SNDT Women's University, Mumbai

Postgraduate Degree / PG Programme

Faculty of Science & Technology

Master of Computer Applications (MCA)

Semester – I to IV

Syllabus

(W.E.F. Academic Year 2026-27)

Programme	MasterofComputer Applications(MCA)
Preamble	In unwavering commitment to the principles outlined herein, the MCAprogramsteadfastlyaimsto shape an educational experience that empowers itsstudents to excel as adept scholars, critical thinkers, and responsible leaders within their respective fields. By articulating these guidingprinciples, we underscore our dedication to fostering a transformative learning environment that goes beyond the acquisitionoftechnicalskillstocultivateholisticand forward-thinking professionals.
Programme Outcomes(POs)	Aftercompletingthisprogramme,Learnerwillbeableto Upon successful completion of the MCA programme, graduates will be able to: 1. To apply knowledge of computing fundamentals, mathematics, programming, and domain-specific concepts to develop appropriate computing solutions. 2. To identify, formulate, analyze, and solve complex computing problems using principles of computer science, mathematics, and logical reasoning. 3. To design, develop, test, and evaluate software systems, applications, and computing solutions that meet user requirements while considering societal, environmental, and ethical aspects. 4. To apply research methods, experimentation, data analysis, and interpretation techniques to investigate and solve complex computing problems. 5. To select and apply appropriate modern computing tools, technologies, frameworks, and methodologies for software development and problem-solving. 6. To understand and apply ethical principles, professional responsibilities, cyber laws, and societal considerations in computing practices. 7. To evaluate the impact of computing solutions on society and the environment and contribute to sustainable development through responsible use of technology. 8. To function effectively as an individual, team member, or team leader in multidisciplinary and diverse professional environments. 9. To communicate effectively with technical and non-technical stakeholders through reports, documentation, presentations, and interpersonal interactions. 10. To apply project management, financial, and organizational principles to manage software projects and computing

	initiatives efficiently. 11. To demonstrate innovation, creativity, and entrepreneurial skills to develop technology-driven solutions and business opportunities.
Programme Specific Outcomes (PSOs)	Programme Specific Outcomes (PSOs) for MCAspecifytheparticularskills,knowledge, and abilities that students are expected to gain upon completion of the program. 1. Professionally skilled and trained in the field of computer science, they can solve complex, real-time problems, which help them grow personally and professionally. 2. Understanding modern computer technologies and their applications to solve complex and critical issues that benefit society and the environment. 3. Trained in performing effectively as an individual, a team, and as a team leader in a multidisciplinary environment using critical thinking skills.
EligibilityCriteriafor Programme	The candidate,- (i) should be a citizen of India; (ii) should have passed any graduation degree (e.g.: B.E. or B.Tech. or B.Sc or B.Com. or B.A. or B. Voc. or BCA etc.,) preferably with Mathematics at 10+2 level or at Graduation level and obtained at least fifty per cent marks (at least forty five per cent. in case of candidates of Reserved Categories, Economically Weaker Section and Persons with Disability category belonging to the Maharashtra State) in the qualifying examination; iii) should have obtained non zero score in MAH-MCA-CET conducted by the Competent Authority for that year.
Intake	60
Duration	4semesters(2years)

Mode : MCA lectures and practical shall be conducted on regular basis as per the schedule.

Attendance: The 75% attendance is mandatory for the student to appear for the university examination.

Evaluation:
External Assessment: University Examination (Semester Pattern) to be conducted in December /May every year as per university norms.

Internal Assessment: Internal evaluation shall be conducted through unit tests, case studies, seminars, project presentations, assignments, and submissions.

MCA SEMESTER-I

Code	Subject	L	Pr.	Cr.	Int. Exam.	Ext. Exam.	Total Marks
1301	**Mathematical Foundation of Computer Science**	4	-	4	50	50	100
1302	**Research Methodology**	4	-	4	50	50	100
1303	**Artificial Intelligence**	2	-	2	-	50	50
1304	**Advanced Java (Contents: spring, MVC, Net Beans and Hibernate)**	2	-	2	-	50	50
1305	**Advanced Databases**	2	-	2	-	50	50
1403	**Artificial Intelligence Lab**	-	4	2	25	25	50
1404	**Advanced Java Lab**	-	4	2	25	25	50
1405	**Advanced Databases Lab**		4	2	25	25	50
1406	**Mini Project**	-	8	4	50	50	100
	Total			**24**			**600**

MCA SEMESTER-II

Code	Subject	L	Pr.	Cr.	Int. Exam.	Ext. Exam.	Total Marks
2301	**Enterprise Architecture**	2	-	2	-	50	50
2302	**Introduction to Data Science**	2	-	2	-	50	50
2303	**Advanced Data Structure**	2	-	2	-	50	50
2304	**Computer Network Programming using Linux and Java**	2	-	2	-	50	50
	Elective-I	2	-	2	-	50	50
2402	**Introduction to Data Science Lab**	-	4	2	25	25	50
2401	**Enterprise Architecture Lab**	-	4	2	25	25	50
2403	**Advanced Data Structure Lab**	-	4	2	25	25	50
2404	**Computer Network Programming using Linux and Java Lab**	-	4	2	25	25	50
	Elective-I Lab	-	4	2	25	25	50
2406	**Mini Project**	-	8	4	50	50	100
	Total			**24**			**600**

MCA SEMESTER-III

Code	Subject	L	Pr.	Cr.	Int. Exam.	Ext. Exam.	Total Marks
3301	**Cyber Security**	4	-	4	50	50	100
3302	**Mobile Application Development Technologies**	2	-	2	-	50	50
3303	**Cloud Computing**	2	-	2	-	50	50
3304	**Machine learning**	2	-	2	-	50	50
	Elective-II	2	-	2	-	50	50
3402	**Mobile Application Development Technologies Lab**	-	4	2	25	25	50
3403	**Cloud Computing Lab**	-	4	2	25	25	50
3404	**Machine learning Lab**	-	4	2	25	25	50
	Elective-II Lab	-	4	2	25	25	50
3406	**Mini Project**	-	8	4	50	50	100
	Total			**24**			**600**

MCA SEMESTER-IV

Code	Subject	L	P/T	Cr	Int. Exam.	Ext. Exam.	Total
Swayam Code	**Swayam based MOOC**	-	-	4	-	100	100
4401	**Internship**	-	-	20	250	250	500
	Total			**24**			**600**

- **MOOC list will be provided at the beginning of the semester**

Code	Elective-I (Semester II)	Code	Elective-II (Semester III)
2311 2411 (Lab)	Generative AI	3321 3421 (Lab)	Agentic AI
2312 2412 (Lab)	Social Network Analysis	3322 3422(Lab)	Big Data Analytics
2313 2413 (Lab)	Internet Of Things	3323 3423 (Lab)	Data Security and Privacy
2314 2414 (Lab)	Block Chain	3324 3424 (Lab)	Computer Vision
2315 2415 (Lab)	Business Intelligence and Analytics	3325 3425 (Lab)	Design for Scalability/Reliability

SEMESTER CREDITS				1 Credit (Practical)=2 Hrs 1 Credit (Tutorial)=2 Hrs 1Credit (Theory)=1 Hr 1 Credit = 25 marks Total Credits=96 Total Marks=96*25=2400
I	II	III	IV	
24	24	24	24	

MCA SEMESTER-I

CourseCode	**1.1**
CourseTitle	**Mathematical Foundation of Computer Science**
CourseCredits	**4**
CourseOutcomes	**AfterCompletionofthisCourse,studentswillbeableto** 1. To understand the basic notions of Induction and state machine. 2. To understand graphs and its applications in Computer Science. 3. To understand the methods of statistical inference and the role of probability in computer science. 4. To be able to perform correct and meaningful statistical analysis of simple to moderate complexity
Module1:Mathematical Induction and Recursive Structures(Credit 1)	
LearningOutcomes	**Afterlearningthismodule,learnerswillbeableto** 1. Apply ordinary and strong induction techniques to prove mathematical statements. 2. Analyze state machines using states, transitions, and invariants. 3. Evaluate program correctness and termination using inductive reasoning. 4. Understand recursive data types and recursive definitions. 5. Use structural induction to verify properties of recursive structures. 6. Apply recursion and induction to strings, expressions, search trees, and computer science problems.
ContentOutline	Induction: Ordinary Induction, Strong Induction. State Machines: States and Transitions, The Invariant Principle, Partial Correctness & Termination. Recursive Data Types, Recursive Definitions and Structural Induction, Strings of Matched Brackets, Recursive Functions on Nonnegative Integers, Arithmetic Expressions, Games as a Recursive Data Type, Search Trees, Induction in Computer Science.
Module2:Graph Theory(Credit 1)	
LearningOutcomes	**Afterlearningthismodule,learnerswillbeableto** 1. Understand the fundamental concepts of graph theory and graph representations. 2. Analyze graph properties such as connectivity, distance, diameter, and isomorphism. 3. Apply trees and spanning trees to solve computational and network problems. 4. Use graph coloring and planar graph concepts in optimization problems.

	5. Model and analyze real-world systems using directed graphs and partial orders.
ContentOutline	Graph Theory, Graph Variations, Applications of Graphs, Some Common Graphs, Isomorphism, Connectivity, A Simple Connectivity Theorem, Distance and Diameter, Adjacency. Matrices. Trees, Spanning Trees, Tree Variations, Graph Theory: Coloring Graphs, kColoring, Planar Graphs, Euler's Formula, Directed graphs & Partial Orders
Module3: Statistical Methods and Probability(Credit 1)	
LearningOutcomes	**Afterlearningthismodule,learnerswillbeableto** 1. Understand data types, data collection methods, and statistical studies. 2. Apply descriptive statistical techniques using graphical and numerical measures. 3. Analyze multivariate statistical models for regression and classification problems. 4. Evaluate model performance and understand overfitting in data analysis. 5. Apply probability concepts, conditional probability, and Bayes' theorem to solve computational problems. 6. Use discrete and continuous probability distributions for modeling real-world scenarios.
ContentOutline	Introduction to data: Types of variables; data collection principles; types of studies. Graphical methods: histograms and other graphs Frequency distribution, measures of central tendency, measures of dispersion, Descriptive Statistics Statistical inference, Introduction to multivariate statistical models: regression and classification problems, principal components analysis, The problem of over fitting model assessment. Elementary probability theory, Conditional probability, Bayes theorem, Random variable Expected values, Discrete Probability distributions: Uniform, Binomial, Poisson, distribution. Continuous distributions: Uniform, Exponential, Normal, Relationship between Poisson and exponential distribution.
Module4: Recent Trends in Distribution Functions (Credit 1)	
LearningOutcomes	**Afterlearningthismodule,learnerswillbeableto** 1. Understand the basics of probability distribution functions. 2. Identify different distribution models used in computer science. 3. Apply distribution functions in data analysis, AI, networking,

	and cybersecurity. 4. Analyze recent trends in probabilistic and statistical modeling. 5. Evaluate the role of distribution functions in emerging technologies such as Big Data and IoT.
ContentOutline	Recent Trends in various distribution functions in mathematical field of computer science for varying fields.

Evaluation Scheme:

Internal Assessment:

Depending on the course content mentioned above a case study/assignments/presentations should be developed of 25 marks. Along with a Unit test of 25 marks should be conducted.

External Assessment: End Semester examination of 50 marks for 2 hours duration will be conducted by the university.

Reference Books:

1. John Vince, Foundation Mathematics for Computer Science, Springer.
2. K. Trivedi. Probability and Statistics with Reliability, Queuing, and Computer Science Applications. Wiley.
3. M. Mitzenmacher and E. Upfal.Probability and Computing: Randomized Algorithms and Probabilistic Analysis.
4. Alan Tucker, Applied Combinatorics, Wiley

CourseCode	**1.2**
CourseTitle	**Research Methodology**
CourseCredits	**4**
CourseOutcomes	**AfterCompletionofthisCourse,studentswillbeableto** 1. Demonstrate an understanding of the key steps involved in the research process, including problem formulation, literature review, data collection, analysis, and interpretation. 2. Formulate clear and focused research questions and hypotheses based on a thorough review of existing literature and identification of research gaps. 3. Conduct comprehensive literature reviews to identify relevant studies, theories, and methodologies within a specific research domain. 4. Apply basic statistical techniques for data analysis, interpretation, and drawing meaningful conclusions from research findings 5. To become aware about current trends in IPR and Govt. steps in fostering IPR.
Module1:Introduction to Research methodology(Credit 1)	
LearningOutcomes	**Afterlearningthismodule,learnerswillbeableto** 1. Understand the fundamentals of research – Explain the meaning, objectives, and significance of research in various disciplines. 2. Identify and define research problems – Formulate clear and researchable problems and hypotheses based on real-world issues. 3. Conduct literature reviews – Search, analyze, and synthesize existing research to support new investigations. 4. Apply appropriate data collection methods – Use surveys, interviews, observations, and experiments effectively in gathering primary data. 5. Differentiate between qualitative and quantitative research – Understand the applications, strengths, and limitations of both approaches.
ContentOutline	**Research methodology** An Introduction Objectives of Research, Types of Research, Research Methods and Methodology, defining a Research Problem, Techniques Involved in Defining a Problem. **Research Design**

	Need for Research Design, Features of Good Design, Different Research Designs, Basic Principles of Experimental Designs, Sampling Design, Steps in Sampling Design, Types of Sampling Design, Sampling Fundamentals, Estimation, Sample size Determination, Random sampling.
Module2:Measurement and Scaling Techniques(Credit 1)	
LearningOutcomes	**Afterlearningthismodule,learnerswillbeableto** 1. Understand the concepts of measurement in research – Explain the meaning, importance, and levels of measurement (nominal, ordinal, interval, and ratio). 2. Understand the principles of scaling – Explain the purpose and use of different scaling techniques in behavioral and social science research. 3. Interpret scaled data effectively – Analyze and draw conclusions from data collected through various scaling methods.
ContentOutline	**Measurement in Research** Measurement Scales, Sources in Error, Techniques of Developing Measurement Tools, Scaling, Meaning of Scale, Scale Construction Techniques **Methods of Data Collection and Analysis** Collection of Primary and Secondary Data, Selection of appropriate method Data Processing Operations, Elements of Analysis, Statistics in Research, Measures of Dispersion, Measures of Skewness, Regression Analysis, Correlation.
Module3: Techniques of Hypotheses(Credit 1)	
LearningOutcomes	**Afterlearningthismodule,learnerswillbeableto** 1. Understand the concept of a hypothesis – Define hypothesis, its role, and importance in the research process. 2. Formulate testable hypotheses – Develop clear, specific, and researchable hypotheses based on the research problem and literature review. 3. Select appropriate statistical tests – Choose suitable parametric or non-parametric tests based on the type of data and research design.
ContentOutline	**Techniques of Hypotheses** Parametric or Standard Tests Basic concepts, Tests for Hypotheses I and II, Important parameters limitations of the tests of Hypotheses, Chi-square Test, Comparing Variance, as a non-parametric Test, Conversion of ChitoPhi, Caution in using Chi-square test.

Module4: IPR Fundamentals(Credit 1)	
LearningOutcomes	**Afterlearningthismodule,learnerswillbeableto** 1. Understand the fundamental concepts, scope, and importance of Intellectual Property Rights along with their legal and philosophical foundations. 2. Apply basic principles of Patent, Copyright, Trademark, and Design laws, including procedures for registration and protection. 3. Develop the ability to analyze and draft basic patent specifications and identify different forms of intellectual property protection. 4. Recognize the role of IPR in the Information Technology sector and evaluate issues related to infringement, digital rights, and international IPR frameworks.
ContentOutline	**INTELLECTUAL PROPERTY RIGHTS** Basic Principles and Acquisition of Intellectual Property Rights:Philosophical Aspects of Intellectual Property Laws, Basic Principles of Patent Law, Patent Application procedure, drafting of a Patent Specification, Understanding Copyright Law, Basic Principles of Trade Mark, Basic Principles of Design Rights, International Background of Intellectual Property Information Technology Related Intellectual Property Rights.

Evaluation Scheme:

Internal Assessment:

Depending on the course content mentioned above a case study/assignments/presentations should be developed of 25 marks. Along with a Unit test of 25 marks should be conducted.

External Assessment: End Semester examination of 50 marks for 2 hours duration will be conducted by the university.

References:

1. Kothari, C. R., & Garg, G. (2019). Research methodology: Methods and techniques (4th ed.). New Age International Publishers.

2. Kumar, R. (2019). Research methodology: A step-by-step guide for beginners (5th ed.). SAGE Publications.

3. Cooper, D. R., & Schindler, P. S. (2014). Business research methods (12th ed.). McGraw-Hill Education.

4. Creswell, J. W., & Creswell, J. D. (2018). Research design: Qualitative, quantitative, and mixed methods approaches (5th ed.). SAGE Publications.

5. Zikmund, W. G., Babin, B. J., Carr, J. C., & Griffin, M. (2012). Business research methods (9th ed.). Cengage Learning.

6. Krishnaswamy, K. N., Sivakumar, A. I., & Mathirajan, M. (2009). Management research methodology: Integration of principles, methods and techniques. Pearson Education India.

7. Dawson, C. (2002). Practical research methods: A user-friendly guide to mastering research techniques and projects. How to Books.

CourseCode	**1.3**
CourseTitle	**ArtificialIntelligence**
CourseCredits	**2**
CourseOutcomes	**AfterCompletionofthisCourse,studentswillbeableto** 1. Understand the fundamental concepts, history, and evolution of Artificial Intelligence, including key domains and applications. 2. Apply problem-solving techniques such as search algorithms (uninformed and informed searches) and game playing strategies for developing intelligent solutions. 3. Demonstrate the ability to implement knowledge representation methods like logic-based models, semantic networks, and ontologies for AI problem domains.
Module1:Introduction to Artificial Intelligence(Credit 0.5)	
LearningOutcomes	**Afterlearningthismodule,learnerswillbeableto** 1. Define Artificial Intelligence and explain its scope, history, and evolution. 2. Identify key areas and applications of AI across various industries like healthcare, education, finance, and automation. 3. Illustrate real-world examples of AI systems and discuss how they improve efficiency and decision-making.
ContentOutline	**Definition, History & Evolution of AI** Types of AI: Narrow AI, General AI, Super AI Applications of AI in various sectors **Intelligent Agents:** Types, Structures, and Environments Solving by searching: Travelling salesman Problem, Monkey Banana Problem, Tower of Hanoi and Block World.
Module2:Problem solving by Searching Methods(Credit 0.5)	
LearningOutcomes	**Afterlearningthismodule,learnerswillbeableto** 1. Explain the role of search in problem-solving within Artificial Intelligence. 2. Differentiate between various search strategies, including uninformed (blind) and informed (heuristic) search techniques. 3. Apply informed search techniques like Greedy Best-First Search and A* Search for efficient problem-solving.
ContentOutline	**Uninformed Search Techniques** Breadth-First Search (BFS) Depth-First Search (DFS) Iterative Deepening Search

	Informed Search Techniques Best-First Search A* Search Hill Climbing **Adversarial Search** Game Playing: Minimax Algorithm Alpha-Beta Pruning
Module3: Knowledge Representation and Reasoning(Credit 0.5)	
LearningOutcomes	**Afterlearningthismodule,learnerswillbeableto** 1. Describe the importance of knowledge representation in building intelligent systems. 2. Apply logic-based methods (Propositional and Predicate Logic) to represent facts, rules, and relationships in problem domains. 3. Implement reasoning techniques such as forward chaining, backward chaining, and resolution to infer new knowledge from existing information.
ContentOutline	**Knowledge Representation Techniques** Propositional Logic Predicate Logic Semantic Networks Frames and Ontologies **Reasoning Methods** Forward Chaining Backward Chaining Resolution **Reasoning under Uncertainty** Bayesian Networks Fuzzy Logic and Reasoning
Module4: Knowledge Representation(Credit 0.5)	
LearningOutcomes	**Afterlearningthismodule,learnerswillbeableto** **1.** Explain various knowledge representation techniques such as logic-based models, semantic networks, frames, and ontologies for structuring information in AI systems. **2.** Describe the architecture and components of Expert Systems, including knowledge base, inference engine, and user interface. **3.** Identify challenges and limitations in representing knowledge, reasoning under uncertainty, and processing natural language.
ContentOutline	**Structured Knowledge Representation:**

	Associative Networks, Semantic Nets, Frames Structures, Conceptual, Dependencies & Scripts, Learning – Concept of Learning – Learning Automata, Learning by induction. **Natural Language Processing:** Overview of Linguistics, Grammars and Languages, basic Parsing techniques, semantic analysis, and representation structures. Natural Language generation and Natural Language Systems. **Expert System:** Knowledge acquisition methods, knowledge engineering process, goals in knowledge system development, basic architecture of expert systems, problem domain versus knowledge domain, Development of ES and life cycle of ES. Advantages of expert system, structure of Rule based expert system, characteristics of conventional system and expert system

Evaluation Scheme:

External Assessment: End Semester examination of 50 marks for 2 hours duration will be conducted by the university.

References:

1. Introduction to AI and Expert Systems - Patterson.

2. Artificial Intelligence A Modern Approach- Stuart Rushell.

3. Artificial Intelligence - Rich E and Knight K.

4. Principles of Artificial Intelligence - Nilsson.

5. Artificial Intelligence – An Engineering Approach – Schalkoff R J.

6. Introduction to Expert System - Peter Jackson.

7. Artificial Intelligence – Janakiraman.

CourseCode	**1.4**
CourseTitle	**Advanced Java (Contents: spring, MVC, Net Beans and Hibernate)**
CourseCredits	**2**
CourseOutcomes	**AfterCompletionofthisCourse,studentswillbeableto** 1. Understand the advance web concepts associated with JAVA. 2. Familiarize with hibernate. 3. Familiarize with the MVC architecture. 4. Familiarize with the Spring framework.
Module1:Core Java and JDBC(Credit 0.5)	
LearningOutcomes	**Afterlearningthismodule,learnerswillbeableto** 1. Explain Java architecture, JVM, JRE, and object-oriented programming concepts. 2. Develop GUI and event-driven applications using AWT and Swing. 3. Implement database connectivity and transaction management using JDBC.
ContentOutline	**Introduction:** History, architecture and its components, Java Class File, Java Runtime Environment, The Java Virtual Machine, JVM Components, The Java API, java platform, java development kit, Lambda Expressions, Methods References, Type Annotations, Method Parameter Reflection. Object Oriented Programming, packages, enumerations, Multi-threading, Exception Handling. Abstract Window Toolkit: Window Fundamentals, Component, Container, Panel, Window, Frame, Canvas. Components – Labels, Buttons, Check Boxes, Radio Buttons, Choice Menus, Text Fields, Text, Scrolling List, Scrollbars, Panels, Frames, JAVA adapter classes. Swing components. AWT vs Swings **Introduction to Event Handling:** Identifying the source of Event, Event Listeners and Event Handlers, the Delegation Event Model, Event classes, Event Listener Interface, Action Listener interface, Mouse Listener Interface Adapter classes-the Mouse Adapter class, the Mouse Motion Listener Interface. **Introduction to JDBC:** What is JDBC. Database connectivity, JDBC Architecture, JDBC drivers, Using JDBC API – Loading a Driver, connecting and executing JDBC statement, Handling SQL Exceptions. Accessing Result Sets, method of Result Set interface, Methods of Prepared Statement interface, retrieving row, inserting row, Managing Database Transactions, creating and calling stored procedures in JDBC, using

	Metadata in JDBC.
Module2:Servlets, JSP and Web Technologies(Credit 0.5)	
LearningOutcomes	**Afterlearningthismodule,learnerswillbeableto** 1. Develop web applications using Servlets and JSP. 2. Implement session management and request handling techniques. 3. Use JavaScript, JSON, and jQuery to create interactive web interfaces.
ContentOutline	**Introduction to servlets:** Servlet vs CGI, Servlet API overview, Servlet Life cycle, Generic servlet, HTTP Servlet, ServletConfig, Servlet Context, Handling HTTP Request and response –GET / POST method, request dispatching, Using cookies, Session tracking. **Web development using JSP:** Introduction to JSP, JSP Architecture, JSP Directives, JSP scripting elements, Default objects in JSP, JSP Actions, JSP with beans and JSP with Database, Error handling in JSP, tracking techniques in JSP, Introduction to custom tags, JSTL tags in detail. Introduction to jQuery, JS, JS JSON, jQuery vs JS
Module3: Spring Framework (Credit 0.5)	
LearningOutcomes	**Afterlearningthismodule,learnerswillbeableto** 1. Explain Spring Framework architecture and AOP concepts. 2. Develop applications using Spring MVC and Spring Boot. 3. Create secure RESTful web services and integrate JMS messaging.
ContentOutline	**Introduction to Spring Framework:** Spring Architecture, Spring Aspect of Object-Oriented Concepts – Join Point and Point Cuts. Spring web applications with Spring MVC. Features of the Spring Boot. Use of Spring Boot to create and configure a Spring application. Customize Spring Boot features. REST web services with Spring. Spring Data Secure with Spring Security. JMS-Introduction, requirement, JMS Programming model. JMS support of Spring.
Module4: Hibernate (Credit 0.5)	
LearningOutcomes	**Afterlearningthismodule,learnerswillbeableto** 1. Explain ORM concepts and Hibernate architecture. 2. Develop persistent applications using Hibernate mappings and transactions.

	3. Perform efficient database operations using HQL, Criteria API, caching, and fetching techniques.
ContentOutline	**Introduction to Hibernate:** Problem with JDBC - paradigm mismatch, ORM. Different components of Hibernate, How to persist objects using Hibernate, mapping files in hibernate, configuration files and Session object, Instance states, Implementing Inheritance in Hibernate, Transactions in Hibernate, Querying with HQL (Hibernate Query Language), Named and native queries, Working with Criteria Interface, Query by example – QBE, Caching and fetching.

Evaluation Scheme:

External Assessment: End Semester examination of 50 marks for 2 hours duration will be conducted by the university.

Reference Books:

1. The Complete Reference, Third Edition, by Patrick Naughton and Herbert Schildt, Tata McGraw Hill Edition 1999.

2. Java Enterprise in a Nutshell: A Desktop Quick Reference (Nutshell Handbook) or any other book with similar contents.

3. Mastering Java2 J2SE1.4 by John Zukouski PBP Publication

4. JavaTM How to Program Sixth Edition by H.M Deitel, P.J. Deitel

5. Core Servlets & JavaServer Pages by Marty Hall, Larry Brown

6. Spring Boot in Action 1st Edition by Craig Walls

7. Beginning Hibernate Second Edition by Jeff Linwood, Dave Minter is the third book for Hibernate beginners.

CourseCode	**1.5**
CourseTitle	**Advanced Databases**
CourseCredits	**2**
CourseOutcomes	**AfterCompletionofthisCourse,studentswillbeableto** 1. Students will grasp the theoretical foundations of advanced database systems, including object-oriented databases, parallel databases, and distributed databases. 2. Students will be able to understand the basic concepts of transaction management, including various concurrency control protocols and recovery mechanisms. 3. Students will be able to analyze the architecture and design of distributed and parallel databases, and understand data distribution, query processing, and system coordination. 4. Students will be able to differentiate NoSQL databases from traditional RDBMS, and explain the need for NoSQL in handling big data, horizontal scalability, and schema flexibility.
Module1:Transaction Management and Concurrency Control(Credit 0.5)	
LearningOutcomes	**Afterlearningthismodule,learnerswillbeableto** 1. Understand the concept of Transactions. 2. Explain the need for concurrency control in multiuser database environments. Understand and apply locking protocol. 3. Determine the serializability of transaction schedules using conflict and view serializability
ContentOutline	**Introduction to Transaction: -** Define the purpose of Transaction, ACID property, Transaction state. **Transaction Processing: -** Transaction life cycle, System and user-define transaction, s vs concurrent execution **Concurrency Control: -** Need for concurrency in DBMS, Problems in concurrency, S and their type. **Serializability: -** Conflict serializability, view serializability. **Concurrency Control Protocol: -** Lock-based protocol Binary locks Shared and exclusive locks, Two-phase locking (2PL) protocol, Strict 2PL and rigorous 2PL,Timestamp-based protocol.
Module2:Introduction to Advance Database Model(Credit 0.5)	

LearningOutcomes	**Afterlearningthismodule,learnerswillbeableto** 1. Compare and contrast advanced data models with traditional relational models. 2. Develop solutions involving object-oriented, temporal, or spatial data using relevant tools or extensions. 3. Analyze use-cases and select appropriate advanced database models for different application domains
ContentOutline	**Introduction to Object-Oriented Databases: -** Features of OODB, Object Oriented Data Models (e.g. ODMG, ORDBMS extensions **Introduction to Temporial Databases: -** Difference between conventional databases and temporal databases. Types of temporal data: valid time, transaction time, bitemporal, temporal Data Model – time-stamped tuples, attribute-version, Event-based models. **Introduction to Spatial Databases: -** Definition and importance of spatial databases, Applications: GIS, urban planning, environmental monitoring, LBS (location-based services), Spatial data types: Points, Lines, Polygons,Raster vs. Vector data models **Introduction to Deductive Databases: -** Definition and motivation, Differences between relational and deductive databases ,Applications of deductive databases (rule-based systems, expert systems, knowledge bases)
Module3: Distributed and Parallel Databases(Credit 0.5)	
LearningOutcomes	**Afterlearningthismodule,learnerswillbeableto** 1. Understand the need for distributed and parallel database systems. 2. Differentiate between centralized, distributed, and parallel database systems. 3. Describe the architectures of distributed and parallel database systems and identify the challenges and trade-offs associated with distributed and parallel database systems.
ContentOutline	**Introduction to Distributed Databases:-** Centralized vs. Distributed vs. Parallel Systems. Need for distributed and parallel databases. Performance metrics (e.g., speedup, scale-up). Basic concepts: fragmentation, replication, and distribution. **Distributed Database Architectures:-** Shared-nothing, shared-disk, and shared-memory architectures. Examples of distributed database systems. **Introduction to Distributed Database Design: -** Data fragmentation

	(horizontal, vertical, derived). Data allocation strategies.
Module4: NoSQL Databases(Credit 0.5)	
LearningOutcomes	**Afterlearningthismodule,learnerswillbeableto** **1.** Explain the differences between NoSQL and traditional relational databases. **2.** Describe the need for NoSQL in handling big data, scalability, and flexible schemas. **3.** Understand BASE properties and how they differ from ACID properties
ContentOutline	**Introduction to NoSQL :-** What is NoSQL? , Comparison with relational databases, **Characteristics of NoSQL databases:-** Schema-free design, Horizontal scalability, BASE vs. ACID properties **Types of NoSQL Data Models:-** Classification of NoSQL databases: Key-Value Stores (e.g., Redis, Riak); Document Stores (e.g., MongoDB, CouchDB); Column-Family Stores (e.g., Cassandra, HBase) ;Graph Databases (e.g., Neo4j)

Evaluation Scheme:

External Assessment: End Semester examination of 50 marks for 2 hours duration will be conducted by the university.

References:-

1. “Database System Concepts” Authors: Abraham Silberschatz, Henry F. Korth, and S. Sudarsha

n, Publisher: McGraw-Hill

2. “Fundamentals of Database Systems” Authors: Ramez Elmasri and Shamkant B. Navathe, Publisher: Pearson

3. “Advanced Database Management System” Author: Dr. Rini Chakravarty Publisher: Dreamtech Press

4. NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence” Authors: Pramod J. Sadalage and Martin Fowler, Publisher: Addison-Wesley

5. Principles of Distributed Database Systems Authors: M. Tamer Özsu and Patrick Valduriez Publisher: Springer

Course Code	**1.6**
Course Title	**Artificial Intelligence Lab**
Course Credits	**2**

A course-based project relevant to the concepts covered in the subject shall be implemented and evaluated. The project should demonstrate the practical application of the techniques, and methodologies learned in the course using appropriate programming languages, tools, and technologies.

Evaluation Scheme:

Internal Evaluation (25 marks):

The project should demonstrate the application of theoretical knowledge to solve a real-world problem through system design, implementation and performance evaluation.

The Internal Assessment for the project shall be conducted based on project implementation, research contribution, project presentation, and report writing, as per the following rubric. Evaluation done at Institute level.

The Rubric will have the following Evaluation Parameters:

The Rubric will have the following Evaluation Parameters:	
Evaluation Parameters	**Marks**
Problem Understanding & Formulation	10
Implementation & Functionality	20
Knowledge Representation / Logic Design	10
Analysis & Result Interpretation	5
Report writing or Documentation	5

External Examination – 25 marks Presentation (internal and external examiners involved)

Course Code	**1.7**
Course Title	**Advanced Java Lab**
Course Credits	**2**

A course-based project relevant to the concepts covered in the subject shall be implemented and evaluated. The project should demonstrate the practical application of the techniques, and methodologies learned in the course using appropriate programming languages, tools, and technologies.

Evaluation Scheme:

Internal Evaluation (25 marks):

The project should demonstrate the application of theoretical knowledge to solve a real-world problem through system design, implementation and performance evaluation.

The Internal Assessment for the project shall be conducted based on project implementation, research contribution, project presentation, and report writing, as per the following rubric. Evaluation done at Institute level.

The Rubric will have the following Evaluation Parameters:

The Rubric will have the following Evaluation Parameters:	
Evaluation Parameters	**Marks**
Problem Understanding & Formulation	10
Implementation & Functionality	20
Knowledge Representation / Logic Design	10
Analysis & Result Interpretation	5
Report writing or Documentation	5

External Examination – 25 marks Presentation (internal and external examiners involved)

Course Code	**1.8**
Course Title	**Advanced Databases Lab**
Course Credits	**2**

A course-based project relevant to the concepts covered in the subject shall be implemented and evaluated. The project should demonstrate the practical application of the techniques, and methodologies learned in the course using appropriate programming languages, tools, and technologies.

Evaluation Scheme:

Internal Evaluation (25 marks):

The project should demonstrate the application of theoretical knowledge to solve a real-world problem through system design, implementation and performance evaluation.

The Internal Assessment for the project shall be conducted based on project implementation, research contribution, project presentation, and report writing, as per the following rubric. Evaluation done at Institute level.

The Rubric will have the following Evaluation Parameters:

The Rubric will have the following Evaluation Parameters:	
Evaluation Parameters	**Marks**
Problem Understanding & Formulation	10
Implementation & Functionality	20
Knowledge Representation / Logic Design	10
Analysis & Result Interpretation	5
Report writing or Documentation	5

External Examination – 25 marks Presentation (internal and external examiners involved)

Course Code	**1.9**
Course Title	**Mini Project**
Course Credits	**4**

Mini Project Evaluation Scheme:

Internal Evaluation (50 marks):

In alignment with the courses offered during the semester, namely Artificial Intelligence, Advanced Java, and Advanced Databases, students shall undertake a project that integrates concepts, methodologies, and technologies from these subjects. The project should demonstrate the application of theoretical knowledge to solve a real-world problem through system design, implementation, research, and performance evaluation.

The project shall involve:

- Identification and formulation of a relevant problem statement.
- Review of existing literature and related work.
- Design and development of a working solution using appropriate tools and technologies.
- The project must be developed using Java .It should also utilize Advanced DBMS concepts for efficient database design, implementation, and management, and incorporate appropriate Artificial Intelligence (AI) techniques.
- Testing, validation, and evaluation of the proposed solution.
- Preparation of a comprehensive project report and presentation.

The Internal Assessment for the project shall be conducted based on projectimplementation, research contribution, project presentation, and report writing, as per the following rubric.Students are required to present a paper based on their project work at a national or international conference. Submission of the conference paper and proof of presentation shall form an integral part of the project requirements.

The Rubric will have the following Evaluation Parameters:

Evaluation Parameters	**Marks**
Problem Identification & Objectives	10
Literature Review & Research	10
System Design & Methodology	10
Project Implementation	10
Report writing or Documentation	5
Conference paper presentation	5

External Examination – 50 marks Presentation (internal and external examiners involved)

MCA Semester II

Course Code	**2.1**
Course Title	**Enterprise Architecture**
Course Credits	**2**
Course Outcomes	**After Completion of this Course, students will be able to** 1. Analyze the principles, components, and design considerations of enterprise software architectures. 2. Apply architectural paradigms such as Domain-Driven Architecture (DDA), Service-Oriented Architecture (SOA), Resource-Oriented Architecture (ROA), and Event-Driven Architecture (EDA). 3. Design enterprise solutions using workflow modeling, business process management, and integration patterns. 4. Evaluate modern enterprise technologies including cloud computing, semantic web technologies, and enterprise data management systems.
Module1: Foundations of Enterprise Software Architecture (Credit 0.5)	
Learning Outcomes	**After learning this module, learners will be able to** 1. Understand the evolution and characteristics of enterprise information systems. 2. Analyze enterprise architecture frameworks and middleware technologies. 3. Design enterprise data models using UML, E-R modeling, XML, and JSON. 4. Apply data processing and integration techniques in enterprise environments.
Content Outline	Introduction to Enterprise Information Systems, Enterprise Software Architecture Principles, Middleware Technologies, Web Services and Cloud Computing, Enterprise Data Modeling, UML and Entity-Relationship Models, XML Schemas and JSON Data Representation, Data Processing Strategies, Introduction to XQuery, Data Integration and Information Exchange in Enterprise Systems.
Module 2: Domain-Driven and Service-Oriented Architectures (Credit 0.5)	
Learning Outcomes	**After learning this module, learners will be able to**

	1. Apply domain-driven design principles for enterprise application development. 2. Design business domains and services using Domain-Driven Architecture. 3. Analyze the principles and patterns of Service-Oriented Architecture. 4. Develop loosely coupled and reusable enterprise services.
Content Outline	Domain-Driven Architecture (DDA), Domain-Driven Design (DDD), Bounded Contexts and Domain Models, Object-Relational Mapping (ORM), Introduction to Service-Oriented Architecture (SOA), Standardized Service Contracts, Service Loose Coupling, Service Abstraction, Service Reusability, Service Autonomy, Stateless Services, Service Discoverability, Enterprise Service Design Patterns.
Module 3: Resource-Oriented, Event-Driven and Integration Architectures (Credit 0.5)	
Learning Outcomes	**After learning the module, learners will be able to** 1. Design RESTful enterprise applications using Resource-Oriented Architecture. 2. Implement enterprise integration using messaging and middleware technologies. 3. Analyze event-driven architectures for scalable enterprise solutions. 4. Apply semantic data modeling techniques in distributed systems.
Content Outline	Resource-Oriented Architecture (ROA), REST Architectural Style, Plain Old XML (POX), Hypermedia Networks, RESTful Services and APIs, Message-Oriented Middleware (MOM), Enterprise Integration Patterns, Asynchronous Communication Models, Event-Driven Architecture (EDA), Complex Event Processing (CEP), Event-Based Systems, Semantic Data Modeling, RDF, RDFS, and Enterprise Knowledge Representation.
Module 4: Business Process Management, Cloud Data and Modern Enterprise Platforms (Credit 0.5)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Model and analyze business processes using BPMN and workflow techniques. 2. Design workflow-enabled enterprise systems. 3. Evaluate cloud-based and NoSQL data management solutions. 4. Apply modern enterprise platform technologies and

	dependency management mechanisms.
Content Outline	Business Process Modeling, BPMN Fundamentals, Descriptive and Analytic BPMN, Workflow Management Systems, Petri Nets and Workflow Nets, Business Process Execution Concepts, Cloud Data Stores, Relational and Object Data Models, NoSQL Databases, Semantic Web Technologies using OWL, Contexts and Dependency Injection (CDI), Transactional APIs, Enterprise Platform Design Considerations, Emerging Trends in Enterprise Architecture.

Evaluation Scheme:

External Assessment: End Semester examination of 50 marks for 2 hours duration will be conducted by the university.

References

1. Dominic Duggan, *Enterprise Software Architecture and Design: Entities, Services, and Resources*, Wiley, 2012.
2. Thomas Erl, *Service-Oriented Architecture: Concepts, Technology and Design*, Pearson.
3. Martin Fowler, *Patterns of Enterprise Application Architecture*, Addison-Wesley.
4. Eric Evans, *Domain-Driven Design: Tackling Complexity in the Heart of Software*, Addison-Wesley.
5. Gregor Hohpe and Bobby Woolf, *Enterprise Integration Patterns*, Addison-Wesley.
6. Michael P. Papazoglou, *Web Services and SOA: Principles and Technology*, Pearson.

Course Code	**2.2**
Course Title	**Introduction to Data Science**
Course Credits	**2**
Course Outcomes	**After Completion of this Course, students will be able to** 1. Define the concept, scope, and evolution of Data Science, and differentiate between Data Science, Machine Learning, and Artificial Intelligence. 2. Understand the phases of the Data Science lifecycle, including data collection, preprocessing, modeling, evaluation, and deployment. 3. Apply descriptive statistics and data visualization techniques to summarize and interpret datasets, and identify patterns, trends, and anomalies in data. 4. Apply basic machine learning algorithms such as regression, classification, and clustering, and explain the principles of model training, testing, and evaluation. 5. Illustrate issues related to data privacy, bias, and fairness, and evaluate the importance of ethical data handling and responsible AI practices.
Module1: Introduction to Data Science and Python (Credit 0.5)	
Learning Outcomes	**After learning this module, learners will be able to** 1. Understand the basic concepts, scope, and components of Data Science. 2. Familiarize with data types, sources, and preprocessing techniques. 3. Learn the foundational tools and applications of data science in real-world problems.
Content Outline	Overview of Data Science: What is data science, its applications, Difference between Data Science, Data Analytics, and Business Intelligence, The Data Science lifecycle (CRISP-DM or OSEMN model) ,Roles in data science: Data Scientist, Analyst, Engineer, etc. Applications across industries (healthcare, finance, education, etc.) Data Science Components: - Structured, Semi-structured, and Unstructured data , Introduction to Big Data, Data Collection: Sources and Methods, Data Warehousing and Data Lakes, Data

	preprocessing: Cleaning, Transformation, Normalization
Module 2: Data Manipulation, Analysis and Visualization with Python(Credit 0.5)	
Learning Outcomes	**After learning this module, learners will be able to** 1. Understand the structure and advantages of NumPy 2. Create and manipulate Pandas Series and DataFrames for structured data. 3. Create basic visualizations such as line plots, bar charts, and scatter plots using Matplotlib
Content Outline	NumPy: Working with arrays, numerical operations, linear algebra, and array manipulation. Pandas: Data structures like Series and DataFrames, data cleaning, filtering, and transformation. Data Input/Output: Reading and writing data from various formats (CSV, Excel, etc.). Matplotlib: Creating basic plots, charts, and graphs to visualize data. Seaborn: Advanced visualization techniques, statistical plots, and creating visually appealing charts. Exploratory Data Analysis (EDA) :-Descriptive statistics (mean, median, mode, std. dev.), Data visualization techniques: Histograms, Box plots, Scatter plots ,Correlation and Causation Outlier detection and missing value handling.
Module 3: Statistical Concepts (Credit 0.5)	
Learning Outcomes	**After learning the module, learners will be able to** 1. Define and compute measures of central tendency (mean, median, mode) to summarize data. Calculate measures of dispersion (range, variance, standard deviation) to describe data spread. 2. Interpret data distribution using graphical techniques (histograms, boxplots) and numerical summaries. 3. Understand the concept of population vs. sample and the need for inferential statistics. 4. Implement fundamental probability concepts including independent, dependent events, and conditional probability.
Content Outline	Descriptive Statistics: Measures of central tendency, dispersion, and distribution. Inferential Statistics: Hypothesis testing, confidence intervals, and statistical significance. Probability: Basic probability

	concepts, distributions (normal, binomial, etc.).
Module 4: Applications of Data Science and Emerging Trends (Credit 0.5)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Identify and explain the applications of Data Science across various domains and industries. 2. Describe modern data collection, storage, and analysis techniques used in contemporary data-driven systems. 3. Explore emerging trends and technologies in Data Science, including Big Data and Generative AI. 4. Demonstrate the use of Data Science tools and workflows for solving real-world problems..
Content Outline	**Applications of Data Science** Data Science applications in Healthcare, Finance, E-commerce, Education, Social Media, and Smart Cities Case studies of successful Data Science projects Data-driven decision making in organizations **Recent Trends in Data Science** **Data Science Project Workflow** Problem identification and formulation, Data acquisition and preparation, Data analysis and visualization, Communicating insights through reports and dashboards

Evaluation Scheme:

External Assessment: End Semester examination of 50 marks for 2 hours duration will be conducted by the university.

References:-

1. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, Author: Aurélien Géron, Publisher: O'Reilly Media
2. Data Science from Scratch: First Principles with Python , Author: Joel Grus, Publisher: O'Reilly Media 3. Think Stats: Probability and Statistics for Programmers ,Author: Allen B. Downey,Publisher: O'Reilly / Green Tea Press
4. Practical Statistics for Data Scientists , Authors: Peter Bruce, Andrew Bruce, and Peter Gedeck Publisher: O'Reilly Media
5. Introduction to Machine Learning with Python Authors: Andreas C. Müller & Sarah Guido Publisher: O'Reilly Media

Course Code	**2.3**
Course Title	**Advanced Data Structures**
Course Credits	**2**
Course Outcomes	**After Completion of this Course, students will be able to** 1. Analyze the time and space complexities of basic data structure operations. 2. Analyze the time and space complexities of algorithms. 3. Apply data structures to solve specific computational problems. 4. Evaluate the efficiency and suitability of data structures in different scenarios.
Module1: Linear Data Structures (Credit 0.5)	
Learning Outcomes	**After learning this module, learners will be able to** 1. Define the concept of data structures and their role in organizing and storing data. 2. Identify and classify linear data structures such as arrays, linked lists, stacks, and queues. 3. Analyze the advantages and limitations of each linear data structure
Content Outline	**Introduction:** Data types, ADT, data structure: Definition & classification Analysis of algorithms (recursive and non-recursive) with emphasis on best case, average case and worst case **Linear Data structures with applications:** **List:** Introduction, implementation using array & linked list (singly, doubly, circular, multi-list), Applications: Polynomial representation, Sparse matrix **Stack**: Introduction, implementation using array & linked list, Applications: Function call, Recursion, balancing of parenthesis, Polish Notation: infix to post fix conversion and evaluation of post fix expression **Queue**: Introduction (queue, circular queue, deque, priority queue), implementation using array &linked list, Applications: Job Scheduling

Module 2: Non-Linear Data Structures (Credit 0.5)	
Learning Outcomes	**After learning this module, learners will be able to** 1. Differentiate between linear and non-linear data structures. 2. Understand tree structures, including binary trees and n-ary trees. 3. Implement tree traversal algorithms (e.g., in-order, preorder, post-order).
Content Outline	**Non-Linear data structures:** **Tree:** Introduction and representation, Forest, Tree traversal, Binary Tree (representation using array and links): Binary tree traversal (recursive & non-recursive implementation), Expression tree , Binary search tree, AVL tree, Splay tree, M-way search tree, B tree (insertion), B+ tree (Definition and introduction), Application of B tree and B+ tree in File Structures
Module 3: Graphs (Credit 0.5)	
Learning Outcomes	**After learning the module, learners will be able to** 1. Define graphs and their components (vertices and edges). 2. Implement basic graph traversal algorithms and graph related operations. 3. Analyze the advantages and limitations of each nonlinear data structure
Content Outline	GRAPHS: Representation of graph - Graph Traversals - Depth-first and breadth-first traversal - Applications of graphs - Topological sort – shortest-path algorithms - Dijkstra's algorithm – Bellman-Ford algorithm – Floyd's Algorithm - minimum spanning tree – Prim's and Kruskal's algorithms. Hashing : Introduction, hash functions and hash keys,Collisions, Resolving collisions, Rehashing
Module 4: Sorting, Searching and Algorithm Analysis (Credit 0.5)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Apply data structures to solve real-world problems and scenarios. 2. Understand and Implement search's and Sort's in various scenarios. 3. Analyze the time and space complexity of search and sort algorithms.

	4. Compare and contrast the efficiency of various sorting algorithms. 5. Choose the most appropriate sorting algorithm for specific scenarios.
Content Outline	Sorting :Bubble, Selection, Insertion, Shell, Merge, Quick, Heap, Radix Searching : Linear Search, Binary Search, Transpose sequential search Algorithm Analysis – Asymptotic Notations - Divide and Conquer – Merge Sort – Quick Sort - Binary Search - Greedy Algorithms – Knapsack Problem – Dynamic Programming – Optimal Binary Search Tree – Warshall's Algorithm for Finding Transitive Closure.

Evaluation Scheme:

External Assessment: End Semester examination of 50 marks for 2 hours duration will be conducted by the university.

References:

1. Lipschutz, S. (2017).Data Structures with C. McGraw-Hill.
2. Weiss, M. A. (2003). Data
Structures and Algorithm Analysis in C (2nd ed.). Pearson Education.
3. Pai, G. A. V. (2008). Data Structures and Algorithms: Concepts, Techniques, and Applications (1st ed.).
4. Horowitz, E., Sahni, S., & Anderson-Freed, S. (2007). Fundamentals of Data Structures in C (2nd ed.). University Press. SNDTWU 2025-26 Integrated MCA programme Structure
5. Tremblay, J.-P., & Sorenson, P. G. (2007). An Introduction to Data Structures with Applications (2nd ed.). Tata McGraw-Hill.
6. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2003). Introduction to Algorithms (2nd ed.). PHI.
7. Kanetkar, Y. (BPB publications). Data Structures Through C

Course Code	**2.4**
Course Title	Computer Network Programming using Linux and Java
Course Credits	**2**
Course Outcomes	**After Completion of this Course, students will be able to** 1. Explain fundamental networking concepts, network architectures, protocols, and Linux networking utilities used in network administration and programming. 2. Apply Linux system programming concepts including file management, process management, signals, and shell scripting for system-level application development. 3. Develop interprocess communication (IPC) applications using pipes, message queues, semaphores, shared memory, and socket programming in Linux. 4. Design and implement network-based client-server applications using TCP/IP, UDP sockets, multithreading, and Remote Method Invocation (RMI) in Java. 5. Analyze and develop distributed network applications by integrating Linux system programming and Java networking concepts.
Module1: Fundamentals of Computer Networks and Linux Environment(Credit 0.5)	
Learning Outcomes	**After learning this module, learners will be able to** 1. Explain the basic concepts, architectures, and services of computer networks. 2. Differentiate between OSI and TCP/IP reference models. 3. Utilize Linux networking and file management utilities for system administration tasks. 4. Develop basic shell scripts using Bash programming constructs.
Content Outline	***Introduction to Computer Networks*** Computer Networks and Applications, Network Hardware and Software Components, Interconnection Networking Devices, Network Operating Systems,Network Topologies and Protocol Hierarchies, Connection-Oriented and Connectionless Services,OSI Reference

	Model,TCP/IP Reference Model, Communication Between Layers ***Linux Utilities and Shell Programming*** File Handling Utilities,Security through File Permissions, Process Utilities,Disk Utilities, Networking Utilities, Filters and Text Processing Utilities, Backup Utilities, Bourne Again Shell (Bash), Pipes and Redirection, Shell Variables and Environment Variables, Command Substitution,Quoting and Meta Characters, Test Command, Control Structures, Arithmetic Operations in Shell, Shell Script Development and Examples
Module 2: Linux System Programming – Files, Processes and Signals(Credit 0.5)	
Learning Outcomes	**After learning this module, learners will be able to** 1. Describe Linux file system architecture and file management mechanisms. 2. Implement file handling operations using system calls. 3. Create and manage processes using Linux process control mechanisms. 4. Develop applications utilizing signal handling techniques.
Content Outline	***File Management*** File Concept and File Types, File System Structure, I-Nodes and File Attributes, File I/O using System Calls, Kernel Support for Files, File Status Information (stat family), File and Record Locking (lockf, fcntl), File Permissions (chmod, fchmod), File Ownership (chown, lchown, fchown), Hard and Soft Links, link(), unlink(), symlink() ***Directory Management*** Directory Structure and Contents, Directory Handling APIs, Scanning Directories ***Process Management*** Process Concept and Process Attributes, Kernel Support for Processes, Process Creation and Control, Replacing Process Images, Waiting for Process Completion, Process Termination, Zombie and Orphan Processes ***Signal Handling*** Introduction to Signals, Signal Generation and Handling, Reliable and Unreliable Signals, Signal APIs:kill(), raise(), alarm(), pause(),

	abort(), sleep()
Module 3: Interprocess Communication and Socket Programming in Linux(Credit 0.5)	
Learning Outcomes	**After learning the module, learners will be able to** 1. Implement interprocess communication using Linux IPC mechanisms. 2. Develop applications using message queues, semaphores, and shared memory. 3. Design client-server applications using Unix and Internet domain sockets. 4. Configure and manage socket options for network communication.
Content Outline	***Interprocess Communication (IPC)*** Introduction to IPC Mechanisms, Pipes and Unnamed Pipes, IPC Between Related Processes, Named Pipes (FIFOs), IPC Between Unrelated Processes, popen() and pclose() ***Message Queues*** Kernel Support for Messages, UNIX System V Message Queue APIs, Client-Server Applications using Message Queues ***Semaphores*** Kernel Support for Semaphores,UNIX System V Semaphore APIs ***Shared Memory*** Kernel Support for Shared Memory, UNIX System V Shared Memory APIs, Shared Memory Client-Server Applications ***Socket Programming*** Introduction to Unix Sockets,Client-Server Model,Unix Domain and Internet Domain Address Formats, Connection-Oriented Communication, Connectionless Communication,Single Client-Server Applications, Multiple Concurrent Client Handling, Socket Options: setsockopt(), getsockopt(), fcntl()
Module 4: Network Programming using Java (Credit 0.5)	
Learning Outcomes	**After learning the module, learners will be able to**

	1. Develop network applications using Java socket programming. 2. Implement TCP and UDP-based communication between distributed systems. 3. Design multithreaded client-server applications capable of handling multiple clients. 4. Build distributed applications using Java Remote Method Invocation (RMI).
Content Outline	***Java Socket Programming*** Introduction to Java Networking APIs,InetAddress Class, URL and URLConnection, TCP Socket Programming, Client and Server Socket Classes, Data Streams in Network Communication ***UDP Programming*** Datagram Packets, Datagram Sockets, Connectionless Communication using UDP ***Multithreaded Network Programming*** Concurrent Client Handling, Thread-Based Server Design, Multi-client Chat and Service Applications ***Remote Method Invocation (RMI)*** Introduction to Distributed Computing, RMI Architecture, Remote Interfaces and Remote Objects, RMI Registry, Developing RMI Applications, Client-Server Implementation using RMI, Exception Handling and Security Considerations in RMI,

Evaluation Scheme:

External Assessment: End Semester examination of 50 marks for 2 hours duration will be conducted by the university.

References

1. TCP/IP Illustrated, Volume 1 by W. Richard Stevens, Addison-Wesley.
2. UNIX System Programming Using C++ by T. Chan, Prentice Hall.
3. The Linux Programming Interface by Michael Kerrisk, No Starch Press, 2010.
4. Java Network Programming by Elliotte Rusty Harold, O'Reilly Media.
5. Distributed Systems: Concepts and Design by George Coulouris, Jean Dollimore, Tim Kindberg and Gordon Blair, Pearson Education.
6. Linux System Programming by Robert Love, O'Reilly Media.

Course Code	**2.5(Elective 1 Sem II)**
Course Title	**Generative AI**
Course Credits	**2**
Course Objective	**After the completion of this course, students will be able to:** 1. To provide a strong foundation in Artificial Intelligence, Deep Learning, and Generative AI techniques for developing intelligent systems. 2. To introduce modern generative models, Large Language Models (LLMs), and Transformer-based architectures for text, image, and music generation. 3. To develop practical skills in implementing and fine-tuning generative AI models using Python and TensorFlow. 4. To enable students to design, evaluate, and deploy AI-driven applications across diverse domains. 5. To promote awareness of ethical, societal, and emerging trends in Generative AI technologies.
Course Outcomes	**After the completion of this course, students will be able to:** 1. Explain the fundamental concepts of Artificial Intelligence, Deep Learning, Neural Networks, and Generative AI architectures. 2. Analyze and apply Large Language Models, GPT, BERT, and Transformer-based techniques for Natural Language Processing tasks. 3. Develop and implement Generative AI solutions for image generation using GANs, VAEs, and TensorFlow-based frameworks. 4. Design and deploy text generation systems using LSTM and Transformer architectures for real-world applications. 5. Create and evaluate AI-based music generation systems using advanced generative models such as LSTM, Transformers, and MuseGAN. 6. Assess the performance, limitations, ethical implications, and future directions of Generative AI technologies and applications.
Module1: Foundations of Deep Learning and Generative AI: (Credit 0.5)	

Learning Outcomes	**After learning this module, learners will be able to** 1. Explain the evolution of AI, Machine Learning, Deep Learning, and Natural Language Processing. 2. Analyze the architecture and training process of Artificial Neural Networks, including backpropagation and hyperparameter tuning. 3. Apply data preprocessing, probability, and statistical techniques for AI model development. 4. Compare and evaluate generative AI architectures such as GANs, VAEs, Transformers, LSTMs, and attention mechanisms.
Content Outline	History and evolution of AI/ML, Deep learning revolution, Transfer learning, History of Neural Natural Language Processing, Structure of Artificial Neural Networks, Steps in Training an Artificial Neural Network, Parameters and Hyperparameters, Backpropagation. **Introduction to Generative AI Models**: Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs), Transformers, Attention Mechanism in detail Long Short-Term Memory Networks (LSTMs). **Data Preprocessing:** Probability and Statistics, Data Preprocessing Techniques, Model Training Techniques. Introduction to advanced architectures,
Module2:Large Language Models and Advanced Transformer Architectures (Credit 0.5)	
Learning Outcomes	**After learning this module, learners will be able to** 1. Describe the fundamentals of Large Language Models and Transformer-based architectures. 2. Analyze GPT architectures and implement text generation and conversational AI applications. 3. Fine-tune and evaluate GPT models for domain-specific tasks and chatbot development. 4. Apply BERT and advanced Transformer techniques to solve various Natural Language Processing tasks.
Content Outline	**Large Language Models :** Overview of Generative AI and Large Language Models. Basics of attention mechanisms andTransformer architecture. Pre-training techniques and transfer learning strategies. **GPT Models and Applications:**

	Study of GPT architecture and variants. Applications of GPT models in text generation anddialogue systems. Case study based implementation of GPT-based tasks. GPT-based chatbotenhances E-Shop's customer support service **BERT and Advanced Techniques:** Understanding BERT architecture and pre-training objectives. Fine-tuning BERT for downstream NLP tasks. Exploration of advanced Transformer architectures and techniques
Module 3:Generative AI for Image using TensorFlow (Credit 0.5)	
Learning Outcomes	**After learning the module, learners will be able to** 1. Utilize Python and TensorFlow 2 for developing Generative AI applications. 2. Implement and train GAN and VAE models for image generation tasks. 3. Evaluate and optimize generative models for image synthesis applications.
Content Outline	**Python and TensorFlow 2 in Generative AI**: Overview of Python and TensorFlow 2, Visualizing data distributions and patterns in Generative AI datasets. Introduction to TensorFlow's computation graph and eager execution. **Image Generation with Generative AI:** Introduction to Image Generation, Implementing GANs for Image Generation Training and Fine-Tuning GANs , Generating Images with VAEs,
Module 4: Audio Generation and Text Generation withGenerative AI(Credit 0.5)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Explain the fundamental concepts, architectures, and applications of Generative AI for audio and text generation. 2. Analyze and compare different generative models, including LSTM-based and Transformer-based approaches for content generation. 3. Apply Generative AI tools and techniques to create speech, voice, music, and textual content. 4. Evaluate the quality and effectiveness of AI-generated audio and text using appropriate assessment methods.

	5. Assess ethical, legal, and societal implications associated with the use of Generative AI technologies for content creation.
Content Outline	**Audio Generation with Generative AI:** Introduction to Generative AI for audio, covering speech synthesis, voice generation, music creation, and virtual assistants. Overview of digital audio concepts, feature extraction, and audio generation models. Hands-on experience with Text-to-Speech (TTS), voice cloning, and AI-based audio tools, along with basic evaluation methods and ethical considerations. **Text Generation with Generative AI:** Introduction to Text Generation, LSTM-based Text Generation, Transformer-based Text Generation,

Evaluation Scheme:

External Assessment: End Semester examination of 50 marks for 2 hours duration will be conducted by the university.

References Book:

1. Generative AI on AWS by Chris Fregly and Antje Barth, O'Reilly Media, 2024.
2. Generative Deep Learning by David Foster, 2nd Edition, O'Reilly Media, 2023.
3. Building LLMs for Production by Louis-François Bouchard, Packt Publishing, 2024.
4. Goodfellow, Yoshua Bengio and Aaron Courville, MIT Press, 2016.
5. Natural Language Processing with Transformers by Lewis Tunstall, Leandro von Werra and Thomas Wolf, O'Reilly Media, 2022.
6. Hands-On Large Language Models by Jay Alammar and Maarten Grootendorst, O'Reilly Media, 2024.
7. Machine Learning with PyTorch and Scikit-Learn by Sebastian Raschka, Yuxi Liu and Vahid Mirjalili, Packt Publishing, 2022.
8. Transformers for Natural Language Processing by Denis Rothman, Packt Publishing, 2024.
9. Artificial Intelligence: A Modern Approach by Stuart Russell and Peter Norvig, 4th Edition, Pearson, 2021.

CourseCode	**2.5(Elective 2 Sem II)**
CourseTitle	**Social Network Analysis**
CourseCredits	**2**
CourseOutcomes	**AfterCompletionofthisCourse,studentswillbeableto** 1. Distinguish between various network types and relationships, including binary, valued, symmetric, and asymmetric relationships. 2. Utilize graph theory for social network analysis, employing methods like adjacency matrices, edge-lists, and graph traversals. 3. Investigate the role of ontology in the Semantic Web and its relevance in network data representation. 4. Analyze network structures, centrality, and centralization in Social Network Analysis (SNA), encompassing measures like density, reachability, and centrality.
Module1:Introduction to social network analysis (SNA) (Credit 0.5)	
Learning Outcomes	**After learning the module, learners will be able to** 1. Analyze network relationships, discerning between various types such as binary, valued, symmetric, and asymmetric. 2. Utilize graph theory techniques like adjacency matrices and edge-lists for social network analysis. 3. Apply clustering methods in social networks, connecting theoretical concepts with real-world electronic sources. 4. Develop skills in understanding and utilizing ego-centric and socio-centric density measures within networks.
Content Outline	**Introduction to networks and relations**: analyzing relationships to understand people and groups, binary and valued relationships, symmetric and asymmetric relationships, multimode relationships. **Using graph theory for social networks analysis**: adjacency matrices, edgelists, adjacency lists, graph traversals and distances, social networks vs. link analysis, ego-centric and socio-centric density , clustering. **Social Network analysis:** Development of Social Network Analysis, Electronic sources for network analysis, Blogs and online communities.
Module2:Networks, Centrality, centralization and Ontology (Credit 0.5)	
Learning Outcomes	**After learning the module, learners will be able to** 1. Understand the role of ontology in the Semantic Web and its

	application in knowledge representation. 2. Gain expertise in Social Network Analysis (SNA) techniques for analyzing network characteristics. 3. Analyze network features like density, reachability, and centrality measures such as degree and closeness. 4. Emphasize interpreting and visualizing network structures using centrality algorithms like PageRank.
Content Outline	**Ontology and their role in the Semantic Web:** Semantic Web , Ontology, Ontology based knowledge Representation , Resource Description Framework – Web Ontology ,State-of-theart in network data representation ,Ontological representation of social individuals ,Ontological representation of social relationships. **Networks, Centrality and centralization in SNA Understanding networks:** density, reachability, connectivity, reciprocity, group-external and groupinternal ties in networks, ego networks, extracting and visualizing ego networks, structural holes, Centralitydegree of centrality, closeness and betweenness centrality, local and global centrality, centralization and graph centers, notion of importance within network, Google pagerank algorithm.
Module3:Extraction and mining communities in web social network (Credit 0.5)	
Learning Outcomes	**After learning the module, learners will be able to** 1. Detect and evaluate communities within web social networks with proficiency. 2. Utilize community detection methods and mining algorithms, including tools like Girvan Newman. 3. Grasp measures of similarity and structural equivalence in SNA, exploring various approaches. 4. Understand clustering techniques and diverse similarity metrics for network analysis.
Content Outline	**Communities in Web Social Network:** Detecting communities in social networks, Definition of community, Evaluating communities, Methods for community detection and mining,Applications of community mining algorithms, Tools for detecting communities social network infrastructures and communities , Girvan Newman algorithm ,Decentralized online social networks , MultiRelational characterization of dynamic social network communities. **Measures of similarity and structural equivalence in SNA:** Approaches to network positions and social rolesdefining equivalence or similarity, structural equivalence, automorphic equivalence, finding

	equivalence sets, brute force and Tabu search, regular equivalence, equivalence of distances: Maxsim, regular equivalence **Understanding clustering:** agglomerative and divisive clusters, Euclidean, Manhattan, and squared distances, binary relations, matches , exact, Jaccard, Hamming
Module4:Two-mode networks for SNA: (Credit 0.5)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Master two-mode networks, including bipartite data structures and quantitative analyses like SVD. 2. Excel in qualitative analysis techniques, such as core-periphery and factions analysis. 3. Explore intricacies of affiliation and attribute networks within two-mode structures. 4. Conduct comprehensive analysis and visualization of two-mode data sets.
Content Outline	**Understanding mode networks:** Bipartite data structures, visualizing twomode data, quantitative analysis using two-mode Singular value decomposition (SVD) analysis, two-mode factor analysis, two-mode correspondence analysis, qualitative analysis using two mode core-periphery analysis, two-mode factions analysis, affiliation and attribute networks

Evaluation Scheme:

External Assessment: End Semester examination of 50 marks for 2 hours duration will be conducted by the university.

References:

1. Mika, P. (2007). Social Networks and the Semantic Web (1st ed.). Springer.

2. Furht, B. (2010). Handbook of Social Network Technologies and Applications (1st ed.).

Springer.

3. Hanneman, R. A., & Riddle, M. (2005). Introduction to Social Network Methods. University
of California. [Published in digital form and available athttp://faculty.ucr.edu/~hanneman/nettext/index.html].

4. Tsvetovat, M., &Kouznetsov, A. (2011). Social Network Analysis for Startups: Finding
Connections on the Social Web. O'Reilly Media.

5. Scott, J. (2012). Social Network Analysis (3rd ed.).

CourseCode	**2.5(Elective 3 Sem II)**
CourseTitle	**Internet of Things**
CourseCredits	**2**
CourseOutcomes	**AfterCompletionofthisCourse,studentswillbeableto** 1. Understand IoT Fundamentals: Explain the basic concepts, architecture, and components of IoT systems including sensors, actuators, communication protocols, and cloud integration. 2. Understand the Impact of IoT on Society and Industry: Assess the potential impacts, challenges, and ethical considerations of IoT in different sectors such as healthcare, agriculture, smart cities, and manufacturing. 3. Design and Develop IoT Solutions: Develop simple IoT applications using microcontrollers (e.g., Arduino, Raspberry Pi) and relevant sensors/actuators to solve real-world problems. 4. Implement Data Collection and Processing: Collect, process, and visualize IoT sensor data using edge computing, cloud platforms, and databases.
Module1:Introduction to IoT& M2M (Credit 0.5)	
Learning Outcomes	**After learning the module, learners will be able to** 1. Understand the fundamental of IoT 2. Understand the components used in IoT design 3. Understand the fundamental of M2M
Content Outline	**Introduction to IoT:** Definition of IoT, Characteristics of IoT, Physical design of IoT, Logical design of IoT, Functional blocks of IoT, Communication models, Communication APIs, Communication Protocols, Challenges in IoT Design challenges **Introduction to M2M :** IoT& M2M Machine to Machine, Difference between IoT and M2M, Software define Network
Module2:IoT Network Architecture, Protocols & Data Management (Credit 0.5)	

Learning Outcomes	**After learning the module, learners will be able to** 1. Understand the concept of IoTWF 2. Understand the various communication protocols and its use 3. Gain knowledge about network & communication concepts
Content Outline	**IoT Network Architecture and Design:** The IoT World Forum (IoTWF) Standardized Architecture, A Simplified IoT Architecture, IoT protocol stack, The Core IoT Functional Stack **IoT Data Management and Compute Stack:** Fog Computing, Edge Computing, MAC protocol survey, Survey routing protocols, Sensor deployment & Node discovery, Data aggregation & dissemination.
Module3:IoT Platforms, Devices & Development using Python (Credit 0.5)	
Learning Outcomes	**After learning the module, learners will be able to** 1. Understand the concepts related to IoT Platforms 2. Gain knowledge about various IoT platforms 3. Understand the basics of python
Content Outline	**IoT Platforms:** IPv6 Tunneling, IPsec in IPv6, Header Compression Schemes, Raspberry Pi, Raspberry Pi Interfaces. **Other IoT Devices:** pcDuino, Beagle Bone Black, CubieBoard, ARDUINO **Developing IoTs:** Introduction to Python, Introduction to different IoT tools, Developing applications through IoT tools
Module4:IoT Applications, Case Studies & Societal Impact (Credit 0.5)	
Learning Outcomes	**After learning the module, learners will be able to** 1. Identify the applications of IoT in various fields 2. Elaborate case studies
Content Outline	**IoT case studies:** Domain specific applications of IoT Home automation, Industry applications, Surveillance applications, Other IoT applications

Evaluation Scheme:

External Assessment: End Semester examination of 50 marks for 2 hours duration will be conducted by the university.

Reference Books

1. Vijay Madisetti, ArshdeepBahga, “Internet of Things: A Hands-On Approach”

2. Smart Internet of things projects AgusKurniawanPackt - Sep 2016 978-1- 78646- 651-8.

3. The Internet of Things Key Olivier Willy Publication 2nd Edition 978

4. WaltenegusDargie,ChristianPoellabauer, "Fundamentals of Wireless Sensor Networks:

Theory and Practice"

5. IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of

Things David Hanes, Gonzalo Salgueiro, Patrick Grossetete Cisco Press – Paperback –

16 Aug 2017 978-1- 58714-456- 1 599.

CourseCode	**2.5(Elective 4 Sem II)**
CourseTitle	**BLOCK CHAIN**
CourseCredits	**2**
CourseOutcomes	**AfterCompletionofthisCourse,studentswillbeableto** 1. Apply understanding of Blockchain Concepts, including its decentralized nature, distributed ledger, and cryptographic security features. 2. Analyze the historical context and evolution of blockchain technology, including the development of the first blockchain. 3. Evaluate the Nakamoto consensus and different consensus algorithms. 4. Design exploration of concepts such as interoperability, portability in Hyperledger Fabric, and the concept of sharding in blockchain.
Module1:Fundamentals of Blockchain (Credit 0.5)	
Learning Outcomes	**After learning the module, learners will be able to** 1. Apply principles of distributed databases, including architecture, advantages, and challenges, to design and manage data across multiple network nodes. 2. Analyze complexities of consensus in distributed systems, considering conflicting or malicious information, and understand the significance of Byzantine fault tolerance. 3. Evaluate the concept of ASIC resistance in crypto-currencies, exploring motivations and implications of designing systems to resist mining centralization through specialized hardware. 4. Design a comprehensive understanding of cryptography principles, including confidentiality, integrity, and authenticity, learning applications and functions of hash functions, digital signatures (specifically ECDSA), memory-hard algorithms, and zero knowledge proofs.
Content Outline	Distributed Database, Two General Problem, Byzantine General problem and Fault Tolerance, Hadoop Distributed File System, Distributed Hash Table, ASIC resistance, Turing Complete. Cryptography: Hash function, Digital Signature - ECDSA, Memory Hard Algorithm, Zero Knowledge Proof.
Module2:Blockchain , Distributed Consensus: (Credit 0.5)	

Learning Outcomes	**After learning the module, learners will be able to** 1. Apply fundamental concepts of blockchain, including its decentralized nature, distributed ledger, and cryptographic security features. 2. Analyze the structure and operation of a blockchain network, focusing on nodes, peers, and the peer-to-peer communication model. 3. Evaluate the differences between private and public blockchains, considering their use cases, access control, and levels of decentralization. 4. Design blockchain solutions by exploring the Nakamoto consensus and various consensus algorithms, such as Proof of Work, Proof of Stake, and Proof of Burn, to understand their strengths and weaknesses.
Module3:Cryptocurrency , Cryptocurrency Regulation: (Credit 0.5)	
Learning Outcomes	**After learning the module, learners will be able to** 1. Apply knowledge of the historical context and evolution of blockchain technology, including the development of the first blockchain with Bitcoin. 2. Analyze the construction of the Ethereumblockchain, focusing on its architecture and functionalities. 3. Evaluate the concept and implementation of smart contracts and their role in Decentralized Autonomous Organizations (DAOs). 4. Design blockchain applications utilizing Ethereum's smart contracts and DAO principles to create decentralized solutions.
Content Outline	History, Distributed Ledger, Bitcoin protocols - Mining strategy and rewards, Ethereum - Construction, DAO, Smart Contract, GHOST, Vulnerability, Attacks, Sidechain, Namecoin. Stakeholders, Roots of Bit coin, Legal Aspects-Crypto currency Exchange, Black Market and Global Economy. Applications: Internet of Things, Medical Record Management System, Domain Name Service and future of Blockchain.
Module4:Hyperledger , Scalability and other challenges : (Credit 0.5)	
Learning Outcomes	**After learning the module, learners will be able to** **1.** Apply the features of Hyperledger Fabric, such asmodular architecture and identity management, to blockchainsolutions. **2.** Analyze scalability challenges in blockchain networks and

	how Hyperledger Fabric addresses them. **3.** Evaluate Hyperledger Fabric's privacy, confidentiality, and deterministic transaction mechanisms. **4.** Design interoperable and portable applications using Hyperledger Fabric to ensure seamless integration with other systems.
Content Outline	Hyperledger as a protocol :The reference architecture Requirements and design goals of Hyperledger Fabric: The modular approach Privacy and confidentiality, Scalability, Deterministic transactions Identity, Auditability Interoperability Portability Rich data queries Fabric Hyperledger Fabric Membership services Blockchain services Consensus services Distributed ledger ,The peer to peer protocol Ledger storage Chaincode services ,Components of the fabric Scalability and Other Challenges: Scalability Network plane ,Consensus plane, Storage plane View plane ,Block size increase ,Block interval reduction Invertible Bloom, Lookup Tables Sharding State channels Private blockchain, Proof of Stake Sidechains Subchains Tree chains (trees) Block propagation Bitcoin-NG, Plasma ,Privacy Indistinguishability Obfuscation Homomorphic encryption ,Zero-Knowledge Proofs State channels Secure multiparty computation Usage of hardware to provide confidentiality CoinJoin Confidential transactions, MimbleWimble Security Smart contract security Formal verification and analysis Oyente tool

Evaluation Scheme:

External Assessment: End Semester examination of 50 marks for 2 hours duration will be conducted by the university.

Reference Books

1. Antonopoulos, A. M. (2014). Mastering Bitcoin: Unlocking digital cryptocurrencies. O'Reilly Media.
2. Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. Link to the Bitcoin Whitepaper
3. Wood, G. (2014). Ethereum: A secure decentralized transaction ledger (Yellow paper). Link to the Ethereum Yellow Paper
4. Atzei, N., Bartoletti, M., &Cimoli, T. (2017). A survey of attacks on Ethereum smart contracts.
5. Bashir, I. (2018). Mastering blockchain. Wiley

Course Code	**2.5(Elective 5 Sem II)**
Course Title	**BUSINESS INTELLIGENCE AND DATA ANALYTICS**
Course Credits	**2**
Course Outcomes	**After Completion of this Course, students will be able to** 1. Understand the role of business analytics in organizational decision-making 2. Analyze and visualize business data using appropriate tools and techniques 3. Apply statistical and predictive models to solve business problems 4. Interpret analytical results and provide actionable business insights
Module 1: Introduction to Business Analytics (Credit 0.5)	
Learning Outcomes	**After learning this module, learners will be able to** 1. Define business analytics and its importance 2. Identify different types of analytics (descriptive, predictive, prescriptive) 3. Understand data sources and data-driven decision-making. 4. Explore real-world applications of analytics
Content Outline	Introduction to business analytics **Types of analytics:** descriptive, diagnostic, predictive, prescriptive Role of analytics in business decision-making Data sources and data collection methods Case studies of analytics applications in business
Module2:Data Preprocessing and Visualization (Credit 0.5)	
Learning Outcomes	**After learning this module, learners will be able to** 1. Clean and preprocess data for analysis 2. Apply data transformation techniques
Content Outline	Data cleaning and preprocessing Handling missing values and outliers Data transformation and normalization

	Data visualization techniques (charts, graphs, dashboards) Tools overview (Excel, Python, Tableau/Power BI)
Module 3:Statistical and Predictive Analytics(Credit 0.5)	
Learning Outcomes	**After learning the module, learners will be able to** 1. Apply statistical techniques for data analysis 2. Understand probability and distributions 3. Build predictive models 4. Evaluate model performance
Content Outline	Descriptive statistics (mean, median, variance) Probability concepts and distributions Correlation and regression analysis Introduction to predictive modeling Model evaluation techniques
Module 4: Analytics-Driven Decision Making (Credit 0.5)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Apply analytics in different business domains 2. Understand prescriptive analytics and optimization 3. Interpret analytical results for decision-making 4. Develop business analytics projects
Content Outline	Prescriptive analytics and optimization Analytics applications in marketing, finance, HR, operations Decision-making using analytics Ethical issues in business analytics Mini project development

Evaluation Scheme:

External Assessment: End Semester examination of 50 marks for 2 hours duration will be conducted by the university.

References:

1. Evans, J. R. – *Business Analytics*
2. Provost, F., & Fawcett, T. – *Data Science for Business*
3. Albright, S. C., & Winston, W. – *Business Analytics: Data Analysis & Decision Making*
4. Sharda, R., Delen, D., & Turban, E. – *Business Intelligence and Analytics*

Course Code	**2.6**
Course Title	**Introduction to Data Science Lab**
Course Credits	**2**

A course-based project relevant to the concepts covered in the subject shall be implemented and evaluated. The project should demonstrate the practical application of the techniques, and methodologies learned in the course using appropriate programming languages, tools, and technologies.

Evaluation Scheme:

Internal Evaluation (25 marks):

The project should demonstrate the application of theoretical knowledge to solve a real-world problem through system design, implementation and performance evaluation.

The Internal Assessment for the project shall be conducted based on project implementation, research contribution, project presentation, and report writing, as per the following rubric. Evaluation done at Institute level.

The Rubric will have the following Evaluation Parameters:

The Rubric will have the following Evaluation Parameters:	
Evaluation Parameters	**Marks**
Problem Understanding & Formulation	10
Implementation & Functionality	20
Knowledge Representation / Logic Design	10
Analysis & Result Interpretation	5
Report writing or Documentation	5

External Examination – 25 marks Presentation (internal and external examiners involved)

Course Code	**2.7**
Course Title	**Enterprise Architecture Lab**
Course Credits	**2**

A course-based project relevant to the concepts covered in the subject shall be implemented and evaluated. The project should demonstrate the practical application of the techniques, and methodologies learned in the course using appropriate programming languages, tools, and technologies.

Evaluation Scheme:

Internal Evaluation (25 marks):

The project should demonstrate the application of theoretical knowledge to solve a real-world problem through system design, implementation and performance evaluation.

The Internal Assessment for the project shall be conducted based on project implementation, research contribution, project presentation, and report writing, as per the following rubric. Evaluation done at Institute level.

The Rubric will have the following Evaluation Parameters:

The Rubric will have the following Evaluation Parameters:	
Evaluation Parameters	**Marks**
Problem Understanding & Formulation	10
Implementation & Functionality	20
Knowledge Representation / Logic Design	10
Analysis & Result Interpretation	5
Report writing or Documentation	5

External Examination – 25 marks Presentation (internal and external examiners involved)

Course Code	**2.8**
Course Title	**Advanced Data Structure Lab**
Course Credits	**2**

A course-based project relevant to the concepts covered in the subject shall be implemented and evaluated. The project should demonstrate the practical application of the techniques, and methodologies learned in the course using appropriate programming languages, tools, and technologies.

Evaluation Scheme:

Internal Evaluation (25 marks):

The project should demonstrate the application of theoretical knowledge to solve a real-world problem through system design, implementation and performance evaluation.

The Internal Assessment for the project shall be conducted based on project implementation, research contribution, project presentation, and report writing, as per the following rubric. Evaluation done at Institute level.

The Rubric will have the following Evaluation Parameters:

The Rubric will have the following Evaluation Parameters:	
Evaluation Parameters	**Marks**
Problem Understanding & Formulation	10
Implementation & Functionality	20
Knowledge Representation / Logic Design	10
Analysis & Result Interpretation	5
Report writing or Documentation	5

External Examination – 25 marks Presentation (internal and external examiners involved)

Course Code	**2.9**
Course Title	**Computer Network Programming using Linux and Java Lab**
Course Credits	**2**

A course-based project relevant to the concepts covered in the subject shall be implemented and evaluated. The project should demonstrate the practical application of the techniques, and methodologies learned in the course using appropriate programming languages, tools, and technologies.

Evaluation Scheme:

Internal Evaluation (25 marks):

The project should demonstrate the application of theoretical knowledge to solve a real-world problem through system design, implementation and performance evaluation.

The Internal Assessment for the project shall be conducted based on project implementation, research contribution, project presentation, and report writing, as per the following rubric. Evaluation done at Institute level.

The Rubric will have the following Evaluation Parameters:

The Rubric will have the following Evaluation Parameters:	
Evaluation Parameters	**Marks**
Problem Understanding & Formulation	10
Implementation & Functionality	20
Knowledge Representation / Logic Design	10
Analysis & Result Interpretation	5
Report writing or Documentation	5

External Examination – 25 marks Presentation (internal and external examiners involved)

Course Code	**2.10**
Course Title	**ElectiveI Lab**
Course Credits	**2**

A course-based project relevant to the concepts covered in the subject shall be implemented and evaluated. The project should demonstrate the practical application of the techniques, and methodologies learned in the course using appropriate programming languages, tools, and technologies.

Evaluation Scheme:

Internal Evaluation (25 marks):

The project should demonstrate the application of theoretical knowledge to solve a real-world problem through system design, implementation and performance evaluation.

The Internal Assessment for the project shall be conducted based on project implementation, research contribution, project presentation, and report writing, as per the following rubric. Evaluation done at Institute level.

The Rubric will have the following Evaluation Parameters:

The Rubric will have the following Evaluation Parameters:	
Evaluation Parameters	**Marks**
Problem Understanding & Formulation	10
Implementation & Functionality	20
Knowledge Representation / Logic Design	10
Analysis & Result Interpretation	5
Report writing or Documentation	5

External Examination – 25 marks Presentation (internal and external examiners involved)

Course Code	**2.11**
Course Title	**Mini Project**
Course Credits	**4**

Mini Project Evaluation Scheme:

Internal Evaluation (50 marks):

In alignment with the courses offered during the semester, namely Enterprise Architecture, Introduction to Data Science, Advanced Data Structures, Computer Network Programming using Linux and Java, and the elective, learners shall develop a project that integrates concepts, methodologies, tools, and technologies from one or more of these courses. The project should demonstrate the application of theoretical knowledge to address a real-world problem through system analysis, design, implementation, experimentation, and evaluation.

The project shall involve:

- Identification and formulation of a relevant problem statement in an application domain.
- Review of literature, existing solutions, and recent technological advancements related to the selected problem.
- System architecture design incorporating appropriate concepts from the relevant course(s).
- Development and implementation of a working prototype, model, framework, or application using suitable programming languages , platforms, and tools.
- Application of domain-specific techniques such as data analytics, machine learning, , network programming, enterprise architecture frameworks, advanced data structures, or business intelligence methodologies, wherever applicable.
- Testing, validation, performance analysis, and evaluation of the proposed solution using appropriate metrics.
- Preparation of a comprehensive project report documenting objectives, methodology, implementation details, results, and conclusions.
- Presentation and demonstration of the project before the evaluation committee.

The Internal Assessment shall be conducted based on project implementation, innovation, research contribution, technical complexity, project presentation, and report writing, as per the prescribed rubric.

Learners are encouraged to publish or present a research paper based on their project work in a National or International Conference, Seminar, Symposium, or Journal. Submission of the manuscript, acceptance letter, publication, or proof of presentation may be considered as an additional component demonstrating research contribution and academic engagement.

The Rubric will have the following Evaluation Parameters:

Evaluation Parameters	**Marks**
Problem Identification & Objectives	10
Literature Review & Research	10
System Design & Methodology	10
Project Implementation	10
Report writing or Documentation	5
Conference paper presentation	5

External Examination – 50 marks Presentation (internal and external examiners involved)

Semester III

CourseCode	**3.1**
CourseTitle	**CyberSecurity**
Course Credits	**4 Credits**
Course Objective	**1.** ProvideanoverviewofthefieldofCyberSecurity, includingitschallenges,constraints,andtherole ofInternet governance.
	2. Understandthe imperativefor acomprehensiveCyber Securitypolicy,theestablishmentofanodalauthority,and the importance of an international convention on cyberspace.
	3.Identifyvulnerabilitiesinsoftware, system administration,networkarchitectures,dataaccess, authentication,broadbandcommunications,andpoor awareness.
	4. Identify intrusion types, such as physical theft, privilege abuse,unauthorized access, malware infection, andimplementtechniquesincludinganti-malwaresoftware, network-basedintrusiondetection/preventionsystemswith its types.
Module1IntroductiontoCyberSecurity(Credit1)	
Learning Outcomes	**Afterlearningthemodule,learnerswill beableto**
	1. Demonstratetheabilitytoconductsecurityaudits, identifying potential, weaknesses in systems and networks. 2. Explaintheroleofcryptography inCyberSecurityand apply cryptographic techniques to secure data communication. 3. Understandtheconceptofethicalhackinganditsrolein proactivelyidentifyingandaddressing vulnerabilities.
Course Outline	**IntroductiontoCyberSecurity** OverviewofCyber Security, InternetGovernance– ChallengesandConstraints,Cyber Threats:- Cyber Warfare-Cyber Crime-Cyber, terrorism-Cyber Espionage, Needfor a Comprehensive CyberSecurityPolicy,NeedforaNodalAuthority,Need for an International conventiononCyberspace **CyberSecurityVulnerabilities andSafeguards** Cyber Security Vulnerabilities Overview, vulnerabilities in software, System administration,Complex Network Architectures,OpenAccesstoOrganizationalData, Weak Authentication, UnprotectedBroadband. Communications, Poor Cyber Security Awareness. Cyber Security Safeguards Overview,

	Accesscontrol, Audit, Authentication, Biometrics, Cryptography, Deception, Denial of Service Filters, Ethical Hacking, Firewalls, Intrusion Detection Systems, Response, Scanning, Security policy, Threat Management.
Module2SecuringWebApplication(Credit1)	
Learning Outcomes	**Afterlearningthemodule,learnerswill beableto**
	1. Defineandexplaintherolesofservicesandserversinthe context of web applications.
	2. Understandidentitymanagementprinciplesandimplement secureidentitypractices withinwebservices.
	3. Applysecuritymeasuresto preventand detect unauthorized access attempts by externalentities.
Course Outline	**SecuringWebApplication** Services and Servers Introduction,Basic security for HTTP Applicationsand Services, Basic Security for SOAPServices,IdentityManagementandWebServices, Authorization Patterns, SecurityConsiderations, Challenges. **IntrusionDetectionandPrevention** Intrusion, Physical Theft, Abuse ofPrivileges, UnauthorizedAccessbyOutsider,Malwareinfection, Intrusion detection and PreventionTechniques, Anti-Malware software,Network based Intrusion detectionSystems,NetworkbasedIntrusionPrevention Systems, Host basedIntrusion prevention Systems, Security Information Management,Network Session Analysis, SystemIntegrity Validation
Module3CryptographyandNetwork Security(Credit1)	
Learning Outcomes	**Afterlearningthemodule,learnerswill beableto**
	1. Define cryptography and explain its role in securing information and communication.
	2. Differentiate between symmetric and asymmetric key cryptography, and understand their applications in securing data.
	3. Understand and apply cryptography in various applications, demonstrating proficiency in securing data in different contexts.
Course Outline	**CryptographyandNetworkSecurity:**

	Introduction to Cryptography, Symmetric key Cryptography, Asymmetric key Cryptography, Message Authentication, Digital Signatures, Applications of Cryptography. Overview of Firewalls-Types of Firewalls, User Management, VPN Security, Security Protocols:-security at the Application Layer-PGP and S/MIME, Security at Transport Layer-SSL And TLS, Security at Network Layer-IPSec.
Module4Cyberjurisprudence&Forensic(Credit1)	
Learning Outcomes	1. Analyze the specificities of theIndian cyber space, includingitsregulatoryframework,challenges,and initiatives.
	2. DefineCyberForensicsandunderstanditssignificancein investigatingcybercrimesanddigitalincidents.
	3. Demonstratetheabilitytoinitiateandconductpreliminary investigationsinresponsetosuspectedcyberincidents, ensuring the preservation ofdigital evidence.
Course Outline	**CyberspaceandtheLaw:** Introduction, Cyber SecurityRegulations, Roles of InternationalLaw,thestateandPrivateSectorinCyber space, Cyber SecurityStandards. The INDIAN Cyber space, National Cyber SecurityPolicy 2013. **Cyber Forensics** Introduction to Cyber Forensics,Handling Preliminary Investigations,Controlling an Investigation,Conducting disk-based analysis,Investigating Information-hiding, ScrutinizingE-mail,ValidatingE-mailheaderinformation, TracingInternet access, Tracing memory inreal-time.

Evaluation Scheme:

Internal Assessment:

Depending on the course content mentioned above a case study/assignments/presentations should be developed of 25 marks. Along with a Unit test of 25 marks should be conducted.

External Assessment: End Semester examination of 50 marks for 2 hours duration will be conducted by the university.

References:

1. Godbole, N. & Belapure, S. *Cyber Security: Understanding Cyber Crimes, Computer Forensics and Legal Perspectives*, Wiley India.
2. Stallings, W. *Cryptography and Network Security: Principles and Practice*, Pearson.

3. Stallings, W. & Brown, L. *Computer Security: Principles and Practice*, Pearson.
4. Whitman, M. E. & Mattord, H. J. *Principles of Information Security*, Cengage.
5. Schneier, B. *Applied Cryptography*, Wiley.
6. Casey, E. *Digital Evidence and Computer Crime*, Academic Press.
7. Stuttard, D. & Pinto, M. *The Web Application Hacker's Handbook*, Wiley.
8. National Cyber Security Policy 2013, Government of India.
9. NIST Cybersecurity Framework Documentation.
10. OWASP Top 10 and OWASP Web Security Resources.

Course Code	**3.2**
Course Title	**MobileApplicationDevelopmentTechnologies**
Course Credits	**2 [2 Th]**
Course Outcomes	**After Completion of this Course, students will be able to** **1.** To provide students with a solid understanding of the mobile app development,Androidoperatingsystem,itsarchitecture,components,andthe software development kit (SDK). **2.**ToteachstudentshowtobuildAndroidapplicationsfromscratch, including UI design, handling user interactions, and integrating various features. **3.** To learn about Android's UI components, layouts, and design principles to create visually appealing and user-friendly interfaces. **4.** To know various methods of data storage in Android applications, such as using SQLite databases, shared preferences, and cloud-based solutions**.**
Module1: IntroductionMobileTechnologies&FundamentalsofAndroid Programming (Credit 0.5)	
Learning Outcomes	**After learning this module, learners will be able to** 1. Understandtheevolutionandscopeofmobiletechnologies&Identifyand compare various mobile operating systems. 2. DescribeAndroidarchitectureanditskeycomponents&Basic UserInterface with UI Widgets
Content Outline	Introduction to MobileComputing-Features, Advantages, Disadvantagesand Applications , Factors in Developing Mobile Applications , Mobile Apps and TypesofMobileApps,MobileAppsDesign&DevelopmentProcess,Mobile Operating System: IOS, BlackBerry, Android, Windows Phone, PalmOS, SymbianOS. IntroductiontoAndroid- OverviewandEvolutionofAndroid,Featuresof Android,AndroidArchitecture,AndroidEnvironmentSetupAndroid-SDK, Emulators /Android AVD , First Android Application., Introduction to

	ComponentsofanAndroidApplication,ResourcesandManifestFile, Android App / Project Folder Structure
Module2: AndroidActivity,Intents,andServices,UILayoutsandGUIDesign (Credit 0.5)	
Learning Outcomes	**After learning this module, learners will be able to** 1. Formulate and solve optimization problems using techniques such as Linear Programming, the Simplex Method, assignment, and transportation models. 2. Apply network-based project management techniques like PERT and Critical Path Method to plan, schedule, and optimize resources in complex projects. 3. Evaluate optimization results using sensitivity analysis and software tools, and analyze trade-offs through network compression and cost-time considerations.
Content Outline	AndroidActivityandAndroidActivitylifeCycle,Toastin,Intents& Types, Android Services, Fragments, View, View Groups-Linear Layout, Relative Layout, Table Layout, Frame Layout, Web View, List View, Grid View ,Android UIControls – TextView,EditText,AutoCompleteTextView, Button, ImageButton, ToggleButton, CheckBox, RadioButton, RadioGroup, ProgressBar,Spinner,TimePicker,DatePicker,SeekBar,AlertDialog,Switch, RatingBar , Event-driven Programming in Android, List and Adaptors
Module 3: Android Menus, Threads, Notification and Alarms (Credit 0.5)	
Learning Outcomes	**After learning the module, learners will be able to** 1. Apply knowledge of error metrics to interpret their implications in the context of regression problems, understanding their significance in model evaluation. 2. Analyse factors that contribute to overfitting in regression models, evaluating their impact on model performance and generalization..
Content Outline	Creating a splash screen, Threads in Android,Threads running on UI thread (runOnUiThread),Workerthread,Handlers&Runnable,AsynTask (indetail) AndroidMenus-Options,Context,Popup5.5.AndroidNotification-Progress and Push

Module 4: ContentProviders,BroadcastReceivers&AdvancedProgramming (Credit 0.5)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Understandthedecisionmakingprocessbasedontheproximityof instances. 2. Definedecisiontreesandtheirroleinclassificationandregression.• Understand the process of tree construction and decision-making.
Content Outline	BasicoperationofSQLiteDatabase,AndroidApplicationPriorities, .Android Content Providers – SQLite Programming : Open Helper and create the database,openandcloseadatabase,andinsert,update,anddeleteoperationin database , Android BroadcastReceiver, Accessing Phone Service (Call, SMS, MMS),Email, Location-based services, Storage in Android-Shared Preferences,InternalandExternalStorage,.MultimediainAndroid– Android Camera, Audio Player. Video player , Android Bluetooth, Android WiFi, Android Sensors

Evaluation Scheme:

External Assessment: End Semester examination of 50 marks for 2 hours duration will be conducted by the university.

References Book:

1. ProfessionalAndroidApplicationDevelopmentbyRetoMeier,WileyIndiaPvtLtd publication.
2. AndroidCookbookbyIanF.DarwinO‟ReillyMedia,Inc.
3. BeginningAndroidbyMarkL.Murphy,WileyIndiaPvtLtdpublication.
4. ProfessionalAndroidbySayedYHashimiandSatyaKomatineni,WileyIndiaPvtLtd publication.
5. BuildingAndroidAppsbyineasySteps,McGraw-HillEducationpublication.6.20Recipes forProgramming PhoneGap: Cross-Platform Mobile Development for Android and iPhoneby Jamie Munro O'Reilly Media

Course Code	**3.3**
Course Title	**CloudComputing**
Course Credits	**2 [2 Th]**
Course Outcomes	**After Completion of this Course, students will be able to** 1. Identifysecurityaspects ofeach cloud model. 2. Develop a risk-management strategy for migrating to the Cloud. Implement a public cloud instance with a public cloud service provider. 3. Applyatrust-basedsecuritymodeltodifferentlayers.
Module1: IntroductiontoCloudComputing(Credit 0.5)	
Learning Outcomes	**After learning this module, learners will be able to** 1. Define cloud computing and its key characteristics, service models, and deployment models. 2. Provide an overview of the historical development of cloud computing and its evolution from traditional models. 3. Compare major cloud service providers like AWS, Azure, and GCP. 4. Identify and analyze potential security risks and challenges in cloud computing.
Content Outline	IntroductiontoCloudComputing OnlineSocial Networksand Applications Cloudintroductionandoverview Differentclouds,Risks,Novelapplicationsofcloudcomputing
Module2: CloudComputingArchitecture,CloudDeploymentModels (Credit 0.5)	
Learning Outcomes	**After learning this module, learners will be able to** 1. Define the requirements driving the emergence of cloud computing and explain CPU virtualization's role. 2. Provide an overview of basic cloud computing principles, discuss hypervisors, and explain the SPI framework 3. Identify key drivers motivating cloud adoption and assess the impact on end-users and businesses. 4. Explore best practices for establishing effective governance structures in cloud environments.

Content Outline	CloudComputingArchitecture:Requirements,Introduction Cloud computing architecture, On Demand Computing Virtualization at the infrastructure level, Security in Cloud computing environments, CPU Virtualization, A discussion on Hypervisors Storage Virtualization Cloud Computing Defined, The SPI Framework for Cloud Computing, The Traditional Software Model, The Cloud Services Delivery Model Cloud Deployment Models: KeyDriversto Adopting theCloud, The Impact of Cloud Computing on Users, Governance in the Cloud, Barriers to Cloud Computing Adoption in the Enterprise
Module 3: SecurityIssuesinCloudComputingandAccessmanagement(Credit 0.5)	
Learning Outcomes	**After learning the module, learners will be able to** 1. Apply knowledge of infrastructure security in cloud computing to identify key components involved. 2. Analyse network-level security measures and protocols relevant to cloud environments. 3. Evaluate application-level security practices and challenges 4. specific to cloud-based applications. 5. Design strategies to ensure data security and storage in cloud computing environments.
Content Outline	Security Issues in Cloud Computing: Infrastructure Security, Infrastructure Security: The Network Level, The Host Level, The Application Level, Data Security and Storage, Aspects of Data Security, Data Security Mitigation Provider Data and Its Security. Identity and Access Management: Trust Boundaries and IAM, IAM Challenges, Relevant IAM Standards and Protocols for Cloud Services, IAM Practices in the Cloud, Cloud Authorization Management.
Module 4: SecurityManagementintheCloud,PrivacyIssues(Credit 0.5)	
Learning Outcomes	**After learning the module, learners will be able to**

	1. Applysecuritymanagementstandardsrelevanttoc loud computing to ensure robust security measures. 2. AnalyseavailabilitymanagementpracticesforSaaS,PaaS,and IaaStooptimizeserviceavailability. 3. Evaluateriskassessmentsspecifictocloudsecurityandprop ose effective mitigation strategies. 4. Designandimplementtailoredincidentresponseplansforc loud computing scenarios to ensure swift and effective responses.
Content Outline	**SecurityManagementintheCloud:** SecurityManagement Standards,SecurityManagementintheCloud,Availability Management: SaaS, PaaS, IaaS. **Privacy Issues:** Privacy Issues, Data Life Cycle, Key Privacy ConcernsintheCloud,ProtectingPrivacy,ChangestoPrivacy Risk Management and Compliance in Relation to Cloud Computing,LegalandRegulatoryImplications,U.S.Lawsand Regulations,InternationalLawsandRegulations.
Activities towards Comprehensive Continuous Evaluation (CCE)	

Evaluation Scheme:

External Assessment: End Semester examination of 50 marks for 2 hours duration will be conducted by the university.

References Book:

1. T.,Mahmood,Z.,&Puttini,R.(2013).CloudComputing:Concepts,Technology& Architecture. Prentice Hall.
2. Reese,G.(2009).CloudApplicationArchitectures:BuildingApplicationsand Infrastructure in the Cloud. O'Reilly Media.
3. Mather,T.,Kumaraswamy,S.,&Latif,S.(2009).CloudSecurityandPrivacy:An Enterprise Perspective on Risks and Compliance. O'Reilly Media.
4. Bahga,A.,&Madisetti,V.(2014).CloudComputing:AHands-On Approach.CreateSpace Independent Publishing Platform.

Course Code	**3.4**
Course Title	**Machine learning**
Course Credits	**2**
Course Outcomes	**After Completion of this Course, students will be able to** 1. Understand the foundational concepts and algorithms of Machine Learning. 2. Implement binary, multiclass classification and regression models using real-world datasets. 3. Apply logic-based and algebraic models to structured learning problems. 4. Develop end-to-end ML solutions including preprocessing, model selection, evaluation, and reporting. 5. Explore recent trends and advanced topics in Machine Learning
Module1: IntroductiontoMachine Learning(Credit 0.5)	
Learning Outcomes	**After learning this module, learners will be able to** 1. Explain basic concepts of supervised, unsupervised, and reinforcement learning. 2. Identify real-world problems where ML can be applied. 3. Understand data preprocessing techniques.
Content Outline	What is Machine Learning? Types and Applications , Supervised vs Unsupervised Learning, Data Cleaning, Feature Scaling, Label Encoding, Overfitting and Underfitting, Bias-Variance Tradeoff, Introduction to Scikit-Learn
Module2: Binary and Multiclass Classification (Credit 0.5)	
Learning Outcomes	**After learning this module, learners will be able to** 1. Build and evaluate classification models. 2. Handle imbalanced datasets and multiclass problems.
Content Outline	Binary Classification: Logistic Regression, Naïve Bayes, k-NN, Multiclass Classification: One-vs-Rest, One-vs-One strategies, Support Vector Machines (Linear & Kernel), Model Evaluation: Confusion Matrix, Accuracy, Precision, Recall, F1 Score, ROC-AUC, Cross-Validation and Grid Search

Module 3:Regression (Credit 0.5)	
Learning Outcomes	**After learning the module, learners will be able to** 1. Develop and evaluate regression models for prediction tasks.
Content Outline	Simple and Multiple Linear Regression, Polynomial Regression, Decision Tree Regression, Evaluation Metrics: MSE, RMSE, MAE, R^2 Score, Feature Selection & Regularization: Lasso, Ridge, ElasticNet
Module 4: Logic Based and Algebraic Models, Trends in Machine Learning(Credit 0.5)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Understand logic-based approaches and ensemble methods. 2. Identify and explore current trends in ML.
Content Outline	Decision Trees, Rule-Based Systems, Ensemble Methods: Bagging, Random Forest, Boosting (AdaBoost, XGBoost), Introduction to Clustering: K-Means, Hierarchical, Dimensionality Reduction: PCA, t-SNE, Trends: AutoML, Explainable AI (XAI), MLops Basics, Ethical AI

Evaluation Scheme:

External Assessment: End Semester examination of 50 marks for 2 hours duration will be conducted by the university.

References Book:

1. Tom Mitchell, Machine Learning, McGraw Hill

2. Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow, O'Reilly

3. Trevor Hastie, Robert Tibshirani, Jerome Friedman, The Elements of Statistical Learning

4. Jason Brownlee, Machine Learning Mastery with Python

5. Python and Scikit-learn official documentation

Course Code	**3.5(Elective 1 Sem III)**
Course Title	**Agentic AI**
Course Credits	**2**
Course Outcomes	**After the completion of this course, students will be able to:** 1. Demonstrate foundational Agentic AI concepts, architecture, and design patterns. 2. Design high-quality prompts to guide agent behavior using LLMs. 3. Use modern Python-based frameworks (CrewAI, Pydantic, OpenAI Agent Swarms) to build autonomous agents. 4. Construct collaborative multi-agent systems using dynamic planning and protocols like MCP & A2A. 5. Deploy end-to-end intelligent workflows using tools like N8N and external APIs. 6. Evaluate, troubleshoot, and ethically manage real-world Agentic AI applications.
Module1: Foundations of Generative AI, Agentic AI (Credit 0.5)	
Learning Outcomes	**After learning this module, learners will be able to** 1. Explain the fundamentals of Generative AI and Large Language Models. 2. Differentiate Agentic AI from traditional AI systems. 3. Design effective prompts for autonomous and multi-agent AI systems.
Content Outline	Understanding Generative AI, Overview of Large Language Models (LLMs), Different LLMs and their capabilities, Applications of Generative AI across various domains, Agentic AI vs. Traditional AI, Components of an Intelligent Agent: Goal Setting, Memory, Reasoning, Action, Autonomy, Tool Usage, Feedback Loops, Role Prompting, Chain-of-Thought (CoT) Prompting, ReAct Prompting, Zero-shot Prompting, Few-shot Prompting, Self-Reflection Prompting, System Prompts vs. User Prompts, Multi-Agent

	Prompt Design, Prompt Injection Attacks, Prompt Hygiene.
Module2:Agentic AI Design Patterns, Architectures and Multi-Agent Collaboration (Credit 0.5)	
Learning Outcomes	**After learning this module, learners will be able to** 1. Analyze various Agentic AI architectures and design patterns. 2. Design multi-agent systems using collaboration and communication strategies. 3. Implement task planning and interoperability mechanisms for autonomous agents.
Content Outline	Autonomy, Reactivity, Proactiveness, Social Ability, Reactive Agents, Deliberative Agents, Reflection Pattern, Tool Use Pattern, ReAct Framework, Planning Pattern, Multi-Agent Collaboration Pattern, Architectural Considerations for Robust Agentic Systems, Direct Communication, Mediated Communication, Structured Message Passing, Unstructured Message Passing, Internal APIs for Agent Communication, Shared Context and State Management, Blackboard Systems, Memory Clouds, Dynamic Task Assignment, DAG-Based Chaining, Priority Queues, Cooperative Planning, Self-Evaluating Planners, Recursive Delegation, Review Mechanisms, Model Context Protocol (MCP), Agent-to-Agent (A2A) Protocol, Secure Cross-Agent Communication Pipelines.
Module 3:End-to-End Agentic AI Pipelines and Workflow Automation (Credit 0.5)	
Learning Outcomes	**After learning the module, learners will be able to** 1. Develop automated workflows using N8N and agent orchestration tools. 2. Integrate agents with APIs, databases, and external applications. 3. Build scalable multi-agent pipelines for real-world automation tasks.
Content Outline	N8N Architecture and Features, Low-Code Workflow Automation, Triggers, Webhooks, Automation Nodes, Workflow-Based Agent Orchestration, API Integration, API Authentication, Secure Token Management, Database Connectivity, Data Retrieval from Emails, Slack and REST APIs, Agent-Controlled Automation, Document

	Summarization, Email Generation, Multi-Agent Workflow Design, Error Handling, Recovery Mechanisms, Retry Strategies, Fallback Agents, Event-Driven Execution, Scheduled Agentic Workflows, Enterprise Automation, Customer Support Systems, Research Assistants, Business Process Automation, Industry Case Studies.
Module 4: Testing, Optimization, Ethics and Governance of Agentic AI (Credit 0.5)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Evaluate the performance and efficiency of Agentic AI systems. 2. Apply debugging, monitoring, and control mechanisms to improve agent reliability. 3. Implement ethical, transparent, and responsible AI practices in autonomous agent development.
Content Outline	Task Success Rate, Latency, Throughput, Reasoning Depth, Prompt Entropy, Response Coherence, Factual Accuracy, Token Usage, Cost Optimization, Agent Monitoring, Sandboxing Techniques, Hallucination Mitigation, Tool Misuse Prevention, Prompt Versioning, Fallback Routing, Recovery Strategies, Ethics of Autonomous Decision-Making, Transparency, Explainability, Auditability, Human-in-the-Loop Design, Safety Guardrails for Autonomous Agents, Regulatory and Governance Considerations.

Evaluation Scheme:

External Assessment: End Semester examination of 50 marks for 2 hours duration will be conducted by the university.

References Book:

1. "Designing Autonomous Agents: Theory and Practice from Biology to

Engineering and Back" – Pattie Maes.

2. "Multi-Agent Systems: Algorithmic, Game-Theoretic, and Logical Foundations" –

Yoav Shoham and Kevin Leyton-Brown.

Course Code	**3.5(Elective 2 Sem III)**
COURSE TITLE	**BIG DATA ANALYTICS**
Course Credits	**2**
Course Outcomes	**After Completion of this Course, students will be able to** 1. Apply big data analytics approaches, including conceptualization, summarization, and machine learning techniques. 2. Analyze the characteristics of datasets to distinguish between trivial data and big data for various applications. 3. Evaluate solutions for problems related to batch learning, online learning, and big data characteristics like high dimensionality and scalability issues. 4. Design scalable solutions to manage and process dynamically growing big data.
Module1: Introduction (Credit 0.5)	
Learning Outcomes	**After learning this module, learners will be able to** 1. Apply understanding of fundamental Big Data concepts to address modern data challenges effectively. 2. Analyze traditional and Big Data approaches to select appropriate technologies for large dataset processing and analysis. 3. Evaluate infrastructure requirements for efficient Big Data handling, focusing on data analytics for valuable insights extraction. 4. Design strategies to overcome challenges inherent in Big Data, considering volume, velocity, and variety factors for effective data management and analysis.
Content Outline	Introduction to Big Data, Big Data Characteristics, Types of Big Data • Traditional Versus Big Data Approach, Technologies Available for Big Data • Infrastructure for Big Data, Use of Data Analytics, Big Data Challenges, Desired Properties of a Big Data System, Case Study of Big Data Solutions
Module2: Analytical Theory and Methods (Credit 0.5)	

Learning Outcomes	**After learning this module, learners will be able to** 1. Apply clustering algorithms, association rules, and the Apriori Algorithm to recognize patterns effectively across diverse datasets. 2. Analyse real-world scenarios by applying association rules, mastering validation, and testing for model reliability. 3. Evaluate linear and logistic regression models, implementing and interpreting regression analyses with proficiency. 4. Design additional regression models to enhance analytical capabilities for addressing diverse data-driven challenges.
Content Outline	Clustering and Associated Algorithms, Association Rules, Apriori Algorithm, Candidate Rules • Applications of Association Rules, Validation and Testing, Diagnostics, Regression, Linear Regression, Logistic Regression, Additional Regression Models
Module 3:Modern Big Data Processing Architecture - Spark (Credit 0.5)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Apply concepts of modern distributed data processing using Apache Spark 2. Analyze architecture of in-memory processing and unified analytics platforms. 3. Evaluate advantages over traditional systems in terms of speed, scalability, and real-time processing. 4. Design scalable big data pipelines using modern architectures
Content Outline	Introduction to Modern Big Data Architecture Limitations of Traditional Systems (Hadoop-based batch processing) Overview of Apache Spark Spark Architecture: Driver, Executors, Cluster Manager Resilient Distributed Datasets (RDDs), DataFrames, and Datasets Spark Components: Spark SQL, Spark Streaming, MLlib, GraphX Real-time Data Processing and Stream Analytics Data Pipelines and

	ETL using Spark
Module 4: NoSQL (Credit 0.5)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Apply comprehensive learning to grasp NoSQL, covering its definition, business drivers, case studies, and data architectural patterns for informed decision-making in data management. 2. Analyse the practical application of MapReduce within the new software stack, understanding its role and implementing algorithms for efficient big data processing. 3. Evaluate the effectiveness of MapReduce in handling largescale data processing tasks, considering its scalability and performance. 4. Design strategies to optimize MapReduce algorithms and workflows for specific big data processing requirements, ensuring efficient and effective data processing.
Content Outline	What is NoSQL?, NoSQL Business Drivers, NoSQL Case Studies, NoSQL Data Architectural Patterns, Variations of NoSQL Architectural Patterns, Using NoSQL to Manage Big Data **Map Reduce:**MapReduce and The New Software Stack, MapReduce, Algorithms Using MapReduce

Evaluation Scheme:

External Assessment: End Semester examination of 50 marks for 2 hours duration will be conducted by the university.

References:-

1.Shankarmani, R. (2016). Big Data Analytics. Wiley.

2. Chellappan, S., & Acharya, S. (2015). Big Data and Analytics. Wiley.

3. Prajapati, V. (2013). Big Data Analytics with R and Hadoop. Pack Publishing.

4. Dasgupta, N. (2018). Practical Big Data Analytics. Pack Publishing

Course Code	**3.5(Elective 3 Sem III)**
Course Title	**Data Security and Privacy**
Course Credits	**2**
Course Outcomes	**After Completion of this Course, students will be able to** 1. Demonstrate an understanding of fundamental concepts, principles, and challenges of data security and privacy in modern information systems. 2. Apply cryptographic techniques and access control mechanisms to protect data confidentiality, integrity, and availability. 3. Analyze privacy risks, threats, and vulnerabilities in data-driven environments and propose appropriate mitigation strategies. 4. Evaluate security and privacy frameworks, regulations, and emerging technologies for safeguarding organizational and personal data.
Module1: Foundations of Data Security and Privacy (Credit 0.5)	
Learning Outcomes	**After learning this module, learners will be able to** 1. Understand the fundamental concepts of information security and privacy. 2. Explain security goals such as confidentiality, integrity, availability, authentication, and non-repudiation. 3. Identify common security threats, vulnerabilities, and attacks. 4. Understand privacy principles and challenges in the digital ecosystem.
Content Outline	Introduction to Data Security and Privacy, Information Security Fundamentals, Security Goals (CIA Triad), Authentication, Authorization and Accountability, Threats, Vulnerabilities and Risk Assessment, Security Attacks and Countermeasures, Introduction to Privacy, Privacy Principles and Models, Data Lifecycle and Privacy Challenges, Security and Privacy in Digital Environments.
Module 2: Cryptography and Access Control Mechanisms (Credit 0.5)	

Learning Outcomes	**After learning this module, learners will be able to** 1. Apply cryptographic techniques to secure data. 2. Differentiate between symmetric and asymmetric encryption methods. 3. Understand digital signatures, certificates, and key management. 4. Implement access control mechanisms for data protection.
Content Outline	Introduction to Cryptography, Classical and Modern Cryptographic Techniques, Symmetric Key Cryptography, Public Key Cryptography, Hash Functions and Message Authentication Codes, Digital Signatures and Certificates, Public Key Infrastructure (PKI), Key Management Techniques, Access Control Models (DAC, MAC, RBAC, ABAC), Identity and Access Management (IAM), Multi-factor Authentication.
Module 3: Data Privacy, Protection and Compliance (Credit 0.5)	
Learning Outcomes	**After learning the module, learners will be able to** 1. Analyze privacy risks associated with data collection, storage, and sharing. 2. Apply privacy-preserving techniques to protect sensitive information. 3. Understand legal and regulatory frameworks governing data protection. 4. Evaluate ethical considerations in data management and analytics
Content Outline	Data Privacy Concepts, Personally Identifiable Information (PII), Privacy Threats and Risks, Data Anonymization and Pseudonymization, Privacy-Preserving Data Publishing, Differential Privacy Fundamentals, Data Governance and Privacy Management, Data Protection Regulations (GDPR, Digital Personal Data Protection Act (DPDP), HIPAA Overview), Privacy Impact Assessment, Ethical Issues in Data Collection and Usage.
Module 4: Emerging Trends in Data Security and Privacy (Credit 0.5)	

Learning Outcomes	After learning the module, learners will be able to
	1. Assess security challenges in cloud computing, IoT, and big data environments. 2. Analyze privacy concerns in artificial intelligence and machine learning systems. 3. Evaluate security incidents and breach case studies. 4. Propose comprehensive security and privacy solutions for modern applications.
Content Outline	Cloud Security and Privacy, Security in Big Data Environments, Internet of Things (IoT) Security Challenges, AI and Machine Learning Privacy Issues, Data Breaches and Incident Response, Security Monitoring and Auditing, Privacy-Enhancing Technologies (PETs), Blockchain for Data Security, Zero Trust Security Model, Case Studies and Emerging Research Trends in Data Security and Privacy.

Evaluation Scheme:

External Assessment: End Semester examination of 50 marks for 2 hours duration will be conducted by the university.

References

1. Nina Godbole and Sunit Belapure, *Cyber Security: Understanding Cyber Crimes, Computer Forensics and Legal Perspectives*, Wiley.
2. William Stallings, *Cryptography and Network Security: Principles and Practice*, Pearson.
3. Charles P. Pfleeger and Shari Lawrence Pfleeger, *Security in Computing*, Pearson.
4. Jonathan Katz and Yehuda Lindell, *Introduction to Modern Cryptography*, CRC Press.
5. Paul DuBois, Michael Schroeck, and others, *Data Privacy and Protection: Concepts and Practices*.
6. Arshdeep Bahga and Vijay Madisetti, *Internet of Things – A Hands-On Approach*, Universities Press.
7. OECD Privacy Guidelines and GDPR Documentation.

Course Code	**3.5(Elective 4 Sem III)**
Course Title	**Computer Vision**
Course Credits	**2**
Course Outcomes	**After Completion of this Course, students will be able to** 1. Understand the fundamentals of deep learning and their application to computer vision tasks. 2. Apply image processing and feature extraction techniques for visual analysis and recognition. 3. Analyze geometric relationships in images and implement 3D scene reconstruction methods. 4. Design and evaluate deep learning-based solutions for image segmentation, object detection, and other advanced computer vision applications.
Module1: Deep Learning Foundations for Computer Vision (Credit 0.5)	
Learning Outcomes	**After learning this module, learners will be able to** 1. Understand the fundamentals of deep learning and neural network architectures. 2. Explain learning algorithms such as gradient descent and backpropagation. 3. Apply regularization and preprocessing techniques to improve model performance. 4. Design and implement CNN and RNN models for visual data analysis.
Content Outline	Introduction to Computer Vision and Deep Learning, Artificial Neurons and Neural Networks, Multilayer Perceptrons (MLP), Gradient Descent Optimization, Backpropagation in MLPs, Optimization Techniques, Regularization Methods, Data Preprocessing, Convolutional Neural Networks (CNNs), CNN Properties and Architectures, Introduction to Recurrent Neural Networks (RNNs), Encoder-Decoder Architectures in RNNs, Applications of Deep Learning in Computer Vision.
Module 2: Image Processing and Feature Extraction (Credit 0.5)	
Learning Outcomes	**After learning this module, learners will be able to** 1. Apply image filtering and enhancement techniques in the spatial and frequency domains.

	2. Detect edges, lines, corners, and salient image features. 3. Analyze local feature descriptors for image matching and recognition. 4. Implement feature-based vision algorithms for image understanding.
Content Outline	Low-Level Vision, Spatial Domain Filtering, Frequency Domain Filtering, Edge Detection Techniques, Line Detection Methods, Feature Detection and Extraction, Harris Corner Detector, Blob Detection Techniques, Scale Invariant Feature Transform (SIFT), Feature Descriptors, Speeded-Up Robust Features (SURF), Feature Matching and Applications in Computer Vision.
Module 3: Geometric Computer Vision and 3D Reconstruction (Credit 0.5)	
Learning Outcomes	**After learning the module, learners will be able to** 1. Understand image formation and geometric transformations. 2. Analyze camera models and geometric relationships between multiple views. 3. Compute epipolar geometry and fundamental matrices. 4. Develop 3D reconstruction systems using structure-from-motion techniques.
Content Outline	Single-View Geometry, 2D Geometric Transformations, Camera Models, Camera Intrinsic and Extrinsic Parameters, Two-View Stereo Vision, Epipolar Geometry, Algebraic Representation of Epipolar Constraints, Fundamental Matrix Computation, Structure from Motion (SfM), Batch Processing in SfM, Dense 3D Reconstruction, Applications in Robotics, Autonomous Systems, and Augmented Reality.
Module 4: Advanced Vision Applications using Deep Learning (Credit 0.5)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Apply deep neural networks for stereo vision and scene understanding. 2. Implement image segmentation techniques using classical and deep learning approaches. 3. Develop object detection systems using modern deep learning architectures. 4. Evaluate advanced computer vision solutions for real-world applications.

Content Outline	Deep Neural Networks for Stereo Vision and Structure from Motion, Mid-Level Vision Concepts, Image Segmentation Techniques, Semantic and Instance Segmentation, Deep Learning-based Segmentation Models, High-Level Vision, Object Recognition and Scene Understanding, Deep Learning-based Object Detection, Region-Based and Single-Shot Detection Methods, Performance Evaluation Metrics, Recent Advances and Applications in Computer Vision.

Evaluation Scheme:

External Assessment: End Semester examination of 50 marks for 2 hours duration will be conducted by the university.

References

1. Richard Szeliski, *Computer Vision: Algorithms and Applications*, Springer.
2. David A. Forsyth and Jean Ponce, *Computer Vision: A Modern Approach*, Pearson.
3. Simon J. D. Prince, *Computer Vision: Models, Learning and Inference*, Cambridge University Press.
4. Rafael C. Gonzalez and Richard E. Woods, *Digital Image Processing*, Pearson.
5. Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep Learning*, MIT Press.
6. Adrian Kaehler and Gary Bradski, *Learning OpenCV 4 Computer Vision with Python*, O'Reilly.
7. Selected Research Papers from CVPR, ICCV, ECCV, NeurIPS, and ICLR.

Course Code	**3.5(Elective 5 Sem III)**
Course Title	**Design for Scalability and Reliability**
Course Credits	**2**
Course Objective	**After the completion of this course, students will be able to:** 1. Introduce the fundamental concepts of scalability, reliability, availability, fault tolerance, and performance in modern software systems. 2. Develop an understanding of architectural principles and design patterns used to build scalable and resilient applications. 3. Provide knowledge of techniques such as load balancing, caching, replication, partitioning, and distributed data management for handling large-scale workloads. 4. Equip learners with the skills required to analyze system failures and implement fault tolerance, monitoring, recovery, and high-availability mechanisms. 5. Familiarize learners with cloud-native architectures, microservices, distributed systems, and Site Reliability Engineering (SRE) practices. 6. Enable the design and evaluation of scalable, reliable, and maintainable software systems capable of meeting real-world performance and availability requirements. 7. Promote the application of scalability and reliability principles in the development of enterprise, cloud-based, and distributed applications.
Course Outcomes	**After the completion of this course, students will be able to:** 1. 1. Explain the principles of scalability, reliability, availability, and fault tolerance in modern software systems. 2. Analyze system architectures and identify design strategies to handle increasing workloads and performance demands. 3. Apply techniques such as load balancing, caching, replication, and distributed data management to build scalable systems. 4. Design reliable systems using redundancy, fault tolerance, monitoring, and recovery mechanisms. 5. Evaluate and propose scalable and reliable architectures for real-world distributed and cloud-based applications.
Module1: Fundamentals of Scalability and Reliability (Credit 0.5)	

Learning Outcomes	**After learning this module, learners will be able to** 1. Define scalability, reliability, availability, maintainability, and fault tolerance. 2. Differentiate between vertical and horizontal scaling approaches. 3. Explain challenges associated with large-scale distributed systems. 4. Identify architectural factors affecting system performance and reliability.
Content Outline	**Introduction to Scalable and Reliable Systems** Evolution of Large-Scale Systems,Characteristics of Modern Distributed Systems, Scalability: Concepts and Types, Reliability and Availability,Maintainability and Serviceability, Fault Tolerance and Resilience **Performance Fundamentals** Throughput, Latency, Response Time, Performance Bottlenecks, Capacity Planning,Service Level Agreements (SLAs), Service Level Objectives (SLOs), Service Level Indicators (SLIs) **Architectural Foundations** Monolithic vs Distributed Architectures,Client-Server Architecture, Microservices Architecture, Event-Driven Architecture ,Scalability Challenges in Modern Applications
Module2: Designing Scalable Systems (Credit 0.5)	
Learning Outcomes	**After learning this module, learners will be able to** 1. Apply architectural patterns for scalability. 2. Design systems capable of handling increasing user and data loads. 3. Utilize caching and load balancing techniques effectively. 4. Evaluate data storage strategies for scalable applications.
Content Outline	**Scaling Techniques** Vertical Scaling,Horizontal Scaling,Stateless and Stateful Services, Partitioning and Sharding , **Load Distribution** Load Balancing Concepts,Load Balancing Algorithms,Reverse

	Proxies, Traffic Distribution Strategies **Performance Optimization** Caching Fundamentals,Client-Side and Server-Side Caching,Content Delivery Networks (CDNs),Database Query Optimization **Scalable Data Management** Database Replication,Data Partitioning,Distributed Databases,NoSQL Databases, Eventual Consistency
Module 3: Designing Reliable and Fault-Tolerant Systems (Credit 0.5)	
Learning Outcomes	**After learning the module, learners will be able to** 1. Design systems that continue functioning despite failures. 2. Implement redundancy and fault tolerance mechanisms. 3. Analyze failure scenarios and recovery strategies. 4. Apply monitoring and observability techniques for system reliability.
Content Outline	**Reliability Engineering** Reliability Metrics, Mean Time Between Failures (MTBF), Mean Time To Recovery (MTTR), Failure Analysis **Fault Tolerance Techniques** Redundancy Models, Active-Active and Active-Passive Architectures, Replication Strategies, Checkpointing and Recovery **High Availability** Failover Mechanisms, Disaster Recovery Planning, Backup and Restore Strategies, Business Continuity Planning **Monitoring and Observability** Logging, Metrics Collection, Distributed Tracing, Alerting Systems, Health Checks and Heartbeats
Module 4: Cloud-Native and Large-Scale System Design (Credit 0.5)	

Learning Outcomes	After learning the module, learners will be able to
	1. Design cloud-native applications with scalability and reliability considerations. 2. Apply modern architectural patterns for large-scale systems. 3. Evaluate trade-offs in distributed system design. 4. Develop scalable and resilient architectures for real-world applications.
Content Outline	**Cloud-Native Architecture** Cloud Computing Fundamentals, Containers and Containerization, Container Orchestration Concepts, Infrastructure as Code (IaC) **Distributed System Design** CAP Theorem, Consistency Models, Distributed Transactions, Event Streaming Architecture **Reliability in Cloud Environments** Auto Scaling, Self-Healing Systems, Chaos Engineering, Site Reliability Engineering (SRE) Concepts **Case Studies** Scalable E-Commerce Systems, Social Media Platforms, Video Streaming Services, Ride-Sharing Applications, Designing Large-Scale Web Applications

Evaluation Scheme:

External Assessment: End Semester examination of 50 marks for 2 hours duration will be conducted by the university.

References

1. Designing Data-Intensive Applications by Martin Kleppmann, O'Reilly Media, 2017.
2. Web Scalability for Startup Engineers by Artur Ejsmont, McGraw-Hill Education, 2015.
3. Distributed Systems: Principles and Paradigms by Andrew S. Tanenbaum and Maarten Van Steen, 3rd Edition, Pearson Education, 2017.
4. Site Reliability Engineering by Google, O'Reilly Media, 2016.
5. Building Microservices by Sam Newman, 2nd Edition, O'Reilly Media, 2021.

6. The Art of Scalability by Martin L. Abbott and Michael T. Fisher, 2nd Edition, Addison-Wesley, 2015.
7. Release It! by Michael T. Nygard, 2nd Edition, Pragmatic Bookshelf, 2018.
8. Cloud Native Patterns by Cornelia Davis, Manning Publications, 2019.

Course Code	**3.6**
Course Title	**Mobile Application Development Technologies Lab**
Course Credits	**2**

A course-based project relevant to the concepts covered in the subject shall be implemented and evaluated. The project should demonstrate the practical application of the techniques, and methodologies learned in the course using appropriate programming languages, tools, and technologies.

Evaluation Scheme:

Internal Evaluation (25 marks):

The project should demonstrate the application of theoretical knowledge to solve a real-world problem through system design, implementation and performance evaluation.

The Internal Assessment for the project shall be conducted based on project implementation, research contribution, project presentation, and report writing, as per the following rubric. Evaluation done at Institute level.

The Rubric will have the following Evaluation Parameters:

The Rubric will have the following Evaluation Parameters:	
Evaluation Parameters	**Marks**
Problem Understanding & Formulation	10
Implementation & Functionality	20
Knowledge Representation / Logic Design	10
Analysis & Result Interpretation	5

Report writing or Documentation	5

External Examination – 25 marks Presentation (internal and external examiners involved)

Course Code	**3.7**
Course Title	**Cloud Computing Lab**
Course Credits	**2**

A course-based project relevant to the concepts covered in the subject shall be implemented and evaluated. The project should demonstrate the practical application of the techniques, and methodologies learned in the course using appropriate programming languages, tools, and technologies.

Evaluation Scheme:

Internal Evaluation (25 marks):

The project should demonstrate the application of theoretical knowledge to solve a real-world problem through system design, implementation and performance evaluation.

The Internal Assessment for the project shall be conducted based on project implementation, research contribution, project presentation, and report writing, as per the following rubric. Evaluation done at Institute level.

The Rubric will have the following Evaluation Parameters:

The Rubric will have the following Evaluation Parameters:	
Evaluation Parameters	**Marks**
Problem Understanding & Formulation	10
Implementation & Functionality	20
Knowledge Representation / Logic Design	10
Analysis & Result Interpretation	5

Report writing or Documentation	5

External Examination – 25 marks Presentation (internal and external examiners involved)

Course Code	**3.8**
Course Title	**Machine learning Lab**
Course Credits	**2**

A course-based project relevant to the concepts covered in the subject shall be implemented and evaluated. The project should demonstrate the practical application of the techniques, and methodologies learned in the course using appropriate programming languages, tools, and technologies.

Evaluation Scheme:

Internal Evaluation (25 marks):

The project should demonstrate the application of theoretical knowledge to solve a real-world problem through system design, implementation and performance evaluation.

The Internal Assessment for the project shall be conducted based on project implementation, research contribution, project presentation, and report writing, as per the following rubric. Evaluation done at Institute level.

The Rubric will have the following Evaluation Parameters:

The Rubric will have the following Evaluation Parameters:	
Evaluation Parameters	**Marks**
Problem Understanding & Formulation	10
Implementation & Functionality	20
Knowledge Representation / Logic Design	10
Analysis & Result Interpretation	5

Report writing or Documentation	5

External Examination – 25 marks Presentation (internal and external examiners involved)

Course Code	**3.9**
Course Title	**Elective II Lab**
Course Credits	**2**

A course-based project relevant to the concepts covered in the subject shall be implemented and evaluated. The project should demonstrate the practical application of the techniques, and methodologies learned in the course using appropriate programming languages, tools, and technologies.

Evaluation Scheme:

Internal Evaluation (25 marks):

The project should demonstrate the application of theoretical knowledge to solve a real-world problem through system design, implementation and performance evaluation.

The Internal Assessment for the project shall be conducted based on project implementation, research contribution, project presentation, and report writing, as per the following rubric. Evaluation done at Institute level.

The Rubric will have the following Evaluation Parameters:

The Rubric will have the following Evaluation Parameters:	
Evaluation Parameters	**Marks**
Problem Understanding & Formulation	10
Implementation & Functionality	20
Knowledge Representation / Logic Design	10
Analysis & Result Interpretation	5

Report writing or Documentation	5

External Examination – 25 marks Presentation (internal and external examiners involved)

Course Code	**3.10**
Course Title	**Mini Project**
Course Credits	**4**

Mini Project Evaluation Scheme:

Internal Evaluation (50 marks):

In alignment with the courses offered during the semester, namely Cyber Security, Mobile Application Development Technologies, Cloud Computing, and Machine Learning, learners shall undertake a project that integrates concepts, methodologies, tools, and technologies from one or more of these courses. The project should demonstrate the application of theoretical knowledge to solve a real-world problem through system design, implementation, experimentation, and evaluation.

The project shall involve:

- Identification and formulation of a relevant problem statement addressing practical or societal challenges.
- Review of literature, existing solutions, industry practices, and recent advancements related to the selected domain.
- Design and development of a secure, scalable, and efficient solution using appropriate technologies and frameworks.
- Development of software applications using suitable mobile application development platforms and programming languages.
- Utilization of cloud computing services and deployment models for hosting, storage, scalability, and resource management, wherever applicable.
- Incorporation of machine learning techniques for data analysis, prediction, classification, recommendation, pattern recognition, or intelligent decision-making, wherever relevant.
- Implementation of cybersecurity principles such as authentication, authorization, encryption, secure communication, vulnerability assessment, and data protection mechanisms.

- Testing, validation, performance evaluation, and security assessment of the developed solution using appropriate metrics and methodologies.
- Preparation of a comprehensive project report documenting objectives, methodology, implementation details, experimental results, and conclusions.
- Presentation and demonstration of the project before the evaluation committee.

The Internal Assessment shall be conducted based on innovation, technical complexity, implementation quality, security considerations, research contribution, project presentation, and report writing, as per the prescribed rubric.

Learners are encouraged to publish or present a research paper based on their project work in a National or International Conference, Seminar, Symposium, or Journal. Submission of the manuscript, acceptance letter, publication, or proof of presentation may be considered as an additional indicator of research contribution and academic engagement.

The Rubric will have the following Evaluation Parameters:

Evaluation Parameters	**Marks**
Problem Identification & Objectives	10
Literature Review & Research	10
System Design & Methodology	10
Project Implementation	10
Report writing or Documentation	5
Conference paper presentation	5

External Examination – 50 marks Presentation (internal and external examiners involved)

Semester: IV

Course Code	**4.1**
Course Title	**MOOC**
Course Credits	**4 [4 Th]**
Course Outcomes	**1.** Apply acquired knowledge and skills to solve real-world problems using appropriate tools and techniques. **2.** Analyze and evaluate different approaches, methods, or solutions within the domain to make informed decisions.
MOOC Courses to be selected from the repository of NPTEL/SWAYAM/MOOC/ etc.	

Evaluation Scheme:

External Assessment:

End Semester examination of 100 marks for 3 hours duration will be conducted by university.

Course Code	**4.2**
Course Title	**Internship**
Course Credits	**20 [20 Pr]**
Course Objectives	1.To give the first-hand experience of analysis, design, implementation and documentation of relevant projects. 2.Analysis of the existing system, Investigating alternatives, Design of a computer-based system Documentation, User training.

Evaluation Scheme:

Internal Assessment: 250 Marks

Internship Project Presentation continuous assessment will be conducted for 250 marks.

External Assessment: 250 Marks

End Semester Internship Project Presentation of 250 marks will be conducted at university level.