** Describes the automated solution and how procedural operations are modernized, hosted, and resolved through structured data control.
- **2.4 Core Features & System Advantages:** Tabulates functional improvements such as faster queries, automated alerts, secure authentication, and dynamic tracking.

Chapter 3: System Design

- **3.1 Input Interface Design:** Defines the form models, data capture fields, event triggers, and front-end validation rules.

- **3.2 Output Interface Design / Report Formats:** Lays out presentation layers, dashboards, downloadable document structures, and reporting formats.
- **3.3 Code Design & Methodology:** Defines the modular programming model, backend architecture style (e.g. MVC), class relationships, and file organization.
- **3.4 Logical Database Design (ER Diagrams):** Constructs the relationship schema mapping table definitions, primary/foreign keys, cardinalities, and normal forms.

Chapter 4: System Development & Testing

- **4.1 Description of Functional Modules:** Breaks down the software into distinct operational units, showing each module's pipeline.
- **4.2 Testing Strategies:** Establishes code compliance validation workflows, covering Unit Testing (independent verification of functions and routines using test data) and Integration Testing (verifying data exchange across application boundaries and endpoints).

Chapter 5: Conclusion & Future Enhancements

- **5.1 Conclusion:** Evaluates whether the project's objectives, as defined during the initial analysis phase, were successfully achieved.
- **5.2 Scope for Future Enhancements:** Identifies potential extensions, such as further automation, mobile scaling, or migration to a microservices architecture.

Chapter 6: Bibliography / References

- **Bibliography:** An indexed list of textbooks, research papers, academic manuals, and online documentation, cited in a consistent reference style.

Appendices

- **A. Data Flow Diagrams (DFD) / UML Diagrams:** Full-page diagrams charting contextual, Level-1, and Level-2 system dependencies, or class/use-case mappings.
- **B. Database Table Structures & Field Specs:** Dictionaries detailing field boundaries, data types, default values, keys, and storage parameters for every table.
- **C. Sample Source Code Snippets:** Highlights of key backend routes, core algorithms, or data security implementations.
- **D. Sample Input Screens:** Screen captures showing filled UI entry views during standard system interactions.
- **E. Sample Output Reports:** Visual reporting logs showing generated summaries, dashboards, and transactional reports.

*Subsections may be selectively adapted, omitted, or substituted with the guide's prior approval if a particular structural phase is not technically relevant to the core development architecture.

ANNEXURE III

A. Format of the Title Page

TITLE OF THE PROJECT WORK

A Project Work submitted in partial fulfilment of the requirements for the degree of

Master of Computer Applications

to the

Periyar University, Salem – 636 011

Submitted by

Name of the Student

Reg. No.

Under the Guidance of

Name of the Guide

(Designation, Name of the Department)

College Emblem

Name of the Department

College Name

(Affiliated to Periyar University)

Place with Pin Code

Month – Year

B. Format of the Certificate

College Name

(Affiliated to Periyar University)

Place with Pin Code

College Emblem

This is to certify that the Project Work entitled *Title of the Project* submitted in partial fulfilment of the requirements of the degree of **Master of Computer Applications** to the Periyar University, Salem is a record of Bonafide work carried out by *Name of the Student*, Reg. No. _____ under my supervision and guidance.

Head of the Department Internal Guide

Submitted for Viva-Voce Examination held on _____ at Name of the College, Place with Pin Code.

External Examiner Internal Examiner

M) STRUCTURE OF M.C.A. PROGRAMME UNDER CBCS PATTERN

Course Code	Title of the Course	Credits	Hours		Maximum Marks		
			Theory	Practical	CIA	EA	Total
SEMESTER I							
PCA01	Core I – Java Programming	4	4		25	75	100
PCA02	Core II – Relational Database Management System	4	4		25	75	100
PCA03	Core III – Discrete Mathematics	4	4		25	75	100
PCA04	Core IV – Advanced Computer Networks	4	4		25	75	100
PCAE01	Elective I – Data Analytics	4	4		25	75	100
PCAP01	Core V – Lab I: Java Programming Lab	3		5	40	60	100
PCAP02	Core VI – Lab II: RDBMS Lab	3		5	40	60	100
	Total	**26**	**20**	**10**			**700**
SEMESTER II							
PCA07	Core VII – Data Mining Techniques	4	4		25	75	100
PCA08	Core VIII – Python Programming	4	4		25	75	100
PCA09	Core IX – Advanced Operating System	4	4		25	75	100
PCAE02	Elective II – Advanced Web UI Frameworks	4	4		25	75	100
PCAEDC01	EDC I – Entrepreneurship Development	4	4		25	75	100
PCAP03	Core X – Lab III: Data Mining Lab	3		4	40	60	100
PCAP04	Core XI – Lab IV: Python Programming Lab	3		4	40	60	100

PHR01	Human Rights (Mandatory)	0	0		25	75	100
SWAYAM	Add On – SWAYAM / MOOC / Soft Skill	3	2		–	–	–
	Total	**29**	**22**	**8**			**800**
SEMESTER III							
PCA12	Core XII – Internet of Things	4	4		25	75	100
PCA13	Core XIII – .Net Programming	4	4		25	75	100
PCA14	Core XIV – Big Data Analytics	4	4		25	75	100
PCAE03	Elective III – Advanced Learning Algorithms & Statistical Modelling	4	4		25	75	100
PCAE04	Elective IV – Generative AI Engineering and Intelligent Agents	4	4		25	75	100
PCAP05	Core XV – .Net Programming Lab	3		5	40	60	100
PCAP06	Core XVI – Industry Literacy / Mini Project	3		5	40	60	100
	Total	**26**	**20**	**10**			**700**
SEMESTER IV							
PCAPR01	Core XVII – Project Work and Viva-Voce	15	–		100	300	400
	Total	**15**					**400**

SEMESTER – I
CORE – I - JAVA PROGRAMMING

Course Objectives:

The main objectives of this course are to:

1. Understand Java fundamentals including JVM architecture, OOP concepts, inheritance, interfaces, and exception handling.
2. Learn multithreading, JDBC database connectivity, and responsive web interface development using HTML, CSS, and React JS.
3. Develop server-side web applications using Java Servlets for request processing and session management.
4. Design dynamic web applications using JSP, Expression Language (EL), and JSTL.
5. Build enterprise applications using MVC architecture, Spring Framework, Spring Boot, Hibernate, and Maven.

Expected Course Outcomes:

On the successful completion of the course, student will be able to:

1	Apply Java programming concepts including classes, objects, inheritance, interfaces, packages, and exception handling to develop object-oriented applications.	K1, K2
2	Develop multithreaded applications, perform database operations using JDBC, and create responsive web interfaces using modern web technologies.	K2, K3
3	Design and implement server-side web applications using Java Servlets, including session management, database connectivity, and file handling.	K3, K4
4	Build dynamic web applications using JSP, Expression Language (EL), JSTL, and related technologies for interactive content generation.	K4, K5
5	Develop enterprise-level applications using Spring Framework, Spring Boot, Hibernate, Maven, and MVC architecture principles.	K5, K6

K1 - Remember; K2 - Understand; K3 - Apply; K4 - Analyze; K5 - Evaluate; K6 - Create

Unit:1	**JAVA FUNDAMENTALS**	**15 hours**

Java Virtual Machine, data types, variables, keywords, operators, expressions, control statements, classes, objects, constructors, access control, method overloading, static members, Arrays, Strings. Inheritance: types, constructors in inheritance, method overriding, use of super, abstract classes, interfaces, dynamic method dispatch. Packages and exception handling.

Unit:2	**MULTITHREADING, JDBC AND WEB CLIENT UI DEVELOPMENT**	**15 hours**

Java Thread Model, Concurrent Programming, Thread Life Cycle, Thread Priorities, Creating and Managing Threads, Thread Methods and Exceptions, Inter-thread Communication, Synchronization. JDBC Architecture and Drivers, CRUD Operations. Web Client UI: HTML and

CSS, Responsive Web Design using Bootstrap, JavaScript, React JS, Creating Reusable UI Components with React.		
Unit:3	**JAVA SERVLETS**	**15 hours**
Java EE Architecture – Application Servers and Containers, Servlet API and Lifecycle, Handling Client Requests and Responses – Working with Databases through Servlets –		
Unit:4	**JAVA SERVER PAGES AND JSTL JSP**	**15 hours**
Architecture and Lifecycle, Action Elements, Implicit Objects, Scripting Elements, Scope and EL (Expression Language), JSP Standard Tag Libraries (JSTL), Developing Dynamic Web Applications with JSP.		
Unit:5	**JAVA FRAMEWORKS**	**15 hours**
MVC Architecture, Spring Framework, Spring Modules, Spring MVC and Spring Boot, Hibernate Framework. Maven Installation, Maven Core Concepts – POM File, Maven Build Life Cycles, Phases and Goals – Maven Build Profiles.		

Mapping of Course Outcomes (COs) with Program Outcomes (POs):

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	2	2	1	1	3
CO2	3	2	2	1	1	3
CO3	2	1	2	1	2	2
CO4	3	3	3	2	2	3
CO5	3	3	3	2	2	3

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Mapping of Course Outcomes (COs) with Program Specific Outcomes (PSOs):

CO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	2	2	1	2
CO2	3	2	2	1	2
CO3	2	1	1	1	2
CO4	3	2	2	2	2
CO5	3	3	2	2	3

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Text Books:

- Schildt H., 'Java: The Complete Reference', 9th Ed., Tata McGraw Hill. (Covers Units: I - V)

https://books.google.com/books?q=Java%3A+The+Complete+Reference+Schildt+H.

- Balagurusamy E., 'Programming with Java: A Primer', 5th Ed., Tata McGraw Hill. (Covers Units: I - V) https://books.google.com/books?q=Programming+with+Java%3A+A+Primer+Balagurusamy+E.

Reference Books:

- Horstmann C.S. & Cornell G., 'Core Java Volume I – Fundamentals', Pearson Education. https://books.google.com/books?q=Core+Java+Volume+I+%E2%80%93+Fundamentals+Horstmann+C.S.+%26+Cornell+G.
- Schildt H., 'Java: A Beginner's Guide', 6th Ed., Tata McGraw Hill. https://books.google.com/books?q=Java%3A+A+Beginner+Schildt+H.
- Moraes E., 'Java EE 8 Cookbook', Packt Publishing. https://books.google.com/books?q=Java+EE+8+Cookbook+Moraes+E.

Free Online Study Resources:

- Think Java 2nd Ed. – Allen B. Downey (Free E-Book) – https://greenteapress.com/wp/think-java-2e/
- Oracle Java SE Tutorials (Free) – https://docs.oracle.com/javase/tutorial/

SEMESTER – I
CORE – II - RELATIONAL DATABASE MANAGEMENT SYSTEM

Course Objectives:

The main objectives of this course are to:

1. Understand the fundamentals of database systems, ER modelling, and the relational model.
2. Learn normalization techniques, functional dependencies, and relational database design principles.
3. Implement SQL for data definition, data manipulation, querying, and transaction management.
4. Design and implement stored procedures, triggers, cursors, and views using PL/SQL.
5. Evaluate concurrency control protocols, database security mechanisms, and backup and recovery strategies.

Expected Course Outcomes:

On the successful completion of the course, student will be able to:

1	Explain the fundamental concepts of database management systems, ER modelling, and relational model components including keys, constraints, and relational algebra.	K1, K2
2	Apply normalization techniques up to BCNF and design well-structured, lossless, and dependency-preserving relational schemas.	K2, K3
3	Write SQL queries for DDL, DML, DCL, and TCL operations including joins, subqueries, views, and aggregate functions.	K3, K4
4	Develop PL/SQL programs including stored procedures, functions, explicit cursors, and row-level and statement-level triggers for database automation.	K4, K5
5	Evaluate concurrency control protocols (2PL, MVCC), database security mechanisms, and backup and recovery strategies.	K5, K6

K1 - Remember; K2 - Understand; K3 - Apply; K4 - Analyze; K5 - Evaluate; K6 - Create

Unit:1	**DATABASE SYSTEM CONCEPTS AND ER MODELLING**	**15 hours**

Database system architecture; advantages of DBMS over file systems. Data models: hierarchical, network, and relational. Entity-Relationship (ER) model: entities, attributes, relationships, ER diagrams. Extended ER features: specialization, generalization, and aggregation. Mapping ER diagrams to relational schemas.

Unit:2	**RELATIONAL MODEL AND NORMALIZATION**	**15 hours**

Relational algebra: select, project, join, union, intersection, and difference. Relational calculus: tuple and domain relational calculus. Functional dependencies, Armstrong's axioms, closure of attributes. Normal forms: 1NF, 2NF, 3NF, BCNF. Multivalued dependencies and 4NF. Lossless-join and dependency-preserving decomposition.

Unit:3	**STRUCTURED QUERY LANGUAGE (SQL)**	**15 hours**

DDL: CREATE, ALTER, DROP, TRUNCATE. DML: INSERT, UPDATE, DELETE. DQL: SELECT with WHERE, ORDER BY, GROUP BY, HAVING. Joins: inner, outer (left, right,

full), cross, and self-joins. Subqueries and correlated subqueries. Aggregate functions: COUNT, SUM, AVG, MAX, MIN. Views, indexes, sequences, and synonyms.		
Unit:4	**PL/SQL PROGRAMMING PL/SQL**	**15 hours**
block structure: DECLARE, BEGIN, EXCEPTION. Variables, constants, composite data types. Control structures: IF-THEN-ELSE, CASE, FOR/WHILE loops. Cursors: implicit and explicit cursors, cursor FOR loops. Stored procedures and functions. Packages. Triggers: BEFORE and AFTER triggers, row-level and statement-level triggers, INSTEAD OF triggers.		
Unit:5	**TRANSACTION MANAGEMENT, SECURITY AND RECOVERY**	**15 hours**
Transaction properties: ACID. Concurrency problems: dirty read, lost update, unrepeatable read, phantom read. Concurrency control: locking protocols – Two-Phase Locking (2PL), Rigorous 2PL, Multi-version Concurrency Control (MVCC). Deadlock detection and prevention. Database security: authentication, authorization, SQL injection prevention. Backup and recovery: log-based recovery, checkpoints, shadow paging.		

Mapping of Course Outcomes (COs) with Program Outcomes (POs):

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	2	2	2	1	2
CO2	3	3	2	2	1	2
CO3	3	3	3	2	2	3
CO4	2	3	3	2	2	3
CO5	3	3	3	2	2	3

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Mapping of Course Outcomes (COs) with Program Specific Outcomes (PSOs):

CO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	2	3	2	2
CO2	3	2	3	2	2
CO3	3	3	3	2	2
CO4	3	3	3	2	2
CO5	3	3	3	2	2

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Text Books:

- Abraham Silberschatz, Henry F. Korth, S. Sudarshan, 'Database System Concepts', McGraw-Hill. (Covers Units: I - V) https://books.google.com/books?

q=Database+System+Concepts+Abraham+Silberschatz%2C+Henry+F.+Korth%2C+S.+Sudarshan

- Ramez Elmasri, Shamkant B. Navathe, 'Fundamentals of Database Systems', Pearson Education. (Covers Units: I - V) https://books.google.com/books?q=Fundamentals+of+Database+Systems+Ramez+Elmasri%2C+Shamkant+B.+Navathe

Reference Books:

- Raghu Ramakrishnan, Johannes Gehrke, 'Database Management Systems', McGraw-Hill. https://books.google.com/books?q=Database+Management+Systems+Raghu+Ramakrishnan%2C+Johannes+Gehrke
- C.J. Date, 'An Introduction to Database Systems', Addison-Wesley. https://books.google.com/books?q=An+Introduction+to+Database+Systems+C.J.+Date
- Ivan Bayross, 'SQL, PL/SQL: The Programming Language of Oracle', BPB Publications. https://books.google.com/books?q=SQL%2C+PL%2FSQL%3A+The+Programming+Language+of+Oracle+Ivan+Bayross

Free Online Study Resources:

- Stanford DB Course – Free SQL Exercises – https://www.db-class.org/
- W3Schools SQL Tutorial (Free) – https://www.w3schools.com/sql/
- Oracle Live SQL (Free Cloud Practice) – https://livesql.oracle.com/

SEMESTER – I
CORE – III - DISCRETE MATHEMATICS

Course Objectives:

The main objectives of this course are to:

1. Comprehend propositional and predicate calculus, normal forms, and inference rules.
2. Analyze algebraic structures including groups, rings, fields, and their applications to coding theory.
3. Evaluate combinatorial problems using permutations, combinations, and recurrence relations.
4. Apply graph theory concepts including Eulerian and Hamiltonian paths, planar graphs, and network flows.
5. Formulate matrix operations, eigenvalue decomposition, SVD, and discrete probability for data science.

Expected Course Outcomes:

On the successful completion of the course, student will be able to:

1	Apply propositional and predicate calculus including well-formed formulas, tautologies, normal forms, and quantifiers to construct valid logical arguments.	K1, K2
2	Analyse algebraic structures such as groups, rings, fields, and their homomorphisms with applications to coding theory and cryptographic key-spaces.	K2, K3
3	Solve combinatorial problems using permutations, combinations, pigeonhole principle, inclusion-exclusion, and recurrence relation techniques.	K3, K4
4	Apply graph theory concepts including Eulerian and Hamiltonian paths, planar graphs, graph colouring, and Max-Flow Min-Cut network optimisation.	K4, K5
5	Evaluate matrix operations, eigenvalue decomposition, singular value decomposition (SVD), and discrete probability for computational data science applications.	K5, K6

K1 - Remember; K2 - Understand; K3 - Apply; K4 - Analyze; K5 - Evaluate; K6 - Create

Unit:1	**PROPOSITIONAL AND PREDICATE CALCULUS**	**15 hours**

Well-formed formulas, Truth tables, Tautologies, Normal Forms (PDNF and PCNF). Predicate Logic: Universal and Existential Quantifiers, Rules of Inference, Prenex Normal Forms, Skolemization techniques for logic programming parsing.

Unit:2	**ALGEBRAIC STRUCTURES AND CODING THEORY**	**15 hours**

Monoids, Semigroups, Groups, Subgroups, Cyclic groups, Cosets, Lagrange's Theorem. Group Homomorphisms, Isomorphisms, Rings, Integral Domains, and Fields. Applications of group structures to error-correcting block codes and cryptographic key-spaces.

Unit:3	**COMBINATORICS AND ADVANCED RECURRENCE RELATIONS**	**15 hours**

Permutations, Combinations, Pigeonhole Principle, Inclusion-Exclusion Principle. Recurrence Relations: Solving homogeneous and non-homogeneous recurrence relations using characteristic roots and particular solutions. Generating Functions: Coefficient calculations for complex integer sequences.		
Unit:4	**GRAPH THEORY AND NETWORK OPTIMIZATION FLOWS**	**15 hours**
Structural properties: Eulerian and Hamiltonian paths/circuits, Planar graphs, Euler's formula, Graph Coloring algorithms, Chromatic numbers. Network Flows: Max-Flow Min-Cut Theorem, Ford-Fulkerson algorithm optimization modeling.		
Unit:5	**MATRIX ALGEBRA AND COMPUTATIONAL STATISTICS**	**15 hours**
Vector Spaces, Linear Independence, Bases. Matrix Transformations: Characteristic equations, Eigenvalues and Eigenvectors, Diagonalization. Singular Value Decomposition (SVD). Discrete		

Mapping of Course Outcomes (COs) with Program Outcomes (POs):

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	2	2	2	1	2
CO2	3	2	2	2	1	2
CO3	3	2	2	2	1	2
CO4	3	2	2	2	1	2
CO5	3	2	2	2	1	2

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Mapping of Course Outcomes (COs) with Program Specific Outcomes (PSOs):

CO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	2	2	3	2	1
CO2	2	2	3	2	1
CO3	2	2	3	2	1
CO4	2	2	3	2	1
CO5	2	2	3	2	1

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Text Books:

- Dr. N. Chandrasekaran, Dr. M. Umapparvathi, 'Discrete Mathematics', PHI Learning Pvt. Ltd. (Covers Units: I - V) https://books.google.com/books?q=Discrete+Mathematics+Dr.+N.+Chandrasekaran%2C+Dr.+M.+Umapparvathi

- Kenneth H. Rosen, 'Discrete Mathematics and its Applications', McGraw-Hill. (Covers Units: I - V) https://books.google.com/books?q=Discrete+Mathematics+and+its+Applications+Kenneth+H.+Rosen

Reference Books:

- J.P. Tremblay, R. Manohar, 'Discrete Mathematical Structures with Applications to Computer Science', Tata McGraw-Hill. https://books.google.com/books?q=Discrete+Mathematical+Structures+with+Applications+to+Computer+Science+J.P.+Tremblay%2C+R.+Manohar
- Gilbert Strang, 'Linear Algebra and Learning from Data', Wellesley-Cambridge Press. https://books.google.com/books?q=Linear+Algebra+and+Learning+from+Data+Gilbert+Strang
- Narsingh Deo, 'Graph Theory with Applications to Engineering and Computer Science', PHI Learning. https://books.google.com/books?q=Graph+Theory+with+Applications+to+Engineering+and+Computer+Science+Narsingh+Deo

Free Online Study Resources:

- Mathematics for Computer Science – MIT OCW (Free PDF) – https://ocw.mit.edu/courses/6-042j-mathematics-for-computer-science-fall-2010/
- Discrete Mathematics: An Open Introduction – Oscar Levin (Free) – https://discrete.openmathbooks.org/

SEMESTER – I
CORE – IV - ADVANCED COMPUTER NETWORKS

Course Objectives:

The main objectives of this course are to:

1. Understand the OSI and TCP/IP reference models, physical and data link layer protocols.
2. Analyze network layer protocols including IP addressing, subnetting, and routing algorithms.
3. Evaluate transport layer services, TCP and UDP protocols, and congestion control mechanisms.
4. Implement application layer protocols including HTTP, FTP, DNS, SMTP, and DHCP.
5. Formulate network security concepts including cryptography, firewalls, VPNs, and intrusion detection systems.

Expected Course Outcomes:

On the successful completion of the course, student will be able to:

1	Explain the OSI and TCP/IP reference models and describe physical and data link layer protocols including HDLC, PPP, Ethernet, and IEEE 802.11 wireless standards.	K1, K2
2	Analyse IPv4 and IPv6 addressing, subnetting, CIDR, ARP, ICMP, and routing algorithms including RIP, OSPF, and BGP.	K2, K3
3	Apply transport layer protocols (TCP and UDP), connection establishment (3-way handshake), sliding window, and congestion control mechanisms.	K3, K4
4	Implement application layer protocols including HTTP/HTTPS, FTP, DNS, SMTP/IMAP, and DHCP for distributed network services.	K4, K5
5	Evaluate network security mechanisms including symmetric/asymmetric cryptography, digital signatures, SSL/TLS, VPNs, firewalls, and IDS/IPS.	K5, K6

K1 - Remember; K2 - Understand; K3 - Apply; K4 - Analyze; K5 - Evaluate; K6 - Create

Unit:1	**INTRODUCTION AND DATA LINK LAYER**	**15 hours**

Network topologies, transmission media. OSI reference model and TCP/IP model comparison. Data Link Layer: framing, error detection and correction (CRC, Hamming code), flow control (Stop-and-Wait, Sliding Window). Protocols: HDLC, PPP. MAC sublayer: CSMA/CD, CSMA/CA. Ethernet (IEEE 802.3), IEEE 802.11 Wi-Fi standards.

Unit:2	**NETWORK LAYER IP**	**15 hours**

addressing: IPv4, classful and classless addressing, subnetting, CIDR, NAT. IPv6 addressing and transition mechanisms. ARP, ICMP, IGMP. Routing algorithms: Dijkstra's shortest path, Bellman-Ford, Link State, Distance Vector. Routing protocols: RIP, OSPF, BGP. Congestion control principles and QoS.

Unit:3	**TRANSPORT LAYER**	**15 hours**

Transport layer services: multiplexing, demultiplexing. Connection-oriented vs. connectionless services. UDP: datagram structure and applications. TCP: segment structure, connection establishment (3-way handshake), connection termination. Flow control: sliding window protocol. Congestion control: slow start, congestion avoidance, fast retransmit, fast recovery.		
Unit:4	**APPLICATION LAYER**	**15 hours**
Client-server and peer-to-peer models. HTTP and HTTPS: request-response model, persistent connections, cookies, HTTP/2. FTP and SFTP. DNS: hierarchy, resource records, iterative and recursive resolution. Electronic mail: SMTP, POP3, IMAP, MIME. DHCP. Network management: SNMP.		
Unit:5	**NETWORK SECURITY**	**15 hours**
Security goals: confidentiality, integrity, availability. Symmetric key cryptography: DES, 3DES, AES. Asymmetric key cryptography: RSA. Digital signatures and certificates, PKI. Hash functions: MD5, SHA-1, SHA-256. SSL/TLS protocol. Firewalls: packet filtering, stateful inspection, proxy firewalls. VPN technologies: IPSec, SSL VPN. Intrusion Detection and Prevention Systems (IDS/IPS).		

Mapping of Course Outcomes (COs) with Program Outcomes (POs):

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	2	3	2	2	2
CO2	3	3	3	2	2	2
CO3	3	3	3	2	2	2
CO4	2	3	3	2	2	2
CO5	3	3	3	2	3	2

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Mapping of Course Outcomes (COs) with Program Specific Outcomes (PSOs):

CO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	2	3	2	2
CO2	3	2	3	2	2
CO3	3	2	3	2	2
CO4	3	2	3	2	2
CO5	3	2	3	2	3

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Text Books:

- Andrew S. Tanenbaum, David J. Wetherall, 'Computer Networks', 5th Ed., Pearson Education. (Covers Units: I - V) https://books.google.com/books?q=Computer+Networks+Andrew+S.+Tanenbaum%2C+David+J.+Wetherall
- William Stallings, 'Data and Computer Communications', 10th Ed., Pearson Education. (Covers Units: I - V) https://books.google.com/books?q=Data+and+Computer+Communications+William+Stallings

Reference Books:

- Forouzan B.A., 'Data Communications and Networking', 5th Ed., Tata McGraw-Hill. https://books.google.com/books?q=Data+Communications+and+Networking+Forouzan+B.A.
- James F. Kurose, Keith W. Ross, 'Computer Networking: A Top-Down Approach', Pearson. https://books.google.com/books?q=Computer+Networking%3A+A+Top-Down+Approach+James+F.+Kurose%2C+Keith+W.+Ross
- Charlie Kaufman, Radia Perlman, 'Network Security: Private Communication in a Public World', Pearson. https://books.google.com/books?q=Network+Security%3A+Private+Communication+in+a+Public+World+Charlie+Kaufman%2C+Radia+Perlman

Free Online Study Resources:

- Computer Networking Companion Website (Free) – https://gaia.cs.umass.edu/kurose_ross/
- Cisco Networking Academy – Networking Basics (Free) – https://skillsforall.com/
- NPTEL – Computer Networks (Free) – https://nptel.ac.in/courses/106105082

SEMESTER – I
ELECTIVE – I - DATA ANALYTICS

Course Objectives:

The main objectives of this course are to:

1. Comprehend the fundamentals of data analytics, statistical measures, and types of analytics.
2. Analyze data using descriptive statistics, probability distributions, and hypothesis testing.
3. Evaluate data preprocessing techniques including handling missing values, outliers, and feature engineering.
4. Implement data visualization techniques using charts, plots, and interactive dashboards.
5. Formulate predictive models using regression, classification, and clustering techniques.

Expected Course Outcomes:

On the successful completion of the course, student will be able to:

1	Explain fundamentals of data analytics, types of analytics (descriptive, diagnostic, predictive, prescriptive), data types, and statistical measures including mean, median, mode, variance, and standard deviation.	K1, K2
2	Apply probability distributions, hypothesis testing (z-test, t-test, chi-square, ANOVA), correlation, and regression analysis to draw inferences from data.	K2, K3
3	Implement data preprocessing techniques to handle missing values, outliers, feature scaling (normalization, standardization), and encoding for analytics pipelines.	K3, K4
4	Create effective data visualizations using bar charts, histograms, box plots, scatter plots, heat maps, and interactive dashboards using Matplotlib, Seaborn, and Plotly.	K4, K5
5	Develop and evaluate predictive models using linear regression, logistic regression, decision trees, and k-means clustering with appropriate performance metrics.	K5, K6

K1 - Remember; K2 - Understand; K3 - Apply; K4 - Analyze; K5 - Evaluate; K6 - Create

Unit:1	**INTRODUCTION TO DATA ANALYTICS**	**15 hours**

Definition and scope of data analytics. Types: descriptive, diagnostic, predictive, and prescriptive analytics. Data types: structured, semi-structured, and unstructured. Statistical measures: mean, median, mode, variance, standard deviation, skewness, kurtosis. Data collection and sampling methods. Roles in data analytics: data analyst, data scientist, data engineer.

Unit:2	**STATISTICAL FOUNDATIONS**	**15 hours**

Probability theory: conditional probability, Bayes' theorem. Probability distributions: Binomial, Poisson, Normal distribution. Central Limit Theorem. Hypothesis testing: z-test, t-test, chi-square test, ANOVA, p-values, confidence intervals. Correlation analysis: Pearson, Spearman, Kendall. Introduction to linear and multiple regression.

Unit:3	**DATA PREPROCESSING AND FEATURE**	**15 hours**

	ENGINEERING	
Data quality issues: missing values, noisy data, inconsistencies. Handling missing values: mean/median/mode imputation, KNN imputation. Outlier detection and treatment: IQR method, Z-score. Data transformation: normalization (Min-Max), standardization (Z-score), log transformation. Feature selection: filter, wrapper, embedded methods. Feature		
Unit:4	**DATA VISUALIZATION**	**15 hours**
Principles of effective data visualization. Chart types: bar charts, pie charts, histograms, line graphs, area charts. Advanced visualizations: box plots, violin plots, scatter plots, bubble charts, heat maps, treemaps. Tools: Matplotlib, Seaborn, Plotly. Dashboard design: layout principles, interactivity, storytelling with data. Introduction to Tableau and Power BI.		
Unit:5	**PREDICTIVE ANALYTICS**	**15 hours**
Regression analysis: simple and multiple linear regression, polynomial regression, regularization (Ridge, Lasso). Classification: logistic regression, decision trees, k-nearest neighbors. Clustering: k-means (Elbow method), hierarchical, DBSCAN. Model evaluation metrics: accuracy, precision, recall, F1-score, ROC-AUC for classification; MSE, RMSE, MAE, R^2 for regression. Cross-validation: k-fold, stratified k-fold.		

Mapping of Course Outcomes (COs) with Program Outcomes (POs):

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	2	3	3	1	2
CO2	3	2	3	3	1	2
CO3	3	2	3	3	1	2
CO4	2	2	3	3	1	2
CO5	3	3	3	3	1	3

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Mapping of Course Outcomes (COs) with Program Specific Outcomes (PSOs):

CO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	2	2	3	3	2
CO2	2	2	3	3	2
CO3	3	2	3	3	2
CO4	3	2	3	3	2
CO5	3	3	3	3	2

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Text Books:

- Wes McKinney, 'Python for Data Analysis', 3rd Ed., O'Reilly Media. (Covers Units: I - V) https://books.google.com/books?q=Python+for+Data+Analysis+Wes+McKinney
- Jake VanderPlas, 'Python Data Science Handbook', O'Reilly Media. (Covers Units: I - V) https://books.google.com/books?q=Python+Data+Science+Handbook+Jake+VanderPlas

Reference Books:

- Gareth James et al., 'An Introduction to Statistical Learning', Springer. https://books.google.com/books?q=An+Introduction+to+Statistical+Learning+Gareth+James+et+al.
- Roger D. Peng, 'Exploratory Data Analysis with R', Lean Publishing. https://books.google.com/books?q=Exploratory+Data+Analysis+with+R+Roger+D.+Peng
- Edward R. Tufte, 'The Visual Display of Quantitative Information', Graphics Press. https://books.google.com/books?q=The+Visual+Display+of+Quantitative+Information+Edward+R.+Tufte

Free Online Study Resources:

- Python Data Science Handbook (Free) – https://jakevdp.github.io/PythonDataScienceHandbook/
- Kaggle Learn – Data Analysis (Free) – https://www.kaggle.com/learn
- NPTEL – Business Analytics and Data Mining (Free) – https://nptel.ac.in/

SEMESTER – I
CORE – V – LAB I - JAVA PROGRAMMING LAB

Learning Objective (LO):
To develop Java applications using core OOP concepts, JDBC, Servlets, JSP, and Spring Boot frameworks.

List of Programs:

01. Core Java Class Implementation: Create classes and objects demonstrating constructors, inheritance, method overloading, and method overriding.
02. Exception Handling Application: Implement exception handling using try-catch-finally blocks and custom exceptions.
03. Concurrent Threads Execution: Implement a multi-threaded application where multiple threads execute concurrently using synchronization.
04. Database Record Insertion: Use JDBC to connect to a MySQL database and insert a new student record.
05. Database Row Selection: Use JDBC to retrieve and display records from an employee database table.
06. React-Based Student Information Form: Create a React application that accepts student details through a form and displays the submitted information using reusable components.
07. Basic User Servlet: Create an HTML form that sends user data to a Java Servlet, which processes the request and displays a welcome message.
08. Servlet Session Tracking: Write a Java Servlet program that tracks and displays user session visits using HttpSession and Cookies.
09. JSP Date and Time Display: Create a JSP page that displays the current server date and time using scripting elements and Expression Language.
10. Spring Boot and Hibernate CRUD Application: Develop a Spring Boot application integrated with Hibernate and MySQL to perform CRUD operations on employee records.

Expected Course Outcomes:
On the successful completion of this practical course, student will be able to:

1	Create Java classes and objects demonstrating constructors, inheritance, method overloading, and exception handling.	K1, K2
2	Develop multithreaded Java applications and perform CRUD database operations using JDBC connectivity.	K2, K3
3	Build server-side web applications using Java Servlets with session management and cookies.	K3, K4
4	Design dynamic web pages using JSP scripting elements and Expression Language.	K4, K5
5	Develop enterprise applications using Spring Boot, Hibernate, and Maven	K5, K6

	for CRUD operations.	
K1 - Remember; K2 - Understand; K3 - Apply; K4 - Analyze; K5 - Evaluate; K6 - Create		

Mapping of Course Outcomes (COs) with Program Outcomes (POs):

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	2	2	1	1	2
CO2	2	3	2	1	1	2
CO3	3	3	2	2	1	2
CO4	2	2	3	2	1	3
CO5	3	3	3	2	2	3

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Mapping of Course Outcomes (COs) with Program Specific Outcomes (PSOs):

CO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	2	1	1	2
CO2	2	3	1	1	2
CO3	3	3	2	1	2
CO4	2	2	2	2	3
CO5	3	3	2	2	3

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

SEMESTER – I
CORE – VI – LAB II - RDBMS LAB

Learning Objective (LO):
To implement SQL, PL/SQL programs, and database design techniques for real-world relational database applications.

List of Programs:

01. Database and Table Creation: Write SQL statements to create a database and define tables with appropriate constraints (PRIMARY KEY, FOREIGN KEY, NOT NULL, UNIQUE, CHECK) for a Student Management System.
02. Insert and Retrieve Records: Write SQL statements to insert multiple records and retrieve data using SELECT with WHERE, ORDER BY, and DISTINCT clauses.
03. SQL Aggregate Functions: Write SQL queries using COUNT, SUM, AVG, MAX, and MIN functions with GROUP BY and HAVING clauses on an employee database.
04. SQL Joins: Write SQL queries demonstrating INNER JOIN, LEFT OUTER JOIN, RIGHT OUTER JOIN, and FULL OUTER JOIN between Student and Department tables.
05. Subqueries and Views: Write SQL queries using subqueries (correlated and non-correlated) and create a view to display students with marks above average.
06. PL/SQL Stored Procedure: Write a PL/SQL stored procedure to calculate and update the bonus amount for employees based on their salary grade.
07. PL/SQL Function and Cursors: Write a PL/SQL function using explicit cursors to retrieve and display department-wise employee count.
08. Trigger Implementation: Create a BEFORE INSERT trigger to validate and automatically set the enrollment date when a new student record is added.
09. Transaction Management: Write SQL statements demonstrating COMMIT, ROLLBACK, and SAVEPOINT to manage bank transfer transactions with error handling.
10. Indexing and Query Optimization: Create appropriate indexes on a large product table and compare query execution plans before and after indexing.

Expected Course Outcomes:
On the successful completion of this practical course, student will be able to:

1	Create database schemas with appropriate constraints using SQL DDL commands and retrieve data using DQL.	K1, K2
2	Write SQL queries using aggregate functions, GROUP BY, HAVING, and different types of joins for data retrieval.	K2, K3
3	Develop subqueries, views, and correlated queries for complex data retrieval requirements.	K3, K4
4	Implement PL/SQL stored procedures, functions, cursors, and triggers for database automation.	K4, K5

5	Apply transaction management, indexing, and query optimization techniques for database performance.	K5, K6
K1 - Remember; K2 - Understand; K3 - Apply; K4 - Analyze; K5 - Evaluate; K6 - Create		

Mapping of Course Outcomes (COs) with Program Outcomes (POs):

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	2	2	1	1	2
CO2	2	3	2	1	1	2
CO3	3	3	2	2	1	2
CO4	2	2	3	2	1	3
CO5	3	3	3	2	2	3

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Mapping of Course Outcomes (COs) with Program Specific Outcomes (PSOs):

CO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	2	1	1	2
CO2	2	3	1	1	2
CO3	3	3	2	1	2
CO4	2	2	2	2	3
CO5	3	3	2	2	3

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

SEMESTER – II
CORE – VII - DATA MINING TECHNIQUES

Course Objectives:

The main objectives of this course are to:

1. Comprehend the data mining process, KDD framework, and data warehouse architectures.
2. Analyze data preprocessing techniques including data cleaning, integration, reduction, and transformation.
3. Evaluate frequent pattern mining algorithms including Apriori and FP-Growth.
4. Implement classification algorithms including decision trees, Naive Bayes, and support vector machines.
5. Formulate clustering techniques including k-means, hierarchical, and density-based methods.

Expected Course Outcomes:

On the successful completion of the course, student will be able to:

1	Explain the data mining process, KDD framework, OLAP operations, and data warehouse architectures including star and snowflake schemas.	K1, K2
2	Apply data preprocessing techniques including data cleaning, integration, transformation, and dimensionality reduction for mining readiness.	K2, K3
3	Implement association rule mining using Apriori and FP-Growth algorithms to discover frequent patterns and interesting associations.	K3, K4
4	Design classification models using decision trees (ID3, C4.5), Naive Bayes classifier, support vector machines, and evaluate using precision, recall, and F1-score.	K4, K5
5	Apply clustering techniques including k-means partitioning, agglomerative hierarchical clustering, and DBSCAN density-based methods with validity measures.	K5,K6

K1 - Remember; K2 - Understand; K3 - Apply; K4 - Analyze; K5 - Evaluate; K6 - Create

Unit:1	**DATA MINING FUNDAMENTALS AND DATA WAREHOUSING**	**15 hours**

Data mining definition, tasks, and applications. KDD process overview. Data warehouse: architecture, OLAP vs. OLTP, data cube, OLAP operations (Roll-up, Drill-down, Slice, Dice, Pivot). Multidimensional data model. Star schema, snowflake schema. ETL process overview.

Unit:2	**DATA PREPROCESSING**	**15 hours**

Data quality: accuracy, completeness, consistency, timeliness. Data cleaning: handling missing values, noisy data (binning, regression, clustering), and inconsistencies. Data integration: schema integration, entity identification, detecting and resolving conflicts. Data transformation: normalization (min-max, z-score), aggregation, attribute construction. Data reduction: PCA, feature selection, numerosity reduction.

Unit:3	**ASSOCIATION RULE MINING**	**15 hours**

Market basket analysis. Frequent itemsets, support, confidence, and lift. Apriori algorithm: candidate generation and pruning strategies. FP-Growth algorithm: FP-tree construction and conditional FP-tree mining. Mining multilevel and multidimensional association rules. Correlation analysis: chi-square test, lift, all-confidence		
Unit:4	**CLASSIFICATION**	**15 hours**
Classification vs. prediction. Decision tree induction: attribute selection measures (Information Gain, Gain Ratio, Gini Index), tree pruning (pre-pruning, post-pruning). Naive Bayes classifier: Bayesian theorem, class conditional independence. Rule-based classification. Support Vector Machines: hyperplane equations, maximal margin, kernel trick. Backpropagation neural network. Model evaluation: accuracy, precision, recall, F1-score, ROC curve, k-fold cross-validation.		
Unit:5	**CLUSTERING**	**15 hours**
Clustering requirements and challenges. Cluster analysis types. Partitioning methods: k-means (Lloyd's algorithm) and k-medoids (PAM). Hierarchical methods: agglomerative and divisive clustering, dendrograms, linkage methods (single, complete, average). Density-based clustering: DBSCAN (core points, border points, noise), OPTICS. Cluster validity measures: cohesion, separation, silhouette coefficient, Dunn index.		

Mapping of Course Outcomes (COs) with Program Outcomes (POs):

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	2	3	3	1	2
CO2	3	2	3	3	1	2
CO3	3	2	3	3	1	2
CO4	3	3	3	3	1	2
CO5	2	3	3	3	1	3

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Mapping of Course Outcomes (COs) with Program Specific Outcomes (PSOs):

CO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	2	2	3	3	2
CO2	2	2	3	3	2
CO3	3	2	3	3	2
CO4	3	3	3	3	2
CO5	3	3	3	3	2

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Text Books:

- Jiawei Han, Micheline Kamber, Jian Pei, 'Data Mining: Concepts and Techniques', 3rd Ed., Morgan Kaufmann. (Covers Units: I - V) https://books.google.com/books?q=Data+Mining%3A+Concepts+and+Techniques+Jiawei+Han%2C+Micheline+Kamber%2C+Jian+Pei
- P. N. Tan, M. Steinbach, V. Kumar, 'Introduction to Data Mining', Pearson Education. (Covers Units: I - V) https://books.google.com/books?q=Introduction+to+Data+Mining+P.+N.+Tan%2C+M.+Steinbach%2C+V.+Kumar

Reference Books:

- I. H. Witten, E. Frank, 'Data Mining: Practical Machine Learning Tools and Techniques', Morgan Kaufmann. https://books.google.com/books?q=Data+Mining%3A+Practical+Machine+Learning+Tools+and+Techniques+I.+H.+Witten%2C+E.+Frank
- M. H. Dunham, 'Data Mining: Introductory and Advanced Topics', Pearson Education. https://books.google.com/books?q=Data+Mining%3A+Introductory+and+Advanced+Topics+M.+H.+Dunham
- Alex Berson, Stephen J. Smith, 'Data Warehousing, Data Mining, and OLAP', McGraw-Hill. https://books.google.com/books?q=Data+Warehousing%2C+Data+Mining%2C+and+OLAP+Alex+Berson%2C+Stephen+J.+Smith

Free Online Study Resources:

- Weka Data Mining Tutorial (Free) – https://www.cs.waikato.ac.nz/ml/weka/
- NPTEL – Data Mining (Free) – https://nptel.ac.in/courses/106106077
- scikit-learn Documentation (Free) – https://scikit-learn.org/stable/

SEMESTER – II
CORE – VIII - PYTHON PROGRAMMING

Course Objectives:

The main objectives of this course are to:

1. Understand Python best practices, dictionary usage, string formatting, and common coding conventions.
2. Analyze core programming concepts including mutability, iterators, and functional programming constructs.
3. Design effective functions using lambda expressions, recursion, and modular decomposition.
4. Organize Python projects using Git for version control and remote repository management on GitHub.
5. Implement object-oriented programming principles including classes, inheritance, and multiple inheritance.

Expected Course Outcomes:

On the successful completion of the course, student will be able to:

1	Apply Python best practices including Pythonic coding conventions, dictionary operations, and string formatting (f-strings) to write clean and efficient programs.	K1, K2
2	Analyse core programming concepts such as mutability, iterators, higher-order functions, and functional programming constructs in Python.	K2, K3
3	Design effective functions using lambda expressions, list comprehensions, recursion, and modular decomposition for robust application development.	K3, K4
4	Organise Python projects using Git for version control, implementing branching, committing, and remote repository management on GitHub.	K4, K5
5	Implement object-oriented programming principles including classes, inheritance, polymorphism, class methods, static methods, and multiple inheritance for scalable software design.	K5, K6

K1 - Remember; K2 - Understand; K3 - Apply; K4 - Analyze; K5 - Evaluate; K6 - Create

Unit:1	**PYTHON BEST PRACTICES AND TOOLS**	**15 hours**

Writing Python Code: The Zen of Python, Significant Indentation, Commonly Misused Syntax, Formatting Strings (f-strings, format ()). Pythonic Ways to Use Dictionaries: get () and setdefault (), collections. defaultdict, using dictionaries instead of switch statements. Working with Variable Values: Chaining Assignment and Comparison Operators, Checking Whether a Variable Is One of Many Values.

Unit:2	**PYTHON PROGRAMMING CONCEPTS**	**15 hours**

Definitions: Garbage Collection, Literals, Keywords, Objects, Values, Instances and Identities, Items, Mutable and Immutable types. Indexes, Keys and Hashes. Containers, Sequences,

Mapping, and Set Types. Modules and Packages: importing modules, __init__.py, relative and absolute imports. Callables and First-Class Objects. Iterators and generators.		
Unit:3	**WRITING EFFECTIVE FUNCTIONS**	**15 hours**
Function Names, Function Parameters and Arguments (positional, keyword, *args, **kwargs, default values). Functional Programming: Side Effects, Higher-Order Functions, Lambda Functions. Mapping and Filtering with List Comprehensions, Dictionary Comprehensions, Generator		
Unit:4	**ORGANIZING CODE PROJECTS WITH GIT**	**15 hours**
Installing Git, The Git Workflow. Creating a Git Repository: Adding Files for Git to Track, gitignore Files, Committing Changes, Viewing Commit History, Deleting and Renaming Files. Branching: creating, switching, merging branches. Resolving merge conflicts. GitHub: remote repositories, git push, git pull, git clone, pull requests. Collaborative workflows.		
Unit:5	**OBJECT-ORIENTED PYTHON**	**15 hours**
Object-Oriented Programming and Classes: Creating Objects from Classes, __init__ method, Instance Attributes. The type () Function, isinstance () and issubclass (). How Inheritance Works, Method Resolution Order (MRO). Class Methods (@classmethod), Static Methods (@staticmethod), Class Attributes. Dunder (magic) methods: __str__, __repr__, __len__, __eq__. Multiple Inheritance and Mixins.		

Mapping of Course Outcomes (COs) with Program Outcomes (POs):

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	3	3	2	2	3
CO2	3	2	3	2	2	3
CO3	3	3	3	2	2	3
CO4	2	2	3	1	1	2
CO5	3	3	3	2	2	3

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Mapping of Course Outcomes (COs) with Program Specific Outcomes (PSOs):

CO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	3	2	2	2
CO2	2	3	2	2	2
CO3	2	3	2	2	2
CO4	2	3	1	2	2
CO5	3	3	2	2	3

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Text Books:

- Al Sweigart, 'Beyond the Basic Stuff with Python', No Starch Press, 2021. (Covers Units: I - V) https://books.google.com/books?q=Beyond+the+Basic+Stuff+with+Python+Al+Sweigart
- Al Sweigart, 'Automate the Boring Stuff with Python', 2nd Ed., No Starch Press, 2019. (Covers Units: I - V) https://books.google.com/books?q=Automate+the+Boring+Stuff+with+Python+Al+Sweigart

Reference Books:

- Luciano Ramalho, 'Fluent Python', 2nd Ed., O'Reilly Media, 2022. https://books.google.com/books?q=Fluent+Python+Luciano+Ramalho
- Eric Matthes, 'Python Crash Course', 3rd Ed., No Starch Press, 2023. https://books.google.com/books?q=Python+Crash+Course+Eric+Matthes
- Mark Lutz, 'Learning Python', 5th Ed., O'Reilly Media. https://books.google.com/books?q=Learning+Python+Mark+Lutz

Free Online Study Resources:

- Automate the Boring Stuff with Python (Free E-Book) – https://automatetheboringstuff.com/
- Think Python 2nd Ed. – Allen B. Downey (Free PDF) – https://greenteapress.com/wp/think-python-2e/
- Python for Everybody – Dr. Chuck Severance (Free) – https://www.py4e.com/book

SEMESTER – II
CORE – IX - ADVANCED OPERATING SYSTEM

Course Objectives:

The main objectives of this course are to:

1. Understand operating system fundamentals, process management, threads, and inter-process communication.
2. Analyze CPU scheduling algorithms and evaluate their performance parameters.
3. Evaluate deadlock conditions, detection algorithms, prevention strategies, and the Banker's algorithm.
4. Implement memory management techniques including paging, segmentation, virtual memory, and page replacement.
5. Formulate file system structures, I/O scheduling strategies, and distributed operating system concepts.

Expected Course Outcomes:

On the successful completion of the course, student will be able to:

1	Explain the structure of operating systems, process life cycle, process states, thread models, and implement inter-process communication mechanisms including semaphores and monitors.	K1, K2
2	Analyse and compare CPU scheduling algorithms including FCFS, SJF, SRTF, Priority, Round Robin, and multilevel feedback queue scheduling with performance metrics.	K2, K3
3	Apply deadlock detection, prevention, avoidance (Banker's algorithm), and recovery strategies in resource allocation scenarios.	K3, K4
4	Implement memory management techniques including contiguous allocation, paging (TLB), segmentation, page replacement algorithms (FIFO, Optimal, LRU), and virtual memory.	K4, K5
5	Evaluate file system structures, disk scheduling algorithms (SSTF, SCAN, C-SCAN), RAID levels, and distributed operating system concepts.	K5, K6

K1 - Remember; K2 - Understand; K3 - Apply; K4 - Analyze; K5 - Evaluate; K6 - Create

Unit:1	**PROCESS MANAGEMENT AND SYNCHRONIZATION**	**15 hours**

OS structure and services; system calls; monolithic, layered, microkernel, and virtual machine structures. Process concept: process states, PCB, context switching. Threads: user-level and kernel-level threads, multithreading models (many-to-one, one-to-one, many-to-many). IPC: shared memory and message passing. Producer-consumer problem. Semaphores: binary and counting. Monitors and condition variables. Classic problems: dining philosophers, readers-writers problem.

Unit:2	**CPU SCHEDULING CPU**	**15 hours**

scheduling criteria: CPU utilization, throughput, turnaround time, waiting time, response time. Scheduling algorithms: First Come First Served (FCFS), Shortest Job First (SJF), Shortest

Remaining Time First (SRTF), Priority scheduling (preemptive and non-preemptive), Round Robin. Multilevel queue scheduling and multilevel feedback queue. Real-time scheduling: Rate Monotonic, Earliest Deadline First. Algorithm evaluation: deterministic modeling, queuing models, simulation.		
Unit:3	**DEADLOCK**	**15 hours**
Deadlock characterization: necessary conditions – mutual exclusion, hold and wait, no preemption, circular wait. Resource allocation graph and deadlock detection. Deadlock prevention: attacking each necessary condition. Deadlock avoidance: safe states, resource allocation graph algorithm, Banker's algorithm for multiple resource types. Deadlock detection: wait-for graph, general resource-request algorithm. Recovery from deadlock: process termination, resource preemption.		
Unit:4	**MEMORY MANAGEMENT AND VIRTUAL MEMORY**	**15 hours**
Memory allocation strategies: first fit, best fit, worst fit. Internal and external fragmentation; compaction. Contiguous memory allocation with protection. Paging: page tables, TLB, multi-level page tables, hardware support. Segmentation with paging. Virtual memory: demand paging, page fault handling, copy-on-write. Page replacement algorithms: FIFO (Belady's anomaly), Optimal, LRU (counter and stack implementations). Working set model and thrashing; frame allocation policies.		
Unit:5	**FILE SYSTEMS AND I/O MANAGEMENT**	**15 hours**
File concepts, access methods (sequential, direct), directory structures (single-level, two-level, tree, acyclic graph), file sharing and protection. File system implementation: allocation methods (contiguous, linked, indexed), free-space management (bit vector, linked list). I/O hardware: interrupts, DMA. I/O software layers: device drivers, interrupt handlers, device-independent I/O. Disk scheduling: FCFS, SSTF,		

Mapping of Course Outcomes (COs) with Program Outcomes (POs):

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	2	3	2	2	2
CO2	3	3	3	2	2	2
CO3	3	3	3	2	2	2
CO4	3	3	3	2	2	2
CO5	3	3	3	2	2	2

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Mapping of Course Outcomes (COs) with Program Specific Outcomes (PSOs):

CO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	2	3	2	2

CO2	3	2	3	2	2
CO3	3	2	3	2	2
CO4	3	3	3	2	2
CO5	3	3	3	2	2

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Text Books:

- Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, 'Operating System Concepts', 10th Ed., John Wiley & Sons. (Covers Units: I - V) https://books.google.com/books?q=Operating+System+Concepts+Abraham+Silberschatz%2C+Peter+Baer+Galvin%2C+Greg+Gagne
- Andrew S. Tanenbaum, Herbert Bos, 'Modern Operating Systems', 4th Ed., Pearson Education. (Covers Units: I - V) https://books.google.com/books?q=Modern+Operating+Systems+Andrew+S.+Tanenbaum%2C+Herbert+Bos

Reference Books:

- Dhananjay M. Dhamdhere, 'Operating Systems: A Concept Based Approach', McGraw-Hill. https://books.google.com/books?q=Operating+Systems%3A+A+Concept+Based+Approach+Dhananjay+M.+Dhamdhere
- William Stallings, 'Operating Systems: Internals and Design Principles', Pearson. https://books.google.com/books?q=Operating+Systems%3A+Internals+and+Design+Principles+William+Stallings
- Milan Milenkovic, 'Operating Systems: Concepts and Design', McGraw-Hill. https://books.google.com/books?q=Operating+Systems%3A+Concepts+and+Design+Milan+Milenkovic

Free Online Study Resources:

- Operating Systems: Three Easy Pieces (Free E-Book) – https://ostep.org/
- MIT 6.S081 OS Engineering (Free) – https://pdos.csail.mit.edu/6.S081/
- NPTEL – Operating Systems (Free) – https://nptel.ac.in/courses/106105044

SEMESTER – II
ELECTIVE – II - ADVANCED WEB UI FRAMEWORKS (ANGULAR & REACT ENTERPRISE ECOSYSTEM)

Course Objectives:

The main objectives of this course are to:

1. Comprehend ES6+ execution paradigms, TypeScript type-safety layers, and SPA workflows.
2. Analyze Angular components, templates, directives, pipes, and hierarchical module organization.
3. Evaluate Angular DI injection, RxJS Observables, HTTP client modules, and router guards.
4. Implement React components, JSX, Virtual DOM, and local state tracking via Hooks.
5. Formulate Redux global state management, asynchronous operations, and production optimization strategies.

Expected Course Outcomes:

On the successful completion of the course, student will be able to:

1	Apply ES6+ features including closures, async/await, Promises, and TypeScript type safety with interfaces, generics, and enums for modern web development.	K1, K2
2	Develop Angular single-page applications using component lifecycles, data binding, structural and attribute directives, custom pipes, and module hierarchies.	K2, K3
3	Implement Angular services with dependency injection, RxJS Observable streams via HttpClient, client-side routing, child routes, and router guards.	K3, K4
4	Build React applications using functional components, JSX, Virtual DOM reconciliation, and Hooks (useState, useEffect, useMemo, useCallback, custom hooks).	K4, K5
5	Apply Redux Toolkit for global state management, Redux Thunk for async operations, and Vite for code splitting, lazy loading, and production bundling.	K5, K6

K1 - Remember; K2 - Understand; K3 - Apply; K4 - Analyze; K5 - Evaluate; K6 - Create

Unit:1	**MODERN JAVASCRIPT AND TYPESCRIPT**	**15 hours**

ES6+ execution paradigms: let/const, arrow functions, template literals, destructuring, spread/rest operators. Closures, asynchronous event loops, Promises, async/await mechanics. TypeScript type-safety layers: Interfaces, Type Aliases, Generics, Enums, Type Assertions, Union and Intersection types. TypeScript compilation: tsconfig. json, strict mode.

Unit:2	**ANGULAR COMPONENTS AND TEMPLATES**	**15 hours**

Single Page Application (SPA) workflow and Angular CLI. Component lifecycle hooks (ngOnInit, ngOnDestroy, ngOnChanges). Templates and data binding: interpolation, property binding, event binding, two-way binding (ngModel). Structural Directives: *ngIf, *ngFor, *ngSwitch. Attribute Directives: ngClass, ngStyle. Custom Directives and Pipes. Hierarchical

module organization: feature modules, shared modules.		
Unit:3	**ANGULAR SERVICES, ROUTING AND HTTP**	**15 hours**
Angular Dependency Injection (DI): token injection, providedIn options. Angular services: singleton services, service hierarchies. Reactive programming with RxJS: Observables, Subject, BehaviorSubject, operators (map, filter, switchMap, mergeMap). HttpClient: GET, POST, PUT, DELETE requests,		
Unit:4	**REACT COMPONENTS AND STATE MANAGEMENT**	**15 hours**
React philosophy and Virtual DOM reconciliation algorithm. Functional components vs. class components. JSX syntax: expressions, conditional rendering, list rendering with keys. React Hooks: useState (state management), useEffect (side effects, cleanup), useRef, useMemo (memoization), useCallback (callback memoization). Context API: createContext, useContext. Custom hooks. Component composition and lifting state up.		
Unit:5	**REDUX STATE MANAGEMENT AND APPLICATION OPTIMIZATION**	**15 hours**
Need for global state management. Redux core concepts: Actions, Reducers, Store. Redux Toolkit: createSlice, createAsyncThunk, configureStore. Redux Thunk middleware for async operations (API calls). React-Redux: useSelector, useDispatch hooks. Performance optimization: React.memo, code splitting with React.lazy, Suspense, bundle analysis. Production bundling workflows via Vite. Deployment to Netlify/Vercel.		

Mapping of Course Outcomes (COs) with Program Outcomes (POs):

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	3	3	2	2	3
CO2	3	3	3	2	2	3
CO3	3	3	3	2	2	3
CO4	3	3	3	2	2	3
CO5	3	3	3	2	2	3

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Mapping of Course Outcomes (COs) with Program Specific Outcomes (PSOs):

CO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	3	2	2	2
CO2	3	3	2	2	2
CO3	3	3	2	2	2
CO4	3	3	2	2	2
CO5	3	3	2	2	3

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Text Books:

- Sabyasachi Mukhopadhyay, 'Modern Frontend Framework Engineering', BPB Publications. (Covers Units: I - V) https://books.google.com/books?q=Modern+Frontend+Framework+Engineering+Sabyasachi+Mukhopadhyay
- Alex Banks, Eve Porcello, 'Learning React: Modern Patterns for Developing React Apps', O'Reilly Media. (Covers Units: I - V) https://books.google.com/books?q=Learning+React%3A+Modern+Patterns+for+Developing+React+Apps+Alex+Banks%2C+Eve+Porcello

Reference Books:

- Adam Freeman, 'Pro Angular 9', Apress. https://books.google.com/books?q=Pro+Angular+9+Adam+Freeman
- Robin Wieruch, 'The Road to React', Independent. https://books.google.com/books?q=The+Road+to+React+Robin+Wieruch
- Boris Cherny, 'Programming TypeScript', O'Reilly Media. https://books.google.com/books?q=Programming+TypeScript+Boris+Cherny

Free Online Study Resources:

- The Odin Project – Full-Stack Web Development (Free) – https://www.theodinproject.com/
- freeCodeCamp – Front End Development Libraries (Free) – https://www.freecodecamp.org/
- React Official Learn Documentation (Free) – https://react.dev/learn

SEMESTER – II
EDC – I - ENTREPRENEURSHIP DEVELOPMENT

Course Objectives:

The main objectives of this course are to:

1. Understand the concept, nature, importance, and types of entrepreneurships in economic and social development.
2. Analyze the entrepreneurial traits, characteristics, motivation theories, and the entrepreneurial decision process.
3. Evaluate business plan preparation, project identification, market analysis, and feasibility study methods.
4. Implement knowledge of institutional support systems, government schemes, and financial assistance for entrepreneurs.
5. Formulate knowledge of small business management, digital marketing, e-commerce, and social entrepreneurship.

Expected Course Outcomes:

On the successful completion of the course, student will be able to:

1	Explain the concept, evolution, types (innovative, imitative, Fabian, drone), and role of entrepreneurship in economic and social development including women and social entrepreneurship.	K1, K2
2	Analyse entrepreneurial traits, McClelland's achievement motivation theory, and the entrepreneurial decision-making process including idea generation, opportunity recognition, and exploitation.	K2, K3
3	Prepare a business plan including project identification, market feasibility study, technical feasibility, financial projections, break-even analysis, and project report preparation.	K3, K4
4	Evaluate institutional support systems (DIC, SIDBI, NABARD, KVIC), government schemes (PMEGP, Startup India, Make in India), and financial assistance mechanisms for new ventures.	K4, K5
5	Apply concepts of small business management, digital marketing, e-commerce, social entrepreneurship, and corporate social responsibility for sustainable business development.	K5, K6

K1 - Remember; K2 - Understand; K3 - Apply; K4 - Analyze; K5 - Evaluate; K6 - Create

Unit:1	**INTRODUCTION TO ENTREPRENEURSHIP**	**15 hours**

Entrepreneurship: concept, definition, characteristics, and importance. Types of entrepreneurships: innovative, imitative, Fabian, drone. Entrepreneur vs. manager vs. intrapreneur. Role of entrepreneurship in economic development. Women entrepreneurship: challenges and opportunities. Social entrepreneurship and corporate social responsibility. Entrepreneurship in the IT sector.

Unit:2	**ENTREPRENEURIAL PERSONALITY AND MOTIVATION**	**15 hours**

Entrepreneurial traits and characteristics. Entrepreneurial competencies. Achievement motivation theory (McClelland). Motivation training programs: EDP (Entrepreneurship Development Programme). Entrepreneurial decision process: idea generation, opportunity recognition, evaluation, and exploitation. Creativity and innovation in entrepreneurship. Risk-taking behaviour and locus of control.		
Unit:3	**BUSINESS PLAN AND PROJECT FORMULATION**	**15 hours**
Business plan: concept, importance, and components. Project identification and selection criteria. Market feasibility study: market survey, demand analysis, competitor analysis. Technical feasibility: technology selection, plant location, plant layout. Financial feasibility: cost of project, means of finance, projected financial statements, working capital requirements, break-even analysis. Project report preparation format.		
Unit:4	**INSTITUTIONAL SUPPORT AND FINANCE**	**15 hours**
Institutional support: DIC (District Industries Centre), SIDCO, NSIC, KVIC, NABARD, SIDBI, NIC. Government schemes: PMEGP, Startup India, Skill India, Make in India, Stand-Up India. Financial institutions: commercial banks, SFCs, SIDCs, venture capital, angel investors, crowdfunding platforms. Procedures for obtaining finance. MSME registration and benefits. Intellectual property rights for entrepreneurs: patents, trademarks, copyrights.		
Unit:5	**SMALL BUSINESS MANAGEMENT AND DIGITAL ENTREPRENEURSHIP**	**15 hours**
Small business management: definition, problems, and prospects. Marketing for entrepreneurs: 4Ps – product, price, place, promotion. Digital marketing strategies: SEO, SEM, social media marketing, content marketing. E-commerce: models (B2B, B2C, C2C), platforms, digital payment systems. Digital entrepreneurship: app development, SaaS products, freelancing platforms. Sustainability in entrepreneurship.		

Mapping of Course Outcomes (COs) with Program Outcomes (POs):

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2	2	2	2	3	3
CO2	2	2	2	2	3	3
CO3	2	2	2	2	3	3
CO4	2	2	2	2	3	3
CO5	2	2	2	2	3	3

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Mapping of Course Outcomes (COs) with Program Specific Outcomes (PSOs):

CO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	2	2	2	2	3
CO2	2	2	2	2	3
CO3	2	2	2	2	3
CO4	2	2	2	2	3
CO5	2	2	2	2	3

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Text Books:

- P. Saravanavel, 'Entrepreneurial Development', Ess Pee Kay Publishing House. (Covers Units: I - V) https://books.google.com/books?q=Entrepreneurial+Development+P.+Saravanavel
- S.S. Khanka, 'Entrepreneurial Development', S. Chand & Company. (Covers Units: I - V) https://books.google.com/books?q=Entrepreneurial+Development+S.S.+Khanka

Reference Books:

- Gupta C.B., Srinivasan N.P., 'Entrepreneurial Development', Sultan Chand & Sons. https://books.google.com/books?q=Entrepreneurial+Development+Gupta+C.B.%2C+Srinivasan+N.P.
- Robert D. Hisrich, Michael P. Peters, 'Entrepreneurship: Starting, Developing, and Managing a New Enterprise', McGraw-Hill. https://books.google.com/books?q=Entrepreneurship%3A+Starting%2C+Developing%2C+and+Managing+a+New+Enterprise+Robert+D.+Hisrich%2C+Michael+P.+Peters
- Peter F. Drucker, 'Innovation and Entrepreneurship', HarperBusiness. https://books.google.com/books?q=Innovation+and+Entrepreneurship+Peter+F.+Drucker

Free Online Study Resources:

- MIT OpenCourseWare – Entrepreneurship (Free) – https://ocw.mit.edu/
- Startup India Learning Hub (Free) – https://learning.startupindia.gov.in/
- NPTEL – Entrepreneurship Essentials (Free) – https://nptel.ac.in/

SEMESTER – II
CORE – X – LAB III - DATA MINING LAB

Learning Objective (LO):
To implement fundamental data mining and machine learning techniques using Python programming and scikit-learn library.

List of Programs:

01. Data Loading and Preprocessing: Write a Python program using Pandas to load a CSV dataset, identify missing values, and apply mean/median imputation techniques.
02. Data Normalization: Apply Min-Max normalization and Z-score standardization on a dataset using scikit-learn's MinMaxScaler and StandardScaler.
03. Exploratory Data Analysis: Use Matplotlib and Seaborn to perform EDA including histograms, box plots, scatter plots, and a correlation heatmap.
04. Apriori Algorithm: Implement the Apriori association rule mining algorithm on a market basket dataset and display rules with support, confidence, and lift values.
05. FP-Growth Algorithm: Implement the FP-Growth algorithm for frequent pattern mining using the mlxtend library and compare execution time with Apriori.
06. Decision Tree Classification: Build a Decision Tree classifier on the Iris dataset using scikit-learn, visualize the decision tree, and evaluate accuracy, precision, and recall.
07. Naive Bayes Classifier: Implement the Gaussian Naive Bayes classifier for email spam detection and evaluate performance using a confusion matrix and F1-score.
08. K-Means Clustering: Implement K-Means clustering on a 2D dataset, determine the optimal k using the Elbow method, and visualize the clusters.
09. Hierarchical Clustering: Implement Agglomerative Hierarchical Clustering using scikit-learn, plot the dendrogram using SciPy, and identify natural clusters.
10. Performance Evaluation and Comparison: Compare Decision Tree, SVM, and k-NN classifiers using 5-fold cross-validation, accuracy, and ROC-AUC metrics on a chosen dataset.

Expected Course Outcomes:
On the successful completion of this practical course, student will be able to:

1	Load, inspect, preprocess, and normalize datasets using Pandas and scikit-learn preprocessing tools.	K1, K2
2	Apply exploratory data analysis techniques using statistical measures and Matplotlib/Seaborn visualization libraries.	K2, K3
3	Implement association rule mining algorithms (Apriori and FP-Growth) and interpret discovered rules using support, confidence, and lift.	K3, K4
4	Build and evaluate classification models using Decision Tree, Naive Bayes, and SVM algorithms with appropriate	K4, K5

	performance metrics.	
5	Apply clustering algorithms (K-Means, Hierarchical) and perform comparative evaluation of multiple classifiers using cross-validation.	K5, K6
K1 - Remember; K2 - Understand; K3 - Apply; K4 - Analyze; K5 - Evaluate; K6 - Create		

Mapping of Course Outcomes (COs) with Program Outcomes (POs):

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	2	2	1	1	2
CO2	2	3	2	1	1	2
CO3	3	3	2	2	1	2
CO4	2	2	3	2	1	3
CO5	3	3	3	2	2	3

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Mapping of Course Outcomes (COs) with Program Specific Outcomes (PSOs):

CO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	2	1	1	2
CO2	2	3	1	1	2
CO3	3	3	2	1	2
CO4	2	2	2	2	3
CO5	3	3	2	2	3

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

SEMESTER – II
CORE – XI – LAB IV - PYTHON PROGRAMMING LAB

Learning Objective (LO):
To implement core Python programming concepts and develop applications using modern Python libraries and frameworks.

List of Programs:

01. Pythonic Code Development: Write a Python program demonstrating best practices using dictionaries, list comprehensions, lambda functions, f-string formatting, and exception handling.
02. Data Analysis using Pandas: Develop a Python application to load, clean, transform, and analyse data from a CSV file using the Pandas library.
03. Object-Oriented Python Application: Create a Python program implementing classes, objects, inheritance, polymorphism, and encapsulation for an Employee Management System.
04. File Handling and Logging: Write a Python application to perform file read/write operations and maintain activity logs using the Python logging module.
05. Database Connectivity: Develop a Python program to connect to a MySQL database and perform CRUD (Create, Read, Update, Delete) operations using MySQL Connector.
06. Version Control using Git and GitHub: Create a Git repository, track project files, perform commits, manage branches, merge branches, and push the project to GitHub.
07. Web Scraping and Data Extraction: Develop a Python application using Requests and BeautifulSoup to extract information from web pages and save results to a CSV file.
08. REST API Development using FastAPI: Create a RESTful API using FastAPI with GET and POST endpoints and test the API endpoints using Postman.
09. Data Visualization Dashboard: Develop a Python application using Matplotlib, Seaborn, and Plotly to create a multi-chart interactive visualization dashboard.
10. Machine Learning Prediction System: Build and evaluate a classification model (Random Forest) using Scikit-learn on a real-world dataset and interpret feature importances.

Expected Course Outcomes:
On the successful completion of this practical course, student will be able to:

1	Apply Python best practices and develop clean, efficient programs using dictionaries, list comprehensions, and exception handling.	K1, K2
2	Load, clean, transform, and analyse real-world datasets using the Pandas library.	K2, K3
3	Design and implement object-oriented Python applications using inheritance, polymorphism, and encapsulation.	K3, K4

4	Develop web scraping, REST API, and data visualization applications using modern Python frameworks.	K4, K5
5	Build and evaluate machine learning classification models using Scikit-learn and interpret results.	K5, K6
K1 - Remember; K2 - Understand; K3 - Apply; K4 - Analyze; K5 - Evaluate; K6 - Create		

Mapping of Course Outcomes (COs) with Program Outcomes (POs):

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	2	2	1	1	2
CO2	2	3	2	1	1	2
CO3	3	3	2	2	1	2
CO4	2	2	3	2	1	3
CO5	3	3	3	2	2	3

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Mapping of Course Outcomes (COs) with Program Specific Outcomes (PSOs):

CO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	2	1	1	2
CO2	2	3	1	1	2
CO3	3	3	2	1	2
CO4	2	2	2	2	3
CO5	3	3	2	2	3

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

SEMESTER – II
FUNDAMENTALS OF HUMAN RIGHTS

Course Objectives:

The main objectives of this course are to:

1. Understand the meaning, characteristics, evolution, and importance of Human Rights.
2. Learn the development of Human Rights in India including Fundamental Rights and Constitutional provisions.
3. Understand rights of marginalized and disadvantaged groups including women, children, and minorities.
4. Learn about major Human Rights Movements including peasant, scheduled caste, and environmental movements.
5. Understand redressal mechanisms including the Protection of Human Rights Act, 1993 and Human Rights Commissions.

Expected Course Outcomes:

On the successful completion of the course, student will be able to:

1	Explain the meaning, evolution, and significance of Human Rights and outline the formation, structure, and functions of the United Nations Organisation.	K1, K2
2	Analyse the development of Human Rights in India including Fundamental Rights, Directive Principles of State Policy, and Fundamental Duties under the Indian Constitution.	K2, K3
3	Evaluate the rights of marginalized groups including women, children, the differently abled, elderly, scheduled castes, tribes, minorities, prisoners, and LGBTQ+ persons.	K3, K4
4	Examine major Human Rights Movements in India including peasant, scheduled caste, tribal, environmental (Chipko, Narmada Bachao), and social reform movements.	K4, K5
5	Apply knowledge of redressal mechanisms including the Protection of Human Rights Act, 1993 (Amendment 2019), and the structure and functions of National and State Human Rights Commissions.	K5, K6

K1 - Remember; K2 - Understand; K3 - Apply; K4 - Analyze; K5 - Evaluate; K6 - Create

Unit:1	**INTRODUCTION**	**2 hours**

Meaning and Definitions of Human Rights – Characteristics and Importance of Human Rights – Evolution of Human Rights – Formation, Structure and Functions of the UNO – Universal Declaration of Human Rights – International Covenants – Violations of Human Rights in the Contemporary Era.

Unit:2	**HUMAN RIGHTS IN INDIA**	**2 hours**

Development of Human Rights in India – Constituent Assembly and Indian Constitution – Fundamental Rights and its Classification – Directive Principles of State Policy – Fundamental

Duties.		
Unit:3	**RIGHTS OF MARGINALIZED AND OTHER DISADVANTAGED PEOPLE**	**2 hours**
Rights of Women – Rights of Children – Rights of Differently Abled – Rights of Elderly – Rights of Scheduled Castes – Rights of Scheduled Tribes – Rights of Minorities – Rights of Prisoners – Rights of Persons Living with HIV/AIDS – Rights of LGBT.		
Unit:4	**HUMAN RIGHTS MOVEMENTS**	**2 hours**
Peasant Movements (Tebhaga and Telangana) – Scheduled Caste Movements (Mahar and Ad-Dharmi) – Scheduled Tribes Movements (Santhal and Munda) – Environmental Movements (Chipko and Narmada Bachao Andolan) – Social Reform Movements (Vaikom and Self Respect).		
Unit:5	**REDRESSAL MECHANISMS**	**2 hours**
Protection of Human Rights Act, 1993 (Amendment 2019) – Structure and Functions of National and State Human Rights Commissions – National Commission for SCs – National Commission for STs – National Commission for Women – National Commission for Minorities – Characteristics and Objectives of Human Rights Education.		

Mapping of Course Outcomes (COs) with Program Outcomes (POs):

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2	1	1	1	3	2
CO2	2	1	1	1	3	2
CO3	2	1	1	1	3	2
CO4	2	1	1	1	3	2
CO5	2	1	1	1	3	2

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Mapping of Course Outcomes (COs) with Program Specific Outcomes (PSOs):

CO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	2	1	1	1	3
CO2	2	1	1	1	3
CO3	2	1	1	1	3
CO4	2	1	1	1	3
CO5	2	1	1	1	3

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Text Books:

- S. K. Kapoor, 'International Law and Human Rights', Central Law Agency. (Covers Units: I - V) https://books.google.com/books?q=International+Law+and+Human+Rights+S.+K.+Kapoor
- U. Chandra, 'Human Rights', Allahabad Law Agency. (Covers Units: I - V) https://books.google.com/books?q=Human+Rights+U.+Chandra

Reference Books:

- A.S. Narang, 'Human Rights: Issues and Perspectives', Gitanjali Publishing House. https://books.google.com/books?q=Human+Rights%3A+Issues+and+Perspectives+A.S.+Narang
- Henry J. Steiner, Philip Alston, 'International Human Rights in Context', Oxford University Press. https://books.google.com/books?q=International+Human+Rights+in+Context+Henry+J.+Steiner%2C+Philip+Alston
- S. Guha Roy, 'Human Rights in India', Calcutta. https://books.google.com/books?q=Human+Rights+in+India+S.+Guha+Roy

Free Online Study Resources:

- UN Human Rights – Official Website (Free) – https://www.ohchr.org/
- National Human Rights Commission India (Free) – https://nhrc.nic.in/
- NHRC E-Learning (Free) – https://nhrc.nic.in/

SEMESTER – III
CORE – XII - INTERNET OF THINGS

Course Objectives:

The main objectives of this course are to:

1. Comprehend IoT system topologies, microcontroller engineering setups, and sensor signal conversions.
2. Analyze IEEE 802.
3. 4, LoRaWAN, ZigBee topologies, and 6LoWPAN network layouts.
4. Evaluate MQTT protocol internals, CoAP design, and AMQP message pipeline configurations.
5. Implement Edge, Fog, and Cloud tiered computing architectures and containerization at the edge.

Expected Course Outcomes:

On the successful completion of the course, student will be able to:

1	Configure IoT system topologies and microcontroller hardware (ESP32, Arduino, Raspberry Pi) for sensor signal conversion (ADC), PWM control, and actuator operations.	K1, K2
2	Analyse low-power embedded network protocols including IEEE 802.15.4, LoRaWAN, ZigBee mesh networks, and 6LoWPAN for constrained IoT environments.	K2, K3
3	Implement application-layer IoT protocols including MQTT publish-subscribe messaging (QoS 0/1/2), CoAP REST- constrained operations, and AMQP pipeline configurations.	K3, K4
4	Design Edge, Fog, and Cloud tiered computing architectures, deploy local analytics on gateways, and containerise edge workloads using lightweight Docker engines.	K4, K5
5	Evaluate Industrial IoT security vulnerabilities (threat modelling), apply lightweight encryption algorithms (AES-128, ECC), and analyse smart grid and connected health operational case studies.	K5, K6

K1 - Remember; K2 - Understand; K3 - Apply; K4 - Analyze; K5 - Evaluate; K6 - Create

Unit:1	**IoT ARCHITECTURE AND EDGE DEVICES**	**12 hours**

IoT definition, scope, and application domains. System topologies: star, mesh, hybrid. Logical IoT architectures (3-layer, 5-layer). Microcontroller engineering: ESP32 (dual-core, Wi-Fi/Bluetooth), Arduino Uno, Raspberry Pi 4 hardware layouts. Sensor types: temperature (DHT11), humidity, light (LDR), motion (PIR). Analog-to-Digital Conversion (ADC) principles. PWM signal generation for actuator control.

Unit:2	**IoT NETWORK PROTOCOLS**	**12 hours**

Data Link Layer optimization for constrained devices: IEEE 802.15.4 link parameters, channel access mechanisms. LoRaWAN: spread spectrum modulation, star-of-stars topology, network server architecture. ZigBee: mesh topology, coordinator/router/end-device roles. IPv6 adaptation

over low-power lossy networks: 6LoWPAN header compression, RPL routing protocol for constrained networks.		
Unit:3	**IoT APPLICATION LAYER PROTOCOLS**	**12 hours**
Application layer requirements for constrained IoT devices. MQTT protocol: publish-subscribe architecture, broker-based messaging, topics and wildcards, Quality of Service levels (QoS 0, 1, 2), retained messages and last-will testament. CoAP protocol: REST-based constrained application operations, request/response model over UDP, resource discovery, CoAP-HTTP proxying. AMQP protocol: message queuing model, exchanges, bindings and queues, reliable pipeline configurations for enterprise IoT integration. Comparative analysis of MQTT, CoAP, and AMQP for bandwidth, latency, and reliability trade-offs. RESTful web services for IoT device management.		
Unit:4	**EDGE AND CLOUD COMPUTING FOR**	**12 hours**
IoT Limitations of centralized cloud for massive IoT streams: latency, bandwidth, privacy. Edge computing: processing at the device or gateway level. Fog computing: distributed intelligence between edge and cloud. Cloud IoT platforms: AWS IoT Core, Google Cloud IoT, Azure IoT Hub. Local analytics on edge gateways using Node-RED and Python. Containerization at the edge using Docker and Kubernetes lightweight variants (K3s).		
Unit:5	**INDUSTRIAL IoT AND SECURITY**	**12 hours**
Cyber security threats in IoT: device hijacking, man-in-the-middle attacks, DDoS using botnets (Mirai). IoT threat modelling: STRIDE methodology. Lightweight cryptography: AES-128, ChaCha20, Elliptic Curve Cryptography (ECC). TLS/DTLS for secure MQTT and CoAP. Case studies: Industry 4.0 – smart manufacturing, predictive maintenance; Smart Energy Grids – demand response; Connected Health – remote patient monitoring.		

Mapping of Course Outcomes (COs) with Program Outcomes (POs):

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	2	3	2	2	3
CO2	3	2	3	2	2	3
CO3	3	2	3	2	2	3
CO4	3	2	3	2	2	3
CO5	3	2	3	2	3	3

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Mapping of Course Outcomes (COs) with Program Specific Outcomes (PSOs):

CO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	2	3	2	2
CO2	3	2	3	2	2

CO3	3	2	3	2	2
CO4	3	2	3	2	2
CO5	3	2	3	2	3

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Text Books:

- Raj Kamal, 'Internet of Things: Architecture and Design Principles', McGraw-Hill India. (Covers Units: I - V) https://books.google.com/books?q=Internet+of+Things%3A+Architecture+and+Design+Principles+Raj+Kamal
- Perry Lea, 'IoT and Edge Computing Architecture', Packt Publishing. (Covers Units: I - V) https://books.google.com/books?q=IoT+and+Edge+Computing+Architecture+Perry+Lea

Reference Books:

- Arshdeep Bahga, Vijay Madisetti, 'Internet of Things: A Hands-On Approach', Universities Press. https://books.google.com/books?q=Internet+of+Things%3A+A+Hands-On+Approach+Arshdeep+Bahga%2C+Vijay+Madisetti
- David Hanes, Gonzalo Salgueiro, 'IoT Fundamentals', Cisco Press. https://books.google.com/books?q=IoT+Fundamentals+David+Hanes%2C+Gonzalo+Salgueiro
- Buyya Rajkumar, 'Fog and Edge Computing: Principles and Paradigms', Wiley. https://books.google.com/books?q=Fog+and+Edge+Computing%3A+Principles+and+Paradigms+Buyya+Rajkumar

Free Online Study Resources:

- NPTEL – Internet of Things (IIT Kharagpur, Free) – https://nptel.ac.in/courses/106105166
- Raspberry Pi Official Documentation (Free) – https://www.raspberrypi.com/documentation/
- Arduino Project Hub & Tutorials (Free) – https://projecthub.arduino.cc/

SEMESTER – III
CORE – XIII - .NET PROGRAMMING

Course Objectives:

The main objectives of this course are to:

1. Understand the .NET framework architecture, CLR, CTS, CLS, and C# programming fundamentals.
2. Learn object-oriented programming concepts in C# including classes, inheritance, interfaces, generics, and delegates.
3. Develop Windows Forms and WPF applications with event-driven programming and data binding.
4. Design and implement ADO.NET for database connectivity, LINQ, and Entity Framework Core for data access.
5. Build web applications using ASP.NET Core MVC and RESTful Web API with authentication and Swagger documentation.

Expected Course Outcomes:

On the successful completion of the course, student will be able to:

1	Explain the .NET framework architecture including CLR, CTS, CLS, MSIL, JIT compiler, and implement C# programming constructs including data types, control flow, and exception handling.	K1, K2
2	Develop object-oriented C# programs using classes, inheritance, interfaces, abstract classes, delegates, events, generics, lambda expressions, and extension methods.	K2, K3
3	Build Windows Forms applications with event-driven programming and WPF applications using XAML, data binding, and MVVM pattern.	K3, K4
4	Implement ADO.NET for connected and disconnected database access, apply LINQ to SQL and Entity Framework Core with Code First approach and migrations.	K4, K5
5	Develop ASP.NET Core MVC web applications and create RESTful Web APIs with JWT authentication, authorization, and Swagger/OpenAPI documentation.	K5, K6

K1 - Remember; K2 - Understand; K3 - Apply; K4 - Analyze; K5 - Evaluate; K6 - Create

Unit:1	**.NET FRAMEWORK AND C# FUNDAMENTALS**	**15 hours**

.NET Framework architecture: CLR, CTS, CLS, MSIL, JIT compiler, assemblies, GAC. .NET Core and .NET 6/7 overview. Namespaces and using directives. C# programming: value types, reference types, boxing/unboxing, operators, control structures (if-else, switch, for, foreach, while, do-while). Arrays, Strings (String vs. StringBuilder). Methods: overloading, optional and named parameters, ref and out parameters. Exception handling: try-catch-finally, custom exceptions, exception hierarchy. File I/O: StreamReader, StreamWriter, FileStream.

Unit:2	**OBJECT-ORIENTED PROGRAMMING IN C#**	**15 hours**

Classes and objects, access modifiers, constructors, destructors, properties, auto-implemented properties, indexers. Inheritance: single-level, multi-level, constructor chaining, sealed classes. Abstract classes and abstract methods. Interfaces, explicit interface implementation, multiple interface inheritance. Polymorphism: method overloading, method overriding, virtual and override keywords. Generics: generic classes, generic methods, constraints (where clause). Delegates: Func< > Action< > Predicate< > Events and event handlers. Lambda expressions and anonymous methods. Extension methods.		
Unit:3	**WINDOWS FORMS AND WPF**	**15 hours**
Windows Forms: form design, common controls (Button, TextBox, ListBox, ComboBox, DataGridView, MenuStrip, ToolStrip), dialog boxes. Event-driven programming: event handlers, event arguments. MDI applications. WPF: XAML markup language, layout panels (Grid, StackPanel, DockPanel, WrapPanel). Controls: TextBox, Button, ListBox, DataGrid, TabControl. Styles, Control Templates. Data binding: one-way, two-way, multi-binding, DataContext, INotifyPropertyChanged. MVVM design pattern: Model, View, ViewModel, RelayCommand.		
Unit:4	**ADO.NET, LINQ, AND ENTITY FRAMEWORK ADO.NET:**	**15 hours**
SqlConnection, SqlCommand, SqlDataReader (connected mode), SqlDataAdapter, DataSet, DataTable (disconnected mode). Transactions in ADO.NET. LINQ (Language Integrated Query): LINQ to Objects (query syntax and method syntax), LINQ to XML. LINQ operators: Select, Where, OrderBy, GroupBy, Join, Aggregate. Entity Framework Core: Code First approach, DbContext, DbSet< T> data annotations and Fluent API. Migrations: Add-Migration, Update-Database. CRUD operations with EF Core. Repository and Unit of Work patterns.		
Unit:5	**ASP.NET CORE AND WEB API**	**15 hours**
ASP.NET Core MVC: project structure, Startup.cs (Program.cs in .NET 6+), middleware pipeline, dependency injection container. Controllers, Views (Razor), Tag Helpers, Partial Views, ViewBag/ViewData/TempData. Routing: conventional and attribute routing, route constraints. Model binding and validation (Data Annotations). Identity Framework:		

Mapping of Course Outcomes (COs) with Program Outcomes (POs):

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	3	3	2	2	3
CO2	3	3	3	2	2	3
CO3	3	3	3	2	2	3
CO4	3	3	3	2	2	3
CO5	3	3	3	2	2	3

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Mapping of Course Outcomes (COs) with Program Specific Outcomes (PSOs):

CO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	3	2	2	2
CO2	3	3	2	2	2
CO3	3	3	2	2	2
CO4	3	3	2	2	2
CO5	3	3	2	2	3

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Text Books:

- Andrew Troelsen, Philip Japikse, 'Pro C# 10 with .NET 6', Apress. (Covers Units: I - V) https://books.google.com/books?q=Pro+C%23+10+with+.NET+6+Andrew+Troelsen%2C+Philip+Japikse
- Christian Nagel, 'Professional C# and .NET', Wrox Wiley. (Covers Units: I - V) https://books.google.com/books?q=Professional+C%23+and+.NET+Christian+Nagel

Reference Books:

- Adam Freeman, 'Pro ASP.NET Core MVC', Apress. https://books.google.com/books?q=Pro+ASP.NET+Core+MVC+Adam+Freeman
- Mark Michaelis, Eric Lippert, 'Essential C#', Addison-Wesley. https://books.google.com/books?q=Essential+C%23+Mark+Michaelis%2C+Eric+Lippert
- Joseph Albahari, Ben Albahari, 'C# in a Nutshell', O'Reilly Media. https://books.google.com/books?q=C%23+in+a+Nutshell+Joseph+Albahari%2C+Ben+Albahari

Free Online Study Resources:

- Microsoft .NET Documentation (Free) – https://docs.microsoft.com/en-us/dotnet/
- Microsoft Learn – C# Fundamentals (Free) – https://learn.microsoft.com/
- Dot Net Perls (Free Tutorials) – https://www.dotnetperls.com/

SEMESTER – III
CORE – XIV - BIG DATA ANALYTICS

Course Objectives:

The main objectives of this course are to:

1. Comprehend the 5 Vs of Big Data, distributed computing challenges, and HDFS cluster architecture.
2. Analyze MapReduce processing steps, distributed data processing jobs, and optimization parameters.
3. Evaluate Apache Hive metadata layout, HiveQL compilation, and Apache Pig Latin data pipelines.
4. Implement Apache Spark Core engine, RDD data lineage models, DAG scheduling, and lazy evaluation.
5. Formulate Spark SQL DataFrames, query optimizations via Catalyst Optimizer, and real-time Spark Structured Streaming.

Expected Course Outcomes:

On the successful completion of the course, student will be able to:

1	Explain Big Data characteristics (5 V's) and configure HDFS architecture including block replication, NameNode high-availability, and read/write operational pipelines.	K1, K2
2	Design and execute distributed MapReduce jobs using custom Mappers, Reducers, Combiners, Partitioners, and Distributed Cache optimisation techniques.	K2, K3
3	Apply Apache Hive for data warehousing with HiveQL, table partitioning, and bucketing, and Apache Pig Latin for scripting relational data pipeline operations with UDFs.	K3, K4
4	Implement Apache Spark computations using RDDs, DAG-based task scheduling, lazy evaluation transformations (map, filter, reduceByKey), and immediate action execution.	K4, K5
5	Build real-time data processing systems using Spark SQL DataFrames, Catalyst Optimizer, Structured Streaming, window operations, and Kafka streaming endpoint consumer mapping.	K5, K6

K1 - Remember; K2 - Understand; K3 - Apply; K4 - Analyze; K5 - Evaluate; K6 - Create

Unit:1	**BIG DATA AND HADOOP DISTRIBUTED FILE SYSTEM**	**15 hours**

The 5 Vs of Big Data: Volume, Velocity, Variety, Veracity, and Value. Big Data challenges: heterogeneity, scale, timeliness, privacy. Hadoop ecosystem overview. HDFS architecture: NameNode (master), DataNodes (slaves), Secondary NameNode. Block replication: default replication factor (3), rack-awareness policy. Read and write data pipelines. NameNode high-availability using ZooKeeper. YARN: ResourceManager, NodeManager, ApplicationMaster.

Unit:2	**MAPREDUCE PROGRAMMING MODEL**	**15 hours**

MapReduce programming paradigm and execution flow. Map phase: input splits, Mapper

function, key-value pair generation. Shuffle and Sort phase: partitioning, sorting, and grouping. Reduce phase: Reducer function, output key-value pairs. Designing distributed data processing jobs. Optimization techniques: Combiner (mini-reducer) for local aggregation, Custom Partitioner for load balancing, Distributed Cache for small file distribution. MapReduce counters and diagnostics.		
Unit:3	**APACHE HIVE AND PIG FOR DATA WAREHOUSING**	**15 hours**
Apache Hive architecture: Hive Metastore, Driver, Compiler, and Execution Engine. HiveQL: data definition (CREATE, ALTER, DROP), data manipulation (LOAD, INSERT), and query operations, joins, and sub-queries. Hive table types: managed and external tables, partitioning and bucketing for query optimization. HiveQL compilation and execution plan generation. Apache Pig architecture and execution modes: local and MapReduce mode. Pig Latin scripting: LOAD, FILTER, FOREACH, GROUP, JOIN, and STORE operators. Data pipeline design using Pig Latin for extract-transform-load (ETL) workloads. Comparative analysis of Hive and Pig for data warehousing and analytical workloads on Hadoop.		
Unit:4	**APACHE SPARK CORE AND RDD**	**15 hours**
MapReduce limitations: disk I/O bottlenecks, high latency. Apache Spark architecture: Driver, SparkContext, Cluster Manager (Standalone, YARN, Mesos), Executors. Resilient Distributed Datasets (RDD): creation (parallelize, textFile), partitioning, lineage graph (DAG). Transformations (lazy): map, flatMap, filter, union, distinct, groupByKey, reduceByKey, sortByKey, join. Actions (eager): collect, count, first, take, saveAsTextFile, reduce. DAG-based task scheduling: stages, tasks, shuffle operations. Persistence: cache () and persist () with storage levels.		
Unit:5	**SPARK SQL AND STREAM PROCESSING**	**15 hours**
Spark SQL: DataFrames and Datasets abstractions, SparkSession. Reading from multiple sources: CSV, JSON, Parquet, JDBC. DataFrame operations: select, filter, withColumn, groupBy, agg, join, window functions. Catalyst Optimizer: logical plan, physical plan, code generation. Spark Structured Streaming: streaming DataFrames, trigger options, output modes (append, update, complete). Window operations: tumbling windows, sliding windows. Kafka integration: KafkaSource connector, consumer group configuration. Checkpointing for fault tolerance.		

Mapping of Course Outcomes (COs) with Program Outcomes (POs):

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2	1	3	3	1	1
CO2	2	1	3	3	1	1
CO3	2	1	3	3	1	1
CO4	2	2	3	3	1	2
CO5	1	1	3	3	1	2

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Mapping of Course Outcomes (COs) with Program Specific Outcomes (PSOs):

CO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	2	1	1	3	1
CO2	2	1	1	3	1
CO3	2	1	1	3	1
CO4	2	1	1	3	1
CO5	1	1	1	3	1

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Text Books:

- V. K. Jain, 'Big Data Analytics using Hadoop and Spark', Khanna Publishing Company. (Covers Units: I - V) https://books.google.com/books?q=Big+Data+Analytics+using+Hadoop+and+Spark+V.+K.+Jain
- Tom White, 'Hadoop: The Definitive Guide', O'Reilly Media. (Covers Units: I - V) https://books.google.com/books?q=Hadoop%3A+The+Definitive+Guide+Tom+White

Reference Books:

- Radha Shankarmani, M. Vijayalakshmi, 'Big Data Analytics', Wiley India. https://books.google.com/books?q=Big+Data+Analytics+Radha+Shankarmani%2C+M.+Vijayalakshmi
- Holden Karau et al., 'Learning Spark', O'Reilly Media. https://books.google.com/books?q=Learning+Spark+Holden+Karau+et+al.
- Bill Chambers, Matei Zaharia, 'Spark: The Definitive Guide', O'Reilly Media. https://books.google.com/books?q=Spark%3A+The+Definitive+Guide+Bill+Chambers%2C+Matei+Zaharia

Free Online Study Resources:

- Apache Spark Official Documentation (Free) – https://spark.apache.org/docs/latest/
- Hadoop Official Documentation (Free) – https://hadoop.apache.org/docs/
- NPTEL – Big Data Computing (Free) – https://nptel.ac.in/

SEMESTER – III
ELECTIVE – III - ADVANCED LEARNING ALGORITHMS & STATISTICAL MODELLING

Course Objectives:

The main objectives of this course are to:

1. Comprehend supervised, unsupervised, and semi-supervised learning approaches and regularization techniques.
2. Analyze logistic regression, decision tree node-splitting measures, and Support Vector Machines.
3. Evaluate ensemble learning algorithms including Bagging with Random Forests and Boosting with XGBoost.
4. Implement K-Means optimization, DBSCAN clustering, PCA, and SVD for pattern discovery.
5. Formulate deep neural networks using Backpropagation, activation functions, and Adam optimizers.

Expected Course Outcomes:

On the successful completion of the course, student will be able to:

1	Explain supervised, unsupervised, and semi-supervised learning paradigms, bias-variance trade-off, loss formulations, regularisation techniques (L1/L2), and cross-validation pipelines.	K1, K2
2	Implement parametric and tree-based classifiers including Logistic Regression, Decision Trees using Entropy and Gini criteria (ID3, C4.5), and Support Vector Machines (linear and kernel-based).	K2, K3
3	Apply ensemble learning methods including Bagging with Random Forests, and Boosting with AdaBoost, Gradient Boosting, and XGBoost with hyperparameter tuning using GridSearchCV.	K3, K4
4	Perform unsupervised clustering using K-Means and DBSCAN algorithms, and apply dimensionality reduction via PCA (explained variance) and SVD for pattern discovery.	K4, K5
5	Design and train deep neural networks using Backpropagation, ReLU/Tanh/Softmax activations, and SGD, Momentum, RMSProp, and Adam optimisation algorithms.	K5, K6

K1 - Remember; K2 - Understand; K3 - Apply; K4 - Analyze; K5 - Evaluate; K6 - Create

Unit:1	**FOUNDATIONS OF MACHINE LEARNING**	**12 hours**

Supervised, Unsupervised, and Semi-supervised learning approaches. Generative vs. Discriminative models. The Bias-Variance Tradeoff: underfitting, overfitting, and the optimal model complexity. Loss functions: Ordinary Least Squares (MSE), Binary Cross-Entropy, Categorical Cross-Entropy, Hinge Loss. Regularization techniques: L1 Lasso (feature selection), L2 Ridge (weight shrinkage), Elastic Net. K-Fold Cross-validation and Stratified K-Fold pipelines.

Unit:2	CLASSIFICATION ALGORITHMS	12 hours
Logistic Regression: sigmoid function, decision boundary, gradient descent optimization, multi-class (softmax). Decision Tree induction: attribute selection measures – Shannon Entropy, Information Gain (ID3), Gain Ratio (C4.5), and Gini Impurity (CART). Tree pruning: pre-pruning (max depth, min samples), post-pruning (cost-complexity pruning). Support Vector Machines (SVM): hyperplane equations, maximal margin classifier, support vectors, Lagrange multipliers, kernel trick (RBF, polynomial, sigmoid).		
Unit:3	**ENSEMBLE LEARNING METHODS**	**12 hours**
Ensemble learning motivation: wisdom of crowds. Bagging: Bootstrap aggregation mechanics, Out-of-Bag error estimation. Random Forests: feature sub-sampling, variable importance (MDI, Permutation Importance). Boosting: sequential residual minimization. AdaBoost: adaptive weighting of weak learners. Gradient Boosting Machines (GBM): gradient in function space. XGBoost: regularized objective, tree pruning (gamma, min_child_weight), hyperparameter tuning using GridSearchCV/RandomizedSearchCV.		
Unit:4	**UNSUPERVISED LEARNING AND DIMENSIONALITY REDUCTION**	**12 hours**
Partitioning Algorithms: K-Means initialization (random, K-Means++), Lloyd's algorithm iterations, Elbow Method, Silhouette Score for optimal k. Density-Based Clustering: DBSCAN core point definition (eps, min_samples), border points, noise points, algorithm flow. Dimensionality Reduction: Principal Component Analysis (PCA) – covariance matrix, eigenvectors, explained variance ratio, scree plot. Singular Value Decomposition (SVD) – relationship to PCA, truncated SVD for sparse data.		
Unit:5	**DEEP NEURAL NETWORKS**	**12 hours**
Biological neuron analogy and Artificial Neuron (perceptron). Multi-Layer Perceptrons (MLP): input, hidden, and output layers. Forward propagation. Backpropagation algorithm derivation using the chain rule: weight update equations. Non-linear activation functions: Step, Sigmoid, Tanh, ReLU (and variants: Leaky ReLU, PReLU), Softmax for multi-class. Weight initialization: Xavier/Glorot, He initialization. Gradient Descent variants: Batch GD, Stochastic GD (SGD), Mini-Batch GD. Momentum, RMSProp, Adam optimizer.		

Mapping of Course Outcomes (COs) with Program Outcomes (POs):

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	2	3	3	2	3
CO2	3	2	3	3	2	3
CO3	2	3	3	3	2	3
CO4	2	3	3	3	2	3
CO5	2	2	3	3	3	3

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Mapping of Course Outcomes (COs) with Program Specific Outcomes (PSOs):

CO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	2	3	3	3
CO2	3	2	3	3	3
CO3	3	3	3	3	3
CO4	3	3	3	3	3
CO5	3	2	3	3	3

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Text Books:

- Anuradha Srinivasaraghavan, S. Shuttleworth, 'Introduction to Machine Learning', Wiley India Edition. (Covers Units: I - V) https://books.google.com/books?q=Introduction+to+Machine+Learning+Anuradha+Srinivasaraghavan%2C+S.+Shuttleworth
- Aurelien Geron, 'Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow', O'Reilly Media. (Covers Units: I - V)

  https://books.google.com/books?q=Hands-On+Machine+Learning+with+Scikit-Learn%2C+Keras%2C+and+TensorFlow+Aurelien+Geron

Reference Books:

- Gareth James et al., 'An Introduction to Statistical Learning', Springer. https://books.google.com/books?q=An+Introduction+to+Statistical+Learning+Gareth+James+et+al.
- Ian Goodfellow, Yoshua Bengio, Aaron Courville, 'Deep Learning', MIT Press. https://books.google.com/books?q=Deep+Learning+Ian+Goodfellow%2C+Yoshua+Bengio%2C+Aaron+Courville
- Christopher M. Bishop, 'Pattern Recognition and Machine Learning', Springer. https://books.google.com/books?q=Pattern+Recognition+and+Machine+Learning+Christopher+M.+Bishop

Free Online Study Resources:

- An Introduction to Statistical Learning (Free PDF) – https://www.statlearning.com/
- Deep Learning Book – Goodfellow et al. (Free Online) – https://www.deeplearningbook.org/
- NPTEL – Introduction to Machine Learning (IIT Madras, Free) – https://nptel.ac.in/courses/106106139

SEMESTER – III
ELECTIVE – IV - GENERATIVE AI ENGINEERING AND INTELLIGENT AGENTS

Course Objectives:

The main objectives of this course are to:

1. Comprehend Transformer architecture, Encoder-Decoder stacks, and Scaled Dot-Product Self-Attention mechanism.
2. Analyze LLM pre-training, fine-tuning, instruction tuning, and autoregressive decoding strategies.
3. Evaluate LangChain framework for building composable AI applications with memory and chain systems.
4. Implement Retrieval-Augmented Generation (RAG) pipelines using vector embeddings and vector databases.
5. Formulate prompt engineering architectures including Few-shot, Chain-of-Thought, and ReAct frameworks.

Expected Course Outcomes:

On the successful completion of the course, student will be able to:

1	Explain the Transformer architecture including Scaled Dot-Product Self-Attention, Multi-Head Attention mathematical formulations, positional encodings, layer normalization, and residual connections.	K1, K2
2	Describe LLM pre-training objectives, fine-tuning (PEFT, LoRA), instruction tuning, and autoregressive decoding strategies (greedy, top-k, top-p, temperature sampling) across GPT, BERT, and Llama architectures.	K2, K3
3	Build composable AI applications using LangChain framework with prompt templates, LLM chain pipelines, conversation memory systems, and multi-agent task collaboration loops.	K3, K4
4	Design Retrieval-Augmented Generation (RAG) pipelines using text chunking strategies, vector embedding models, and vector database retrieval with cosine similarity and dot product measures.	K4, K5
5	Apply prompt engineering techniques (few-shot, Chain-of-Thought, Tree-of-Thought, ReAct), integrate LLM APIs for code generation, and evaluate model alignment using RLHF and RLAIF benchmarks.	K5, K6

K1 - Remember; K2 - Understand; K3 - Apply; K4 - Analyze; K5 - Evaluate; K6 - Create

Unit:1	**TRANSFORMER ARCHITECTURE AND ATTENTION MECHANISM**	**12 hours**

Limitations of sequence models: RNN vanishing gradient, LSTM computational bottlenecks. The Transformer architecture blueprint: Encoder stack and Decoder stack. Scaled Dot-Product Attention: Query, Key, Value matrices, attention score computation, softmax normalization. Multi-Head Attention: parallel attention heads, concatenation, linear projection. Positional

encodings: sinusoidal encoding. Layer normalization (pre-LN vs. post-LN). Residual (skip) connections. Feed-Forward sublayer: position-wise FFN.		
Unit:2	**LARGE LANGUAGE MODELS AND TEXT GENERATION LLM**	**12 hours**
pre-training objectives: Causal Language Modelling (CLM) for GPT, Masked Language Modelling (MLM) for BERT. Fine-tuning paradigms: full fine-tuning, PEFT (Parameter-Efficient Fine-Tuning): LoRA (Low-Rank Adaptation), adapters, prefix tuning. Instruction tuning (InstructGPT). Autoregressive token prediction mechanics. Decoding strategies: greedy decoding, beam search, temperature scaling, Top-k sampling, Top-p (nucleus) sampling. Architectural profiles: GPT series (decoder-only), BERT (encoder-only), T5 (encoder-decoder), Llama (open-source decoder).		
Unit:3	**AI APPLICATION DEVELOPMENT WITH LANGCHAIN**	**12 hours**
LangChain framework architecture: Chains, Agents, Tools, Memory modules. Prompt templates: PromptTemplate, ChatPromptTemplate, FewShotPromptTemplate. LLM wrappers: OpenAI, HuggingFace, Ollama. Sequential chains: SimpleSequentialChain, SequentialChain. Memory systems: ConversationBufferMemory, ConversationWindowMemory, ConversationSummaryMemory. LangChain agents: ReAct agent, Tool usage (search, calculator). Multi-agent collaboration frameworks: AutoGen overview.		
Unit:4	**RETRIEVAL-AUGMENTED GENERATION (RAG) RAG**	**12 hours**
architecture: motivation, Retrieval + Generation pipeline, benefits over pure LLM. Document loading and text chunking: fixed-size chunking, recursive text splitter, semantic chunking strategies. Vector Embeddings: word embeddings (Word2Vec, GloVe), sentence embeddings (SBERT). Vector Databases: Chroma, FAISS, Pinecone, Weaviate. Indexing mechanisms: HNSW (Hierarchical Navigable Small World), IVF (Inverted File Index). Similarity measures: cosine similarity, Euclidean distance, dot product. Retrieval pipelines: similarity search, MMR (Maximum Marginal Relevance). RAG evaluation: context relevance, faithfulness, answer relevance (RAGAS framework).		
Unit:5	**PROMPT ENGINEERING AND AI TOOLS INTEGRATION**	**12 hours**
Prompt Engineering principles: role-setting, task specification, background context, constraints (RTBC framework). Few-shot prompting: in-context learning with examples. Chain-of-Thought (CoT) prompting: step-by-step reasoning		

Mapping of Course Outcomes (COs) with Program Outcomes (POs):

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	2	3	2	3	2
CO2	2	2	3	2	3	2
CO3	2	2	3	2	3	2
CO4	3	3	3	2	3	3

CO5	2	2	3	2	3	3

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Mapping of Course Outcomes (COs) with Program Specific Outcomes (PSOs):

CO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	2	3	2	3
CO2	2	2	3	2	2
CO3	2	2	3	2	2
CO4	3	3	3	2	3
CO5	3	2	3	2	3

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Text Books:

- Kamal Kant Hiran, 'Artificial Intelligence and Generative AI Concepts', BPB Publications. (Covers Units: I - V) https://books.google.com/books?q=Artificial+Intelligence+and+Generative+AI+Concepts+Kamal+Kant+Hiran
- Ben Auffarth, 'Generative AI with LangChain: Build LLM apps', Packt Publishing. (Covers Units: I - V) https://books.google.com/books?q=Generative+AI+with+LangChain%3A+Build+LLM+apps+Ben+Auffarth

Reference Books:

- Denis Rothman, 'Transformers for Natural Language Processing', Packt Publishing. https://books.google.com/books?q=Transformers+for+Natural+Language+Processing+Denis+Rothman
- James Phoenix, Mike Taylor, 'Prompt Engineering for LLMs', O'Reilly Media. https://books.google.com/books?q=Prompt+Engineering+for+LLMs+James+Phoenix%2C+Mike+Taylor
- Lewis Tunstall et al., 'Natural Language Processing with Transformers', O'Reilly Media. https://books.google.com/books?q=Natural+Language+Processing+with+Transformers+Lewis+Tunstall+et+al.

Free Online Study Resources:

- Hugging Face NLP Course (Free) – https://huggingface.co/learn/nlp-course/
- The Illustrated Transformer – Jay Alammar (Free) – https://jalammar.github.io/illustrated-transformer/
- Google Generative AI Learning Path (Free) – https://cloudskillsboost.google/journeys/118

SEMESTER – III
CORE – XV – .NET PROGRAMMING LAB

Learning Objective (LO):
To develop .NET applications using C#, ADO.NET, LINQ, Entity Framework Core, and ASP.NET Core frameworks.

List of Programs:

01. C# OOP Fundamentals: Create a C# console application with classes demonstrating constructors, inheritance, method overriding, and the SOLID principles.
02. Exception Handling in C#: Implement a C# program with multiple try-catch-finally blocks, custom exception classes, and exception chaining.
03. Generic Collections: Use List<T>, Dictionary<K, V>, Queue<T>, and Stack<T> to implement a student grade management system.
04. Delegates, Events, and Lambda: Implement a publisher-subscriber model using delegates, events, and lambda expressions for an order notification system.
05. Windows Forms Calculator: Design and implement a scientific calculator using Windows Forms with arithmetic, trigonometric, and logarithmic functions.
06. ADO.NET Database Operations: Use SqlConnection, SqlCommand, and SqlDataAdapter to perform INSERT, SELECT, UPDATE, and DELETE operations on an Employee table in SQL Server.
07. LINQ Queries: Write C# programs demonstrating LINQ to Objects queries (filtering, grouping, sorting, joining) on a list of student objects.
08. Entity Framework Core CRUD: Implement a Code First EF Core application with a Product entity, migrations, and full CRUD operations.
09. ASP.NET Core Web API: Create a RESTful Web API for a Library Management System with GET, POST, PUT, DELETE endpoints and Swagger documentation.
10. ASP.NET Core MVC Application: Develop a full ASP.NET Core MVC application for Student Management with views, controllers, Razor Pages, and SQL Server backend.

Expected Course Outcomes:
On the successful completion of this practical course, student will be able to:

1	Implement C# class hierarchies, constructors, inheritance, method overriding, exception handling, and generic collections.	K1, K2
2	Develop C# applications using delegates, events, lambda expressions, and LINQ for data querying and manipulation.	K2, K3
3	Build Windows Forms desktop applications with event-driven programming and complex UI controls.	K3, K4
4	Implement ADO.NET database connectivity and Entity	K4, K5

	Framework Core with Code First approach and migrations.	
5	Develop ASP.NET Core MVC web applications and RESTful Web APIs with Swagger documentation.	K5, K6
K1 - Remember; K2 - Understand; K3 - Apply; K4 - Analyze; K5 - Evaluate; K6 - Create		

Mapping of Course Outcomes (COs) with Program Outcomes (POs):

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	2	2	1	1	2
CO2	2	3	2	1	1	2
CO3	3	3	2	2	1	2
CO4	2	2	3	2	1	3
CO5	3	3	3	2	2	3

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Mapping of Course Outcomes (COs) with Program Specific Outcomes (PSOs):

CO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	2	1	1	2
CO2	2	3	1	1	2
CO3	3	3	2	1	2
CO4	2	2	2	2	3
CO5	3	3	2	2	3

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

SEMESTER – III
CORE – XVI – INDUSTRY LITERACY / MINI PROJECT

Learning Objective (LO):
To apply software development skills to design, develop, test, and present a complete mini project solving a real-world industry problem.

List of Programs:

01. Industry Problem Statement Analysis: Study and document a real-world industry problem from sectors such as healthcare, finance, retail, or education and prepare a formal problem statement.
02. Software Requirements Specification (SRS): Prepare a complete SRS document including scope, functional and non-functional requirements, use case diagrams, and user stories.
03. System Architecture Design: Design the system architecture using UML diagrams – Use Case Diagram, Class Diagram, Sequence Diagram, and Activity Diagram.
04. Database Design: Design the ER diagram, normalize tables to 3NF, and create the complete database schema with DDL scripts.
05. Technology Stack Selection and Justification: Evaluate and document the selection of programming languages, frameworks, databases, cloud platforms, and development tools with technical justification.
06. UI/UX Prototype Design: Create low-fidelity wireframes and a high-fidelity interactive prototype using Figma for all major application screens.
07. Backend Module Development: Implement at least three core backend modules (e.g., user authentication, business logic, data management) with proper API design.
08. Frontend Implementation: Implement responsive frontend screens connected to the backend using HTML/CSS/JS or a frontend framework with proper form validation.
09. Integration Testing and Bug Fixing: Integrate all modules, perform integration testing with documented test cases, track and fix bugs, and maintain a test log.
10. Final Project Demonstration and Viva: Present the completed mini project with a live demonstration, submit the project report, and defend the project design and implementation.

Expected Course Outcomes:
On the successful completion of this practical course, student will be able to:

1	Identify and analyse real-world industry problems and prepare formal Software Requirements Specification (SRS) documents.	K1, K2
2	Design system architecture, database schemas, UML diagrams, and UI/UX prototypes for the proposed software project.	K2, K3
3	Develop functional backend and frontend modules of a software application using current industry	K3, K4

	technologies.	
4	Integrate system components, perform systematic testing, document test results, and track and resolve defects.	K4, K5
5	Present and demonstrate a complete working software project with technical documentation and formal viva-voce defence.	K5, K6
K1 - Remember; K2 - Understand; K3 - Apply; K4 - Analyze; K5 - Evaluate; K6 - Create		

Mapping of Course Outcomes (COs) with Program Outcomes (POs):

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	2	2	1	1	2
CO2	2	3	2	1	1	2
CO3	3	3	2	2	1	2
CO4	2	2	3	2	1	3
CO5	3	3	3	2	2	3

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Mapping of Course Outcomes (COs) with Program Specific Outcomes (PSOs):

CO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	2	1	1	2
CO2	2	3	1	1	2
CO3	3	3	2	1	2
CO4	2	2	2	2	3
CO5	3	3	2	2	3

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

SEMESTER – IV
CORE – XVII - PROJECT WORK AND VIVA-VOCE

Project Course Outcomes:

S.No	Course Outcomes	Bloom's Level
1	Formulate a real-world problem statement, conduct a feasibility study, and design an appropriate software solution architecture with UML diagrams.	K1, K2
2	Apply the complete software development lifecycle (SDLC) including requirements analysis (SRS), system design, and relational database design.	K2, K3
3	Develop a functional software application implementing industry-standard technologies, frameworks, and software engineering methodologies.	K3, K4
4	Conduct systematic unit testing, integration testing, and user acceptance testing, documenting test cases and bug resolution.	K4, K5
5	Produce professional project documentation following prescribed format and deliver a formal technical presentation and viva-voce defence.	K5, K6

K1 - Remember; K2 - Understand; K3 - Apply; K4 - Analyze; K5 - Evaluate; K6 - Create

Mapping of Course Outcomes (COs) with Program Outcomes (POs):

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	2	2	1	1	2
CO2	2	3	2	1	1	2
CO3	3	3	2	2	1	2
CO4	2	2	3	2	1	3
CO5	3	3	3	2	2	3

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Mapping of Course Outcomes (COs) with Program Specific Outcomes (PSOs):

CO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	2	1	1	2
CO2	2	3	1	1	2
CO3	3	3	2	1	2
CO4	2	2	2	2	3
CO5	3	3	2	2	3

Mapping Scale: 3 = Strong / High; 2 = Moderate / Medium; 1 = Substantial / Low.

Assessment of Project Work:

Component	Marks
CIA – Review I (Internal)	25 Marks
CIA – Review II (Internal)	25 Marks
CIA – Progress Report and Attendance	50 Marks
CIA Total	100 Marks
EA – Project Evaluation (External Examiner)	225 Marks
EA – Viva-Voce (Internal + External)	75 Marks

Guidelines for Project Work:

1. Each student shall independently design, develop, and document a substantial software project under the guidance of an internal guide.
2. The student shall submit the filled-in project approval format to the Department during the first week of Semester IV.
3. The project shall be periodically reviewed: Review I (25 marks) at the end of the first month and Review II (25 marks) at the end of the second month.
4. The student shall submit three hardbound copies of the project report following the prescribed University format.
5. Students may use a presentation during the Viva-Voce examination conducted jointly by internal and external examiners.
6. The project report must be original work; plagiarism will be dealt with as per university norms.