SNDT Women's University, Mumbai

Undergraduate Degree / UG Programme (Syllabus as Per NEP) - Faculty of Science & Technology

Bachelor of Computer Application

(B.C.A.)

As Per NEP – 2020

Semester – V & VI

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

Terminologies

Vertical	Full-Form/Definition	Remarks	Related To Major And Minor Courses
Major (Core)	Subject Comprising Mandatory and Elective Courses, Major Specific IKS, Vocational Skill Courses, Internship/ Apprenticeship, Field Projects, Research Projects Connected to Major	Minimum 50% Of Total Credits Corresponding to Three/Four - Year UG Degree- Mandatory Courses	Related To The Major
Minor Course	Course From Same Or Different Faculty	Minimum 18-20 Credits to Be Completed in The First Three Years of UG Programme	Related To the Minor
OEC	Open Elective Courses/ Generic Courses	10-12 Credits to Be Offered in I And/Or II Year. Faculty-Wise Baskets of OEC To Be Prepared	OEC Is to Be Chosen Compulsorily from Faculty Other Than That of the Major
VSC	Vocational Skill Courses, Including Hands On Training Corresponding To The Major And/Or Minor Subject	8-10 Credits, To Be Offered in First Three Years, Wherever Applicable Vocational Courses Will Include Skills Based on Advanced Laboratory Practical's of Major	Related To the Major or Minor
SEC	Skill Enhancement Courses	06 Credits, To Be Offered in I And II Year, To Be Selected from The Basket of Skill Courses Approved by University	Related To the Major or Minor Any Relevant Skill
AEC	Ability Enhancement Courses	08 Credits, To Be Offered in I And II Year, English: 04 Credits to Be Earned in Sem - I, Modern Indian Language Of 04 Credits to Be Offered in II Year	NA
VEC	Value Education Courses	Understanding India, Environmental Science/Education, Digital and Technological Solutions, Health & Wellness, Yoga Education,	NA

		Sports, And Fitness	
IKS	Indian Knowledge System	Generic IKS Course: Basic Knowledge Of The IKS To Be Offered At First Year Level	Major-Specific IKS Courses: Advanced Information About the Major, Part of the Major Credit to Be Offered at Second- Or Third- Year Level
OJT	On-Job Training (Internship / Apprenticeship)	Corresponding To the Major Subject	Related To The Major
FP	Field Projects	Corresponding To the Major Subject	Related To the Major
CC	Co-Curricular Courses	Health And Wellness, Yoga Education Sports, And Fitness, Cultural Activities, NSS/NCC And Fine/ Applied/Visual/ Performing Arts	NA
CE	Community Engagement and Service		Related To Major
RP	Research Project	Corresponding To the Major Subject	Related To Major

Programme Template

Degree		**Bachelor of Computer Application (BCA)**
Programme		BCA
Preamble		The Bachelor of Computer Applications (BCA) program is a four-year undergraduate degree program as per NEP-2020 designed to provide students with a strong foundation in computer science and its applications. The program aims to equip students with the knowledge and skills required to excel in the rapidly evolving field of computer science and information technology. The BCA program combines theoretical knowledge with practical applications to ensure that students develop a comprehensive understanding of computer systems, software development, database management, networking, and other core areas of computer science. It is an ideal choice for students who are interested in pursuing a career in the IT industry or furthering their studies in computer science. During the course of the BCA program, students are exposed to a wide range of subjects that cover various aspects of computer science. These subjects typically include programming languages, data structures, algorithms, computer architecture, operating systems, software engineering, web development, database management systems, computer networks, and information security. Upon successful completion of the BCA program, graduates have a wide range of career opportunities in the IT industry. They can work as software developers, system analysts, database administrators, network administrators, web developers, IT consultants, and other related roles. Graduates may also choose to pursue higher education, such as a Master's degree in computer science or a specialized field within the IT domain. By combining theoretical knowledge, practical skills, and industry exposure, the program equips students with the necessary tools to thrive in the IT industry and contribute to technological advancements
Programme Specific Outcomes (PSOs)		After completing this programme, Learner will
	1	Describe a strong foundation in computer application, including knowledge of Programming languages, Database, Mathematics, Operating system and Networking.

	2	Analyze the ethical and professional responsibilities in the field of computer applications by evaluating the Implications of adhering to professional standards and practices.
	3	Applying programming knowledge to develop a software application to solve specific problems.
	4	Evaluate software designs and architectures for efficiency, security and user experience.
	5	Design a software application to meet the requirements of the Industrial Standards.
Eligibility Criteria for Programme		As per AICTE approval process
Intake **(For SNDT WU Departments and Conducted Colleges)**		

Structure with Course Titles

Bachelor of Computer Application (BCA)

Semester – V

Sr. No.	Course	Type of Course	Credits	Marks	Int Marks	Ext Marks
	Semester – V					
50135411	Software Engineering (Th)	Major (Core)	4	100	50	50
50135412	Computer Networks (Th+Pr) (2+2)	Major (Core)	4	100	50	50
51035411	Introduction to Ancient Indian Computing (Th)	IKS (Major Specific)	2	50	0	50
50235411	Artificial Intelligence (Th+Pr) (2+2)	Major (Elective) **(Any One)**	4	100	50	50
50235412	Cloud Computing (Th+Pr) (2+2)					
50235413	Internet of Things (Th+Pr) (2+2)					
50335411	Introduction to Data Science (Th+Pr) (2+2)	Minor Stream	4	100	50	50
50635401	Software Testing (Pr)	VSC-4	2	50	50	0
51335401	Field Project (Mini projects)(Pr)	FP	2	50	50	0
			22	**550**	**300**	**250**

Semester – VI

Sr. No.	Course	Type of Course	Credits	Marks	Int Marks	Ext Marks
	Semester - VI					
60135411	Full Stack Web Development (Th+Pr) (2+2)	Major (Core)	4	100	50	50
60135412	Cyber Security (Th)	Major (Core)	4	100	50	50
60235411	Machine Learning (Th+Pr) (2+2)	Major (Elective) **(Any One)**	4	100	50	50
60235412	Blockchain Technology (Th+Pr) (2+2)					
60235413	Natural Language Processing (Th+Pr) (2+2)					
60335411	IT Project Management (Th)	Minor Stream	2	50	0	50
60335412	Big Data Analytics (Th+Pr) (2+2)	Minor Stream	4	100	50	50
61235421	Field Project (Major Project / Internship) (Pr)	OJT	4	100	50	50
			22	**550**	**250**	**300**

Exit with Degree (3-year)

Note:-
this UG Credit Structure format is followed from Faculty of Science & Technology. Following is the link

https://sndt.ac.in/pdf/academics/syllabus-as-per-nep/faculty-of-science-and-technology/ug-credit-structure-science-and-technology.pdf

Semester – V

.5.1 Major (Core)

Course Title	**Software Engineering (Th)**
Course Credits	4 Credits
Course Outcomes	**After going through the course, learners will be able to**
	Explain fundamental concepts, principles, and models of software engineering.
	Apply software development life cycle (SDLC) models
	Analyse software requirements and design appropriate system models.
	Evaluate different software design and testing techniques for quality assurance.
	Develop small software systems using standard engineering practices and tools.
Module 1(Credit 1) - Introduction to Software Engineering & Process Models	
Learning Outcomes	Understand software engineering principles and challenges
	Compare different SDLC models
	Identify appropriate models for given scenarios

Content Outline	• Introduction to Software Engineering
	• Software Crisis & Importance
	• Software Development Life Cycle (SDLC)
	• Process Models:
	  ◦ Waterfall Model
	  ◦ Incremental Model
	  ◦ Spiral Model
	  ◦ Agile Models (Scrum, XP basics)
	• Comparison of Process Models

Module 2(Credit 1) - Software Requirements Engineering	
Learning Outcomes	Identify and classify software requirements
	Analyse and document requirements
	Create requirement models

Content Outline	• Requirements Engineering Process • Types of Requirements (Functional & Non-functional) • Requirement Elicitation Techniques • Software Requirement Specification (SRS) • Use Case Modelling • Introduction to UML Diagrams (Use Case, Activity)
Module 3 (Credit 1) - Software Requirements Engineering	
Learning Outcomes	Apply design principles and concepts
	Develop system design using UML
	Evaluate design quality

Content Outline	• Design Concepts (Abstraction, Modularity, Cohesion, Coupling)
	• Architectural Design
	• Design Patterns (Introduction)
	• UML Diagrams:
	  • Class Diagram
	  • Sequence Diagram
	• User Interface Design Principles

Module 4 (Credit 1) - Software Testing & Maintenance	
Learning Outcomes	Apply testing techniques
	Analyse defects and ensure software quality
	Understand maintenance strategies
Content Outline	• Software Testing Fundamentals • Testing Levels: Unit, Integration, System, Acceptance • Testing Techniques: Black-box & White-box • Debugging & Quality Assurance • Software Maintenance Types
Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)	

Module 1:

1. Propose a software project (e.g., Library System, E-learning App).
2. Identify the type of requirements (stable/changing).
3. Select a suitable SDLC model.
4. Justify your choice with at least three reasons.
5. Draw a basic process diagram of your selected model.
6. Explain how your model handles:
   - Risk
   - Customer feedback
   - Time management

Module 2:

Prepare a SRS document for any one system:

- E-learning platform
- Hospital management system
- Food delivery app

Include: Introduction, Overall Description, Functional Requirements, Non-functional Requirements, Use Case Summary

Module 3:

Design a system for any one:

- E-learning platform
- Hospital management system
- Food delivery

app Questions:

1. Identify modules (modularity).
2. Apply abstraction in design.
3. Ensure high cohesion and low coupling (explain how).
4. Draw:
   - Class Diagram
   - Sequence Diagram
5. Suggest one design pattern and justify its use.
6. Design a basic UI layout.

Module 4:

Perform testing for a simple system (e.g., ATM / Login / Shopping Cart)
Questions to address:

1. Identify testing levels applicable.
2. Design at least 5 test cases (Black-box).
3. Identify logic paths (White-box approach).
4. Simulate possible errors and debugging steps.
5. Suggest SQA practices for the system.
6. Identify possible maintenance requirements.

Text Books:

- Jalote, P. (2024). *Software engineering: With open source and GenAI*. Wiley India.
- Dutt, S., & Subramanian, C. G. C. (2015). *Software engineering*. Pearson India.

Reference Books:

- Mishra, J., & Mohanty, A. (2011). *Software engineering*. Pearson India.
- Mathew, S. *Software engineering*. S. Chand.
- Pressman, R. S. (2010). *Software engineering: A practitioner's approach*.

Assessment:

Internal Assessment:
(Marks 50) Evaluation
Scheme:
Internal written exam for 25 marks
Internal examination on practicals/project (rubrics as follows)

Component	**Activity**	**Marks**
1	Assignment (Concept + Analytical Questions)	5
2	Case Study Analysis	5
3	UML & Design Activity	5
4	Testing & SRS Assignment	5
5	Presentation/viva	5

Given below are sample project but it is expected to work on similar sort of projects

Project: Choose a software system Apply:

- SDLC model
- Requirements
- Design (UML)
- Testing strategy

External Assessment: (50 Marks)
End Semester examination of 50 marks for 2 hours duration will be conducted

Semester – V

.5.2 Major (Core)

Course Title	**Computer Networks** [2 Theory + 2 Practicals]
Course Credits	**4 Credits**
Course Outcomes	**After going through the course, learners will be able to**
	Explain and **Understand** fundamental concepts, phases and models of Computer Network
	Apply networking concepts to configure basic network setups, including IP addressing, sub netting, etc.
	Analyse and different network protocols and **Evaluate** performance issues to identify bottlenecks and security vulnerabilities
	Develop and implement network-based solutions
Module 1 (Credit 1) - Introduction to Networking	
Learning Outcomes	Demonstrate understanding and workings of Layers of OSI model
	Explain components of computer network hardware and software
Content Outline	**Introduction to Networking:** Introduction to computer network, network application, (Interconnection networking devices- NIC, Hub, switch, Cable, Router, Modem), Network topology (Bus, Star, Ring), connection oriented and connectionless services. **Reference models:** Layer details of OSI, TCP/IP models.
Module 2 (Credit 1) – Physical Layer	
Learning Outcomes	Distinguish between analog and digital signals and understand their characteristics
	Understand the basic concepts of data communications.
	Identify appropriate transmission media for networking

Content Outline	**Physical Layer:** Data and signals, periodic analog signals, digital signals, transmission impairment, data rate limits, performance**.** **Digital and Analog transmission:** **Digital to Digital conversion:** ● Unipolar ● Polar - NRZ-L, NRZ-I, RZ, Biphase – Manchester, Differential Manchester ● Bipolar **Analog to Digital conversion:** ● Pulse Amplitude Modulation – PAM ● Pulse Code Modulation – PCM **Digital to Analog conversion:** ● Amplitude Shift keying (ASK) ● Frequency Shift keying (FSK) ● Phase shift keying (PSK) **Analog to Analog conversion:** ● Amplitude Modulation ● Frequency Modulation ● Phase Modulation **Transmission modes:** simplex, Half Duplex, Full Duplex. **Multiplexing, Transmission Media and Switching:** **Multiplexing** – FDM, WDM and TDM **Transmission Media** – *Guided Media* (Twisted Pair, Coaxial and Fiber Optics) and *Unguided Media* i.e. Wireless Media (Radio waves, Microwave, Bluetooth, Infrared) **Switching** – Circuit and Packet Switching.
Module 3 (Credit 1) - Data Link Layer	
Learning Outcomes	Identify issues of network design
	Differentiate Error Detection and Correction Techniques
	Analyze different Retransmission policies
Content Outline	**Data link Layer:** **DLL Design Issues-services (Framing, Error Control, Flow Control):** **Framing**-Types of Framing, **Error control**-what is error, types of error, **Error Detection Techniques** (CRC, Checksum), **Error Correction Techniques-Retransmission** (Stop and Wait ARQ, Go Back N ARQ, Selective Repeat ARQ). **Medium Access Protocols:** Channel Allocation problem, Multiple access Protocol (Aloha, Carrier Sense Multiple Access (CSMA/CD).
Module 4 (Credit 1) - Network Layer, Transport Layer & Application Layer	
Learning Outcomes	Understand Network Layer design issues
	Categorize IPv4 Addressing
	Select best path by analyzing various Routing algorithms

	Explain prevention and management of network congestion
	Examine end-to-end communication between devices.
	Discover protocols used for interaction between applications and end users
Content Outline	**Network Layer :** Network Layer design issues, Communication Primitives: Unicast, Multicast, Broadcast. IPv4 Addressing (class full and classless) Commands : ipconfig,ipconfig/all, ping, tracert, route print(for displaying routing table) **Routing algorithms :** Shortest Path (Dijkastra's), Link State Routing, **Congestion control algorithms:** Open loop congestion control, Closed loop congestion control **Transport Layer :** Introduction, Transport layer Services. The User Datagram Protocol (UDP). Commands : netstat(for showing active connections, different variations netstat -a, netstat -n, netstat -an(tcp buffer queue), netstat -p **Application Layer :** Introduction, Services, Client Server Model Protocols : DNS, HTTP, SMTP, FTP, Telnet Commands : Various DNS commands- nslookup, ftp, tftp, telnet, arp -a
Assignments towards Comprehensive Continuous Evaluation:	
Module 1: Q.1) Write a short note on Computer network components A) NIC B) Hub C)switch D) Cable E)Router F)Modem Q.2) Explain architecture of OSI model in detail. Q.3) Explain Block Diagram of TCP /IP Model in detail. Q.4) Write down comparison between OSI model and TCP /IP Model Q.5) Explain types of Network topology (Bus, Star, Ring). Q.6) Explain connection oriented and connectionless services in networking Q.7) Differentiate between connection oriented and connectionless services	

Module 2:

Q.1) Write a short note on Analog Signal
Q.2) Write a short note on Digital Signal
Q.3) Write down difference between Analog Signal and Digital Signal
Q.4) Write a short note on Transmission impairment
Q.5) Explain following types of conversion
A)Digital to Digital conversion B) Analog to Digital conversion C) Digital to Analog conversion D) Analog to Analog conversion
Q.6) Write a short notes on Transmission modes
Q.7) Explain Guided Transmission Media in detail
Q.8) Explain Unguided Transmission Media in detail
Q.9) Write a short notes on 1) FDM 2)WDM 3)TDM
Q.10) What is Multiplexing? Explain techniques of Multiplexing.
Q.11) What is Circuit switching?
Q.12) What is Packet Switching?
Q.13) Write down comparison between Circuit switching and Packet Switching.

Module 3:

Q.1) What is framing? Explain types of framing.
Q.2) What is error? Explain types of error.
Q.3) Explain Error detection techniques in detail. Or Write a short note on- 1) CRC 2) Checksum
Q.4) Explain Error correction techniques in detail Or Write a short note on- 1) Stop and Wait ARQ 2) Go Back N ARQ 3) Selective Repeat ARQ
Q.5) Explain Multiple access Protocol - Aloha.
Q.6) Explain Multiple access Protocol - Carrier Sense Multiple Access (CSMA/CD).

Module 4:

Q.1) Write down functions of Network layer.
Q.2) What is casting? Explain types of casting.
Q.3) Describe Network layer design issues.
Q.4) What is IP address? Explain types of IP addresses.
Q.5) Explain IPv4 addressing system.
Q.6) Explain classfull and classless addressing
Q.7) Explain Shortest Path Routing algorithm.
Q.8) Explain Link State Routing algorithm.
Q.9) Explain Open loop congestion control and Closed loop congestion control
Q.10) Write a short note on UDP
Q.11) Write down various functions of Application layer protocol.

Text Books:

1. A.S. Tannenbaum ,"Computer Networks", 4th edition Prentice hall of India

References Books:

1. Douglas E Comer. *Internetworking with TCP/IP: Principles and Architecture* (5th Edition). Pearson Education
2. W. Richard Steven. *TCP/IP illustrated*. (volume 1)
3. Douglas E. Comer. *Computer Networks and Internets* (4th Edition). Prentice Hall.

Assessment:
Evaluation Scheme:

Internal Assessment:
(Marks 50) Internal
written exam for 25 marks
Internal examination on practicals/project (rubrics as follows)

Criteria	Description	Marks
1. **Conceptual Understanding**	Understanding of key concepts such as OSI Model, TCP/IP, protocols, and networking principles	**5**
2. **Problem Solving / Numerical Ability**	Ability to solve problems related to subnetting, delay, throughput, and routing	**5**
3. **Practical / Lab Skills**	Ability to perform configurations, simulations, and packet analysis using tools like Wireshark or Packet Tracer	**5**

4. **Application & Analysis**	Applying networking concepts to real-world scenarios and analyzing network behavior	**5**
5. **Presentation & Communication**	Clarity in explanation, structured answers, diagrams, and proper terminology	**5**

Given below are sample projects but it is expected to work on similar sort of projects. Project 1: Network Topology Simulation

Use tools like Cisco Packet Tracer
Create star, mesh, bus, and ring topologies Compare performance and cost

https://learningnetwork.cisco.com/s/question/0D53i00000Kt599CAB/download-packet-tracer
https://www.netacad.com/skillsforall/files/Cisco_Packet_Tracer_Download_and_Installation_Instructions.pdf
https://www.netacad.com/learning-collections/cisco-packet-tracer?courseLang=en-US

Project 2: IP Addressing & Sub netting

Tool Develop a small program (C/Java/Python) Input: IP address
Output: Subnet mask, network ID, host range
E.g.: Write a program in java to print **whois** information of your computer

Project 3: Simple Chat Application (Client-Server)

Use socket programming in Java to enable message exchange

Project 4: Network Traffic Monitoring Tool
(Python) Project 5: Lan Based Quiz System using python
Project 6: Checksum Error Detection Simulator using Python

External Assessment: (Marks 50) End Semester examination of 50 marks for 2 hours duration will be conducted.

Semester – V

.5.3 Indian Knowledge System (IKS) (Major Specific)

Course Title	**Introduction to Ancient Indian Computing (Th)**
Course Credits	2 Credits
Course Outcomes	**After going through the course, learners will be able to**
	To understand the computational power of Vedic Sutras, Shakha traditions, and ancient mathematical texts.
	To know the importance of Indian science
	To learn how ancient Indian logic (Nyaya) relates to modern computer logic and AI.
	To see how Panini's grammar acted as the world's first formal programming language.
Module 1(Credit 1) - History, Binary Maths, Hashing & Syntax	
Learning Outcomes	Understand (L2): Explain the historical evolution of binary logic & decimal systems.
	Apply (L3): Use Katyapayadi formula to encode/ decode numerical data.
	Analyze (L4): Compare Paninian grammar structures to modern context-free grammars.
	Examine (L4): Deconstruct Pingala's algorithms for generating binary sequences in prosody.
Content Outline	Introduction: History of Vedic knowledge vedic systems and the shift toward structured scientific thought. Overview of Indian Science: The decimal place-value system, the concept of Zero (Shunya), and scientific categorization. Binary Numbers: Pingala's binary system for poetry and the Meru Prastara (Pascal's Triangle) for combinations. Katapayadi Formula: Using alphanumeric "hashing" to compress large numbers into mnemonic words and verses. Panini-Backus Form: Analyzing Ashtadhyayi as the world's first formal language and its link to modern programming syntax (BNF).
Module 2(Credit 1) - Automata, Logic, Planets & Consciousness	
Learning Outcome	Analyze (L4): The Mathematical frameworks of ancient Indian logic systems (Nyaya) and their parallels in modern boolean logic.

s	Evaluate (L5): The efficiency of Paninian grammar when tested against modern theoretical models like Fowler's Automation.
	Critique (L5): The scientific accuracy of cosmological constants in Puranic literature, including calculations related to the speed of light.
	Synthesis (L6): Ancient Indian theories of consciousness with contemporary cognitive science to propose new frameworks for AI.
Content Outline	Panini tested by Flower's Automation: Introduction of Calibration, Fluorescence Mechanism, Methodology, Improvement, Enhancement Formal structure in Indian Knowledge: Layered Structure, Chaturdaśa-Vidyāsthāna, Vaidik vs. Avaidik Planets in vedic literature: The speed of light and puranic cosmology Concepts Sayana's Calculation, Puranic Cosmology Basis, Cosmic Egg (Hiranyagarbha), Heliocentrism and Time

External Assessment – 50 Marks

End Semester examination of 50 marks for 2 hours duration will be conducted

References -

1. "Computing Science in Ancient India" – Edited by T.R.N. Rao & Subhash Kak.
2. "Mahabhasya on Panini's Sutras - A foundational commentary by Patañjali" (2nd Century BCE) on Pāṇini's work.
3. "Elements of Vedic Astrology (Vols 1 & 2)" - by Dr. K.S. Charak on the Parashari system.

Semester – V

.5.4 A. Major (Elective)

Course Title	**Artificial Intelligence (Major Elective-I)** **[2 theory + 2 practicals]**
Course Credits	**4 Credits**
Course Outcomes	**After going through the course, learners will be able to**
	Explain fundamental concepts of Artificial Intelligence.
	Apply AI techniques to solve problems.
	Analyze intelligent systems and their components.
	Evaluate different AI approaches and tools.
	Develop simple AI-based applications.
Module 1(Credit 1) - Introduction to Artificial Intelligence	
Learning Outcomes	Understand AI concepts and history
	Identify AI applications and intelligent agents

Content Outline	• Introduction to Artificial Intelligence
	• History and Evolution of AI
	• Applications of AI
	• Intelligent Agents
	  ◦ Agent structure
	  ◦ Agent and Environment
	  ◦ Types
	• Problem Solving in AI
	  ◦ Search strategies (Informed and Uninformed search)
	  ◦ Classical problems (Water Jug, 8-Queens Puzzle, Traveling salesman)

Module 2(Credit 1) - Knowledge Representation & Reasoning	
Learning Outcomes	Understand knowledge representation techniques
	Apply logic and reasoning methods
Content Outline	• Knowledge Representation • Propositional Logic • Predicate Logic • Inference Mechanisms • Semantic Networks
Module 3 (Credit 1) – Introduction to Fundamentals of Machine Learning	
Learning Outcomes	Understand ML concepts

	Apply basic ML algorithms

Content Outline	Introduction to Machine Learning
	● Types of learning
	  ○ Supervised
	  ○ Unsupervised
	  ○ Reinforcement
	● Classification-Binary and Multiclass
	● Regression-Linear and Non-Linear
	● Clustering
	● ML algorithms-SVM,K-Mean Clustering

Module 4 (Credit 1) - AI Applications & Ethics	

Learning Outcomes	Analyse AI applications
	Understand ethical implications
Content Outline	Natural Language Processing • Computer Vision • Robotics • AI Ethics & Bias • Future Trends
Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)	
Module 1: Explain different AI applications with examples. Identify intelligent agents in real-world systems.	
Module 2: Solve simple problems using propositional logic. Represent knowledge using semantic networks.	
Module 3: Identify real-world problems suitable for ML. Compare supervised and unsupervised learning	
Module 4: Analyse ethical issues in AI systems. Suggest improvements to reduce bias in AI.	

Text Books:

1. Russell, S., & Norvig, P. Artificial Intelligence: A Modern Approach.
2. Elaine Rich. Artificial Intelligence.

Reference Books:

1. Nils J. Nilsson. Artificial Intelligence: A New Synthesis.
2. Stuart Russell. Artificial Intelligence Basics.

Assessment:

Internal Assessment: (50 Marks)

Evaluation Scheme:

Internal Assessment:

(Marks 50) Internal
written exam for 25 marks
Internal examination on practicals/project (rubrics as follows)

Component	Activity	Marks
1	Assignment (Concept + Analytical Questions)	5
2	Case Study Analysis	5
3	Practical/ML Task	5
4	Mini Project	5
5	Presentation	5
Total		**25**

Given below are sample project but it is expected to work on similar sort of projects Project Guidelines:

- Choose a suitable AI-based system (e.g., Chatbot, Recommendation System, Image Classifier, etc.)
- Apply relevant AI concepts such as problem solving, knowledge representation, or machine learning.
- Identify requirements and design the system approach.
- Implement or simulate the solution using appropriate tools/techniques.
- Evaluate system performance and discuss limitations.

External Assessment: (50 Marks)
End Semester examination of 50 marks for 2 hours duration will be conducted.

Semester – V

.5.4 B. Major (Elective)

Course Title	**Cloud Computing (Major Elective)** **[2 theory + 2 practicals]**
Course Credits	**4 Credits**
Course Outcomes	**After going through the course, learners will be able to**
	Understand the fundamentals and architecture of cloud computing.
	Analyse different cloud service and deployment models.
	Explain virtualization and cloud resource management techniques.
	Evaluate cloud platforms and implement basic cloud-based solutions.
	Identify security, privacy, and ethical issues in cloud environments.
Module 1(Credit 1) - Introduction to Cloud Computing	
Learning Outcomes	Explain the concept and evolution of cloud computing
	Identify key characteristics and benefits
	Compare traditional vs cloud computing
	Explain cloud infrastructure components
Content Outline	Introduction to Cloud Computing: ● Introduction to Cloud Computing: Definition, Characteristics, Evolution and benefits of Cloud Computing, Advantages and disadvantages, ● Cloud Architecture: Components of cloud ● Types of cloud based on deployment model- Public, Private, Hybrid, Community. Types of cloud based on service model- Infrastructure as a service. Platform as a service. Software as a service. ● Cloud Technologies-Virtualization, SOA, Grid Computing, Utility Computing ● Technologies and processes required for deploying web web services, Deploying a web service from inside and outside a cloud architecture,
Module 2(Credit 1) – Virtualization and Cloud IT Model	
Learning Outcomes	Understand virtualization concepts
	Analyse types of virtualizations

Content Outline	●Virtualization: concept, Need, Pros and cons of virtualization ●Hypervisor, types of hypervisors (Type-1 and Type-2), VMs and containers, advantages and disadvantages ●Taxonomy model, Server virtualization, Application virtualization, Desktop virtualization, Storage virtualization, Network virtualization ●concept of data Center ●Multi-tenancy and scalability ●Serverless Computing ●Analyse the case "How to decide if the cloud is right for your requirements" ●Analysis of case study on "How to choose cloud-based service, applications and development platform deployment so as to improve the total cost of ownership"
Module 3 (Credit 1) – Cloud Services and Providers	
Learning Outcomes	Explore major cloud service providers
	Use basic cloud services
	Compare cloud platforms
Content Outline	● Compute Services ● Storage Services ● Database Services ● Networking in Cloud ● Overview of major cloud providers: ● Amazon Web Services (AWS) - Introduction, Advantages, Disadvantages, core AWS Services -Compute, Storage, Database, networking, security and management ● Microsoft Azure- Introduction, Advantages, Disadvantages, core AWS Services -Compute, Storage, Database, networking, security and management ● Google Cloud Platform (GCP)- Introduction, Advantages, Disadvantages, core AWS Services -Compute, Storage, Database, networking, security and management ● Cloud -based Application Development technologies (DevOps,CI/CD, containerization, kubernetes)- Definition, use, tools needed.
Module 4 (Credit 1) - Cloud Security, Management and Applications	
Learning Outcomes	Understand cloud security challenges
	Apply basic security practices
	Evaluate real-world cloud applications

Content Outline	● Cloud Security -Definition & importance of cloud security, Security challenges in cloud computing, Shared Responsibility Model ● Data Security and Privacy - importance ● Identity and Access Management (IAM) -Authentication & authorization Roles, permissions, MFA, Principle of Least Privilege ● Risks and Threats in Cloud ● Backup and Disaster Recovery - importance, types ● Service Level Agreements (SLA) - Definition, importance ● Case Studies and Applications (E-commerce, Education, Healthcare) - How cloud is used? benefits of using the cloud..

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)

Module 1:

- Define cloud computing and explain its key characteristics. Why to use Cloud Computing?
- Compare traditional computing with cloud computing.
- Differentiate between: i) IaaS, PaaS, SaaS
  ii) Public, Private, Hybrid Cloud
- List real-world examples of cloud usage (e.g., Gmail, Dropbox).
- Explain Advantages and Disadvantages of Cloud Computing
- Explain Cloud Computing Architecture along with Components
- Explain Three types of Cloud Computing Services in brief
- Explain Cloud Computing Technologies (Virtualization, Service Oriented Architecture (SOA), Grid Computing, Utility Computing)

Module 2:

- Explain virtualization and its importance in cloud computing
- Differentiate between i) Type 1 and Type 2 hypervisors ii) VMs and containers
- Discuss about Taxonomy of Virtualization
- What is a data center?
- Explain the concept of serverless computing.

Module 3:

- Compare the following platforms: Amazon Web Services, Microsoft Azure and Google Cloud Platform with respect to Services offered, Pricing model, Ease of use, Popular use cases
- Explain the role of cloud providers in modern IT infrastructure.
- Explain the features and services offered by Amazon Web Services.
- Explain key advantages of Microsoft Azure.
- Discuss the major services provided by Google Cloud Platform.
- What is Kubernetes? Why Kubernetes is needed
- Discuss Cloud-based Application Development technologies with tools needed.

Module 4:

- What are major cloud security threats?
- Explain Identity and Access Management (IAM)
- Discuss data privacy challenges in cloud
- Explain Service Level Agreements (SLA)
- Identify risks in cloud computing

Text Books:

1. Rajkumar Buyya, Christian Vecchiola, & S. Thamarai Selvi. (2013). *Mastering cloud computing: Foundations and applications programming*. Burlington, MA: Morgan Kaufmann.
2. M. Scott Kingsley. (2024). *Cloud technologies and services: Theoretical concepts and practical applications*. Cham, Switzerland: Springer.

Reference Books:

1. Anthony T. Velte, & Toby J. Velte. (2010). *Cloud computing: A practical approach*. New York, NY: McGraw-Hill.
2. Rajkumar Buyya, Broberg, J., & Goscinski, A. (2011). *Cloud computing: Principles and paradigms*.
3. Ronald L. Krutz, & Russell Dean Vines. (2010). *Cloud security: A comprehensive guide to secure cloud computing*. Hoboken, NJ: Wiley.
4. Thomas Erl, Zaigham Mahmood, & Ricardo Puttini. (2013). *Cloud computing: Concepts, technology & architecture*. Upper Saddle River, NJ: Prentice Hall.

Assessment:
Internal Assessment: (50 Marks) Evaluation Scheme:
Internal Assessment: (Marks 50) Internal written exam for 25 marks
Internal examination on practicals/project (rubrics as follows)

Component	Activity	Marks
1	Assignment (Concept + Analytical Questions)	5
2	Conceptual knowledge	5
3	Understanding of tools/platforms	5
4	Correct implementation	5
5	Mini Project + Presentation /Documentation	5
Total		**25**

Given below are sample projects but it is expected to work on similar sort of projects

1. Project: Create a basic virtual machine using any free-tier cloud platform such as

- Amazon Web Services OR
- Microsoft Azure

2. Project: Deploy a static website Or Create cloud storage and upload files

External Assessment: (50 Marks)

End Semester examination of 50 marks for 2 hours duration will be conducted

Semester – V

.5.4 C. Major (Elective)

Course Title	**Internet of Things** **[2 theory + 2 practicals]**
Course Credits	**4 Credits**
Course Outcomes	**After going through the course, learners will be able to**
	● Understand evolution of the Internet into the Internet of Things and to define the building blocks of an IoT ecosystem. ● Understand Communication Standards required to handle the data traffic of billions of constrained devices. ● To empower students to design and implement smart solutions. ● **Interpret** the fundamental concepts and architecture of IoT
	● **Ability** to work with hardware like **Arduino, Raspberry Pi**
	● **Describe** the working of various protocols and its enabling Technologies and data analytics for IOT
Module 1(Credit 1) - Introduction of Internet of Things	
Learning Outcomes	Understand fundamental concepts of IOT
	Understand design and communication model of IOT
	Understand sensor network and embedded system
	Understand IOT issues and challenges
Content Outline	● Definition and Features of IoT ● Physical Design and Components of IOT ● Logical Design of IOT ● IOT functional blocks and communication model ● IOT Frameworks ● IOT communication API's ● Wireless Sensor Networks ● Introduction of Embedded Systems ● IoT Issues and Challenges- Planning, Costs and Quality ,Security and Privacy, Risks. ● IOT Testing ● Applications of IOT

Module 2(Credit 1) - Working of IOT Protocols and its enabling Technologies	
Learning Outcomes	Understand various classification of IOT protocols.
	Interpret the role of IOT protocol for efficient network communication.
	Explain communication Technologies
Content Outline	Protocols to be covered: ● IEEE 802.15.4 ● LoRaWAN, MQTT, CoAP, XMPP, AMQT, AMQP Enabling technologies to be covered: ● Bluetooth ● Zigbee ● NFC, RFID, WiFi ● IoT and cloud integration
Module 3(Credit 1) - Data and its Analytics for IoT	
Learning Outcomes	Understand data analytics for IOT
Content outline	● **Introduction to Data Analytics for IoT** :-Types of IOT analytics, Steps in IOT data analytics, Benefits, Real-world applications of IOT data analytics ● Structured and Unstructured Data ● Challenges in data analytics in IOT ● Data Visualization Techniques
Module 4(Credit 1) - Introduction to Arduino Programming, Raspberry Pi, SDN	
Learning Outcomes	Understand basics of Arduino programming
	Understand efficient use of Raspberry Pi
	Understand virtual IOT circuit simulator/virtual device/virtual IOT tool practically
	Understand Software Defined Networking Technology
Content Outline	● Basic Arduino Programming ● Extended Arduino Libraries ● Working of Sensors and Actuators with Arduino ● Implementation of IoT with Raspberry Pi ● Study the working of any one virtual IOT simulator ● SDN for IoT, Fog Computing, Smart Homes, Connected Vehicles, Industrial IOT ● Case Study: Agriculture, Healthcare, Smart Homes
Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)	

Module1:

Conceptual Assignment: Write short notes (150–200 words each):

- Definition & Features of IoT
- Physical vs Logical Design of IoT
- IoT Functional Blocks

Activity1: Draw & explain IOT functional blocks and communication model.
Activity2: Design a simple WSN for Forest fire detection or smart agriculture.
Activity3: Conduct Group Discussion on the topics: Security risks in IoT, Privacy concerns of IOT, Cost & scalability issues in IOT Systems
Activity4: PPT Presentations on topics: Latest IOT Trends, Smart IOT Applications, etc.

Module2:

Activity1: Create a comparison table for: MQTT vs CoAP vs AMQP (choose your own parameter for comparison)
Activity2: Explain following enabling technologies with ***working Principle, Range, Applications.***
Bluetooth, Zigbee, NFC & RFID
Case Study Assignment: Choose any one real-world application: Smart Agriculture, Smart Home, Smart City
Explain: Which protocol is used (MQTT/CoAP/LoRaWAN), Why it is suitable, Advantages over others.

Module3:

Assignment: Define both data types (structured & unstructured). Give at least 5 examples of each & explain their role in IOT.
Activity1: Draw and explain Steps in IOT Data Analytics.
Activity2: Topic for Group Discussion: Importance of Data Analytics in IoT.
Case Study Assignment: Using any one Domain (Smart Healthcare, Smart Agriculture, Smart City) explain Type of data collected, type of analytics used and benefits achieved for that particular domain.

Module4:

Activty1: Write and explain simple program for LED blinking.
Activity2: Write code using any one extended library. Explain its function. **Activity3:** Explain working of Sensors and Actuators (any1) with Arduino. **Case Study:**
Domains: Agriculture / Healthcare / Smart Homes
Create a Project Report of any one domain based on following task to accomplish: Problem statement, IoT system design, Technologies used (Arduino/RPi/Sensors) for that domain
Benefits & challenges of using IOT for that domain.
Case Study Simulation: Study the working of any one virtual IOT simulator

Text Books:

1. Arshdeep Bahga, Vijay Madisetti, "Internet of Things – A hands-on approach", Universities Press, 2020.
2. Anand Tamboli ,"Build Your Own IoT Platform", 1st Edition, Apress, 2019
3. David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Rob Barton, Jerome Henry, "IoT Fundamentals – Networking Technologies, Protocols, and Use Cases for the Internet of Things", 1st Edition,

Published by Pearson Education, Inc, publishing as Cisco Press, 2017.
4. Getting Started with Raspberry Pi, Matt Richardson & Shawn Wallace, O'Reilly (SPD), 2014, ISBN: 9789350239759

Reference Books:

1. Cuno Pfister, Getting Started with the Internet of Things, O"Reilly Media, 2011, ISBN: 978-1-4493-9357-1.
2. Perry Lea, "Internet of things For Architects", 1st Edition, Packt Publication, 2018
3. Francis daCosta, "Rethinking the Internet of Things: A Scalable Approach to Connecting Everything", 1st Edition, Apress Publications, 2013.
4. Hakima Chaouchi, "The Internet of Things - Connecting Objects to the Web", 1st Edition, Wiley, 2010

Assessment:

Internal Assessment: (50 Marks) Evaluation Scheme:
Internal Assessment:
(Marks 50) Internal written exam for 25 marks
Internal examination on practicals/project (rubrics as follows)

Component	Activity	Marks
1	Assignment	5
2	One project with IOT kit/Study of at least 2 Research papers on IOT	5
3	Showcase the usage of virtual tool/circuit simulator/device for IOT application	5
4	Case study Presentation	10
Total		**25**

External Assessment: (50 Marks)

End Semester examination of 50 marks for 2 hours duration will be conducted

Semester – V

.5.5 Minor Stream

Course Title	**Introduction to Data Science** **[2 Theory + 2 Practicals]**
Course Credits	**4 Credits**
Course Outcomes	**After going through the course, learners will be able to**
	Explain What is Data science, Life cycle of Data science, understanding of NumPy, Pandas, Matplotlib, Seaborn and EDA
	Apply the use of NumPy, Pandas, Matplotlib, Seaborn and EDA
	Analyse the different data sets by performing Data visualization and preprocessing the datasets.
	Evaluate the different datasets after performing the EDA and evaluate the model
	Develop a small analysis report using data analytical skills using Anaconda notebook tool.
Module 1(Credit 1) - I Introduction to Data Science	
Learning Outcomes	Understand Data Science, its evolution, and its components
	Understand the life cycle of Data Science, Compare with different fields
	Identify roles and responsibilities in the field of Data Science and Real-world applications
Content Outline	●What is Data Science? Definition and evolution of Data Science ●Relationships with Data Analytics, Machine Learning, Artificial Intelligence, Components of Data Science: Data collection, Data processing, Modelling, Interpretation ●Data Science Lifecycle -Overview of lifecycle stages: Problem definition, Data collection, Data cleaning & preprocessing, Exploratory Data Analysis (EDA), Model building, Model evaluation, Deployment & monitoring Iterative nature of lifecycle ●Roles in Data Science -Data Scientist, Data Analyst, Data Engineer, Machine Learning Engineer, Business Analyst ●Real world Applications of Data Science
Module 2(Credit 1) - NumPy for Data Science	
Learning Outcomes	Understand NumPy, Array creation, built-in functions
	Understand built-in functions, create and manipulate arrays
	Understand mathematical and statistical operations

Content Outline	● **Introduction to NumPy -** What is NumPy and why it is used, Advantages over Python lists, Installation and setup Importing NumPy (import NumPy as np) ● **NumPy Arrays Creation -** Creating arrays from Python lists ● **Built-in array creation functions:** np.array(), np. zeros(), np.ones(), np.arange(), np.linspace() ● **Mathematical Operations -** Element-wise operations: Addition, subtraction, multiplication, division Universal functions (ufuncs): np.sqrt(), np.exp(), np.log() ● Aggregation functions: np.sum (), np.mean(), np.std(), np.min(), np.max()
Module 3 (Credit 1) - Pandas for Data Handling	

Learning Outcomes	Understand Pandas data structure and concepts
	Learn to create a data frame, load, read the datasets
	Learn to apply data cleaning and data preprocessing techniques
Content Outline	●**Introduction to Pandas: What** is Pandas and its role in Data Science, Advantages over traditional data handling methods, Installation and setup, Importing Pandas (import pandas as pd) Core data structures: Series, DataFrame ●**Data Loading and Exporting :** Reading data: read_csv(), read_excel() Writing data: to_csv(), to_excel() Handling file paths and basic parameters: Delimiters, Headers, Index handling ●**Pandas DataFrame :** Structure of DataFrame (rows, columns, index) Creating DataFrames: From dictionaries,From lists of lists, From CSV/Excel files,Viewing and inspecting data: head(), tail(),info(), describe() Selecting data:Column selection ●**Data Cleaning and Preprocessing :**Identifying missing data: isnull(), notnull(), Handling missing values: dropna(),fillna() ,Detecting and removing duplicates: duplicated(), drop_duplicates(), Data type conversion (astype()), Renaming columns,Handling inconsistent data
Module 4 (Credit 1) - Data Visualization & Exploratory Data Analysis (EDA)	
Learning Outcomes	Understand visualization and its techniques and different types
	Understand different libraries in python
	Understand Exploratory Data Analysis before model building

Content Outline	● **Introduction to Data Visualization:** Definition and importance of data visualization, Role of visualization in Data Science lifecycle, Benefits: Simplifies complex data, Reveals trends and insights, Supports decision-making. ● **Types of visualizations:** Comparison (Bar chart, Line Chart), Distribution (Histogram), Composition (Pie chart), Relationship (Pair plot,scatter plot and correlation chart) ● **Visualization Libraries in Python:** Introduction to Matplotlib, Seaborn ● **Matplotlib:** Basic plotting: plot(), bar(), hist(), scatter() **Customization:** Titles, labels, legends, Axis limits and styles ● **Seaborn:** Advantages over Matplotlib, Built-in themes and styling,**Common plots:** sns.countplot(), sns.boxplot(), sns.heatmap(), sns.pairplot() ● **Exploratory Data Analysis (EDA):** Definition and purpose of EDA, Univariate, Bivariate and Multivariate analysis ● Steps in EDA: Understanding dataset structure, Summary statistics, detecting missing values, identifying outliers, Checking distributions ● Tools used: Pandas (describe (), info(), Visualization libraries ● Understand Train and Test split before model building
Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)	

Module 1:

1. Write a Python program to demonstrate basic data collection (manual input / CSV reading).
2. Program to simulate steps of Data Science lifecycle.
3. Case study: Identify a problem and outline lifecycle steps.
4. Write short notes / code snippets explaining roles (no heavy coding).
5. Readers can work with many similar programs

Module 2:

1. Create NumPy arrays using -np.array(), np.zeros(), np.ones().
2. Generate sequences using- np.arange() and np.linspace().
3. Perform element-wise operations (add, subtract, multiply, divide).
4. Use universal functions (np.sqrt(), np.log(), np.exp()).
5. Compute aggregation functions (sum, mean, std, min, max).
6. Program to reshape and index arrays.

Module 3:

1. Create Series and DataFrame from lists and dictionaries.
2. Load dataset using read_csv() and display first/last rows.
3. Perform column selection and filtering.
4. Handle missing values using dropna() and fillna().
5. Detect and remove duplicates.
6. Convert data types using astype().
7. Export cleaned data using to_csv() or to_excel()

Module 4:

Matplotlib Programs

1. Plot a simple line graph using plot().
2. Create a bar chart for categorical data.
3. Draw a histogram for frequency distribution.
4. Create a scatter plot to show relationships.
5. Customize plots (title, labels, legend, grid).

Seaborn Programs

6. Create a count plot using sns.countplot().
7. Draw a boxplot to detect outliers.
8. Generate a heatmap for correlation matrix.
9. Create a pairplot for multiple variable relationships.

EDA Programs

10. Perform EDA on a dataset:
    - Display summary describe()
    - Check missing values
    - Plot distributions
11. Identify and visualize outliers.
12. Analyse (univariate, bivariate and multivariate) relationships using scatter plots / pairplots/correlation chart.

Text Books:

1. Python for Data Analysis" by Wes McKinney, 3rd edition ,publisher O'Relly Media

Reference Books:

2. Python Data science hand book, **Author:** Jake VanderPlas, **Publisher:** O'Reilly Media **Latest Edition:** 1st Edition,

Publication Year: 2016

3. Python Data analytics, by Fabio Nelli,**Publisher:** Apress,**Latest Edition:** 3rd Edition **Publication Year:** 2023

Assessment:
Internal Assessment: (50 Marks) Evaluation Scheme:
Internal written exam for 25 marks
Internal examination on practicals/project (rubrics as follows)

Component	Activity	Marks
1	Continuous Assessment (Concept + Analytical Questions)	5
2	Practical Lab/work	5
3	Record book/Assignment	5
4	Mini Project + Case study Presentation	10
Total		**25**

Given below are sample project but it is expected to work on similar sort of projects

- Sales data analysis, student performance analysis, whether data analysis etc

External Assessment: (50 Marks)

End Semester examination of 50 marks for 2 hours duration will be conducted

Semester – V

.5.6 Vocational Skill Courses (VSC-4)

Course Title	**Software Testing** [2 Practicals]
Course Credits	**2 Credits**
Course Outcomes	After going through the course, learners will be able to
	Define and explain fundamental software testing concepts, principles, life cycle models, and types of testing.
	Apply black box and white box testing techniques to design effective test cases.
	Analyze and evaluate testing levels, techniques, and quality practices to select appropriate testing strategies.
	Design and develop test plans, test cases, and defect reports for simple software applications.
Module 1(Credit 1) - Foundations & Testing Techniques	
Learning Outcomes	**Define and explain** software testing concepts, terms, principles, and testing life cycle models.
	Differentiate and analyze testing approaches including verification vs validation, static vs dynamic testing, and black box vs white box testing.
	Apply black box testing techniques such as equivalence partitioning, boundary value analysis, and decision tables.
	Analyze and evaluate program logic using white box testing techniques like statement and path coverage.

Content Outline	• Definition and Role of Software Testing
	• Error, Fault, Defect, Bug, Failure
	• Goals and Principles of Testing
	• Software Testing Life Cycle (STLC)
	• Verification vs Validation (V-Model)
	• Static vs Dynamic Testing
	
	Testing Techniques
	
	• Black Box Testing
	  ○ Concept and Applications
	  ○ Equivalence Class Partitioning
	  ○ Boundary Value Analysis
	  ○ Decision Table Testing
	• White Box Testing
	  ○ Concept and Need
	  ○ Statement Coverage
	  ○ Path Coverage

Module 2(Credit 1) - Testing Levels, Management & Quality Assurance	

Learning Outcomes	**Explain and analyze** different levels of testing including unit, integration, system, and acceptance testing.
	Apply test planning, test case design, and execution techniques for simple applications.
	Analyze and evaluate defect reports, defect life cycle, and testing outcomes for decision-making.
	Explain and apply basic quality assurance practices to improve software quality and prevent defects.

Content Outline	**Levels of Testing** ● Unit Testing ● Integration Testing (Overview) ● System Testing (Performance, Security basics) ● Acceptance Testing (Alpha, Beta) **Test Management** ● Test Planning (basic concepts) ● Test Case Design ● Test Execution and Reporting ● Defect Reporting and Defect Life Cycle **Quality Assurance in Software Testing** ● Concept of Quality Assurance (QA) ● QA vs Quality Control (QC) ● Role of QA in Software Testing ● QA Activities: Reviews, Audits, Process adherence ● Introduction to Standards: ISO (overview), CMMI (basic levels) ● Importance of QA in defect prevention

Internal Assessment – 50
Marks Internal written
exam for 25 marks
Internal examination on practicals/project for 25 marks (rubrics as follows)

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)

Sr. No	**Evaluation**	**Description**	**Marks**
1	Test Case Design Assignment	Students design test cases using black box techniques (equivalence partitioning, boundary value analysis) for given problem statements.	5
2	Practical Activity (Bug Identification)	Students analyze a sample application/interface and identify defects with proper justification.	5
3	Class Test / Quiz	Written or MCQ-based test covering concepts like STLC, testing types, and techniques and basic QA concepts.	5
4	Mini Project / Case Study	Students prepare a test plan, design and execute test cases, generate defect reports, and include **basic QA practices** such as review checklist, process adherence, or	10

		simple audit observations.	

Given below are sample project but it is expected to work on similar sort of projects

Project 1. College Event Management System Project 2. Online Student Registration System

References – Text Books:

1. Myers, G. J., Sandler, C., & Badgett, T. (2011). *The Art of Software Testing* (3rd ed.). John Wiley & Sons.
2. Pressman, R. S., & Maxim, B. R. (2019). *Software Engineering: A Practitioner's Approach* (9th ed.). McGraw-Hill Education.

Reference Books:

1. Jorgensen, P. C. (2013). *Software Testing: A Craftsman's Approach* (4th ed.). CRC Press.
2. Beizer, B. (1995). *Black-Box Testing: Techniques for Functional Testing of Software and Systems*. John Wiley & Sons.

Semester – V

.5.7 Field Project

Course Title	**Field Project [practicals]**
Course Credits	**2 Credits**
Course Outcomes	**After going through the course, learners will be able to**
	CO1: Identify problems in IT or business environments.
	CO2: Apply computing knowledge to design practical solutions.
	CO3: Collect, analyse, and interpret field data effectively.
	CO4: Demonstrate teamwork, communication, and project management skills.
	CO5: Prepare professional project reports and presentations.
Module 1(Credit 1) - Project Planning & Problem Identification Field Work & Data Collection	
Content Outline	**Project Planning & Problem Identification (Week 1–2)** • Selection of project domain (IT, business, social sector) • Identifying problems in IT (operational, functional, economical, Technical, etc.,) • Defining objectives and scope • Preparing project proposal **Field Work & Data Collection (Week 3–6)** • Industry visit / community interaction • Data collection methods: • Surveys • Interviews • Observation • Use of digital tools (Google Forms, Excel, etc.)
Module 2(Credit 1) - Data Analysis & Implementation Report Writing & Presentation	

Content Outline	**Data Analysis & Implementation (Week 7–9)** ● Data organization and analysis ● Use of basic tools: ● MS Excel / Python basics ● Problem-solving approach ● Prototype or solution development (if applicable) **Report Writing & Presentation (Week 10–12)** Report structure: ● Introduction ● Methodology ● Findings ● Conclusion ● Documentation standards ● PPT preparation ● Final presentation & viva

Assessment:
Internal Assessment (50 Marks) Evaluation Scheme:
Internal examination on practicals/field project (rubrics as follows)

Assignments/activities will be developed based on the above modules given as examples Project 1: Website Analysis of Local Businesses
Evaluate the design, usability, and performance of websites used by local businesses. Identify strengths, weaknesses, and suggest improvements for better user experience.

Project 2: Cybersecurity Awareness Survey
Conduct surveys to evaluate awareness of cybersecurity practices among users or businesses. Analyse findings and suggest measures to improve digital safety and awareness.

Component	Marks
Project Proposal	5
Field Work & Data Collection	10
Data Analysis / Implementation	10
Report Submission	10
Presentation & Completion Certificate from the employer	15
Total	**50**

Guidelines for Students

- Students should work in groups of 2–4
- Students should maintain a project diary/logbook
- They should ensure originality (no plagiarism)
- There should be Minimum 1 field visit which is mandatory
- Submission of both hardcopy and softcopy report is mandatory as per the prescribed format given in the Module.
- Feedback from the employer is mandatory upon completion of the project
- Completion Certificate from the College will be given upon completion of the Field Project with Credits and Hours mentioned.

BCA

Semester VI

Course Syllabus

Semester – VI

.6.1 Major (Core)

Course Title	**Full Stack Web Development** **[2 Theory + 2 Practicals]**
Course Credits	**4 Credits**
Course Outcomes	**After going through the course, learners will be able to**
	CO1: Design and build responsive, accessible user interfaces using modern HTML, CSS, and JavaScript.
	CO2: Implement server-side logic and RESTful APIs using Node.js and Express.
	CO3: Architect and manage relational or non-relational databases to persist application data.
	CO4: Integrate front-end and back-end systems into a cohesive, deployed web application.
Module 1(Credit 1) - The Modern Front-End (The Presentation Layer)	
Learning Outcomes	Understand: Explain the Document Object Model (DOM) and how browsers render content.
	Apply: Create a responsive landing page that adapts to mobile, tablet, and desktop views.
	Analyse: Debug layout issues and optimize CSS performance for faster load times.
Content Outline	● **Advanced HTML5/CSS3**: Moving beyond soup. Using header, nav, section, and article tags for SEO and screen readers. ● **Flexbox/Grid**: Deep dive into Flexbox and CSS Grid. Understanding the Box Model, positioning, and Z-index. ● **Responsive Design**: Media queries, "Mobile-First" strategy, and an introduction to utility-first CSS (like Tailwind CSS) or component libraries (like Bootstrap). ● **JavaScript (ES6+)**: Arrow functions, Template literals, Destructuring, and Modules.
Module 2(Credit 1) - Server-Side Foundations (The Logic Layer)	

Learning Outcomes	Apply: Build a functional web server capable of handling GET and POST requests.
	Create: Design a RESTful API structure to handle CRUD (Create, Read, Update, Delete) operations.
	Evaluate: Select appropriate middleware for authentication and error handling.
Content Outline	**Python Runtime & Async** ● The asyncio Event Loop ● Non-blocking I/O (async/await) ● Virtual Environments (venv) and pip **Web Frameworks (FastAPI/Flask)** ● Server setup and uvicorn ● Decorator-based Routing ● Request/Response handling ● Automatic Data Validation **Middleware & Security** ● Custom Middleware for Logging ● Built-in Data Parsing ● CORS Configuration **Routing & Standards** ● URL Naming Conventions ● HTTP Methods (GET, POST, DELETE) ● HTTP Status Codes (200, 404, 500) ● DELETE) ● HTTP Status Codes (2xx, 4xx, 5xx)
Module 3(Credit 1) - Data Persistence (The Data Layer)	
Learning Outcomes	Understand: Contrast the differences between relational and non-relational databases.
	Apply: Execute database queries to retrieve and manipulate stored information.
	Create: Design a database schema that maintains data integrity and supports application requirements.

Content Outline	● **Introduction to Databases:** Choosing between **SQL** (Structured/Relational) and **NoSQL** (Flexible/Document-based). ● **NoSQL with MongoDB:** Understanding Collections and Documents. CRUD operations using the MongoDB shell. ● **Object Data Modeling (ODM) with Mongoose:** Defining strict Schemas in a schemaless DB, validation rules, and "Models." ● **Data Relationships:** Implementing One-to-One, One-to-Many, and Many-to-Many relationships using Document Referencing (normalization) or Embedding. ● **Advanced Querying:** Sorting, Pagination, and Filtering data to improve application performance.
Module 4(Credit 1) - Integration, Security & Deployment	
Learning Outcomes	Analyse: Identify security vulnerabilities like XSS and CSRF in a web application.
	Apply: Implement secure user authentication using JSON Web Tokens (JWT).
	Create: Deploy a full-stack application to a live cloud environment with a CI/CD pipeline.

Content Outline	•**Client-Side Frameworks (Intro to React/Vue):** Understanding Components, Props, and basic State management to connect the front-end to your API. •**Authentication & Authorization:** Implementing User Sign-up/Login. Hashing passwords with **Bcrypt** and maintaining sessions with **JSON Web Tokens (JWT)**. •**Web Security Best Practices:** Sanitizing inputs to prevent SQL/NoSQL Injection and using **Helmet.js** for HTTP header security. •**Version Control & CI/CD:** Advanced Git (branching/merging) and setting up automated deployment pipelines. •**Cloud Deployment:** Deploying the backend to platforms like **Render** or **AWS**, and the frontend to **Vercel** or **Netlify**, including Environment Variable management (.env).

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)
Module 1: The Modern Front-End **Assignment 1**: Semantic Web & Layout Theory **Task:** Explain the difference between and <section>. Why is SEO important for a business website? **Comparison Table:** Create a table comparing Flexbox vs. CSS Grid (Use cases, strengths, and limitations). **Theory:** Describe the "Critical Rendering Path"—how a browser turns HTML/CSS into pixels. **Activity 1**: The "Component Recreator" **Task:** Find a complex UI component (like a Netflix movie card or an Instagram post) and recreate it using only HTML/CSS. **Activity 2:** Accessibility (A11y) Audit **Task:** Use a tool like Lighthouse or an extension to audit a popular website. Identify 3 accessibility errors and write the code to fix them.

Module 2: Server-Side Foundations

Assignment 1: The Anatomy of a Request (Python – FastAPI/Flask)

Task: Define the 5 main HTTP Methods (GET, POST, PUT, PATCH, DELETE) and provide a real-world scenario for each in the context of a Python web API.

Logic: Explain the difference between Synchronous and Asynchronous execution in Python using a "Restaurant Order" analogy (compare normal functions vs async/await with asyncio).

Activity 1: API Documentation (Postman)

Task: Build a small FastAPI or Flask server and use Postman to create a "Collection." Export the collection as documentation for a "Product Catalog" API.

Activity 2: Middleware Roleplay

Task: Draw a flowchart showing how a request passes through a Logger Middleware, an Authentication Middleware, and finally the Route Handler in a Python web framework (FastAPI or Flask).

Module 3: Data Persistence

Assignment 1: The Data Modeling Challenge

Task: Design the database schema for a "Simple Blog."

Comparison: Differentiate between Embedding vs. Referencing in NoSQL. When would you choose one over the other?

Diagram: Draw an Entity-Relationship (ER) diagram for a User, their Posts, and their Comments.

Activity 1: The "Query Master"

Task: Given a sample dataset of 100 "Students," write the Mongoose/SQL queries to find all students over age 21 with a GPA higher than 3.5.

Activity 2: CRUD Lab

Task: Create a simple script that connects to a database and performs a "Bulk Update"—for example, increasing the price of all "Electronics" category items by 10%.

Module 4: Integration, Security & Deployment

Assignment 1: Security & Auth Deep Dive

Task: Explain the concept of Hashing vs. Encryption. Why should we never store plain-text passwords?

Research: Write a 1-page report on the OWASP Top 10 web security risks (specifically XSS and SQL Injection).

Activity 1: JWT Flow Analysis

Task: Create a diagram or step-by-step list showing how a "JSON Web Token" is issued by the server and sent back by the client for every subsequent request.

Activity 2: The Deployment Checklist

Task: Create a "Production Readiness" checklist. What environment variables (.env) need

to change when moving from localhost to a live server like AWS or Render?

Assessment:

Evaluation Scheme:
Internal Assessment: (50 marks) Internal 25 marks
Written Test
Internal 25 marks Practical/Project (rubrics as follows)

Evaluation Parameters	**Description / Evaluation Points**	**Marks**
Conceptual Understanding	Responsive UI, DOM manipulation, and frontend logic	5
Accuracy and Completeness	API development, data handling, and correct functionality	5
Application and Analysis	CRUD operations, database integration, and search features	5
Presentation and Neatness	UI design, code organization, and basic security practices	5
Innovation and Effort	Originality, creativity, and complexity of implementation	5
Total		**25**

External Assessment – 50 Marks
End Semester examination of 50 marks for 2 hours duration will be conducted

References -

Textbooks (Foundational)

1. "Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics" by Jennifer Niederst Robbins (O'Reilly).
2. "Eloquent JavaScript: A Modern Introduction to Programming" by Marijn Haverbeke.
3. "Web Development with Node and Express" by Ethan Brown (O'Reilly).

Reference books:

1. "Clean Code: A Handbook of Agile Software Craftsmanship" by Robert C. Martin.
2. ”Full-Stack React Projects" by *Shama Hoque* (Packt Publishing).

Semester – VI

.6.2 Major (Core)

Course Title	**Cyber Security (Th)**
Course Credits	**4 Credits**
Course Outcomes	**After going through the course, learners will be able to**
	Explain fundamental concepts of cybersecurity, principles, different types of cyber-attacks, Cryptography and network security, application of cyber security, cyber laws, cyber forensics and real-world applications.
	Apply different Cryptography and Network security algorithms
	Analyse different systems and application security.
	Evaluate different cyber laws, digital forensics and different applications areas and its effects
	Develop and test different areas how security breaches happen
Module 1(Credit 1) - Fundamentals of Cyber Security	
Learning Outcomes	Understand fundamental concepts of cybersecurity
	Understand different malwares
	Understand cybersecurity goals, steps for better security and basics of security terminology
Content Outline	● Introduction to Cyber Security, need of Cyber security ● Importance & Challenges in Cyber Security ● Cyber Threats, Vulnerabilities & Types of Cyber Attacks ● Malware (Virus, Worm, Trojan, Ransomware) ● Phishing & Social Engineering ● Man-in-the-Middle (MITM) ● Denial of Service (DoS / DDoS) ● Types of Hackers (White hat, Black hat, Grey hat) ● Security Goals: Confidentiality, Integrity, Availability (CIA Triad) ● Cyber Security Policies,Steps for Better Security, Basics Security Terminology(Cryptography, Hacking, Encryption, Decryption)
Module 2(Credit 1) - Cryptography & Network Security	
Learning Outcomes	Understand the fundamentals of Cryptography
	Compare between Symmetric and Asymmetric key cryptography

	Different algorithms for substitution and transposition
Content Outline	Fundamentals of Cryptography Encryption & Decryption Concepts Types of Cryptography: ● Symmetric Key (AES, DES basics) ● Asymmetric Key (RSA basics) **Substitution Ciphers:** Ceaser, Monoalphabetic, Playfair, Hill, Polyalphabetic, One-tine Pad or Vernam. **Transposition Ciphers:** Single Columnar, Double Columnar. Cryptanalysis, Steganography. **Data Encryption Standard: Working** of DES **Symmetric Ciphers:** Blowfish and IDEA (International Data Encryption Standard)
Module 3 (Credit 1) - System & Application Security	
Learning Outcomes	Understand public key cryptosystem, different authentication and authorization techniques
	Understand nature of digital signatures
	Understand security system in different areas
Content Outline	● **Public Key Cryptosystems:** Introduction, Public Key Cryptography, RSA Algorithm (Working of RSA, Key length and Security) ● **Authentication and Authorization techniques:** Introduction, Authentications Methods (Password-based, Two-factor, Biometric, Extensible). ● **Digital Signatures:** Introduction, Implementation, Association of Digital Signatures and Encryption, Using Different Key pairs for Signing and Encryption. ● Operating System Security ● Access Control Models (DAC, MAC, RBAC) ● Database Security Concepts ● Web Application Security (SQL Injection, XSS, CSRF) ● Secure Coding Practices ● Cloud Security Basics ● Mobile Security Issues
Module 4 (Credit 1) - Cyber Forensics, Cyber Laws, Governance & Emerging Trends	
Learning Outcomes	Understand different cyber laws
	Understand the nature of Digital Forensics
	Different areas and case studies of network security

Content Outline	**Cyber Laws:** Introduction to cyber laws, Cyber Security Regulations, Cyber Security Standards, Indian Cyber Space, Cyber Laws in India (IT Act 2000 & Amendments) National Cyber security, **DPDP Act, 2023 (**The Digital Personal Data Protection Act, 2023) **Digital Forensic:** Introduction to cybercrimes & Digital Forensic, Types of Digital Forensics, Digital Forensics Process, Areas of Application of computer forensics, Understanding the Suspects, Examples of Computer Forensics, Free space and Slack Space. **Case Studies on Cryptography and security:** Cryptographic Solutions, Denial of Service (DOS) attacks, IP Spoofing attacks

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)

Module 1:

1. Assignment 1: Define cybersecurity and explain its importance in today's digital world Differentiate between: Threat vs Vulnerability vs Risk
2. Explain **CIA Triad** with real-life examples
3. Activity 1: Malware Analysis (Basic) (Create a comparison table of virus, worms, trojan, ransomware etc)
4. Activity 2: Social Engineering Role Play (Phishing attack scenario)
5. Assignment 2: Cyber Attack Case Study (DOS/DDOS/MITM/etc
6. Activity 3: Security Awareness Poster

Module 2:

Assignment 1: Encryption Practice

- Encrypt and decrypt a message using:
  - Substitution cipher, Transposition cipher, DES and Symmetric

cipher **Activity 1: Cipher Implementation:** Write a simple program using (Python/C/Java) **Activity 2: Steganography Demo**

Module 3:

Assignment 1: RSA Algorithm

Assignment 2: Digital Signature

- Explain:
  - How digital signatures work
  - Difference between encryption & signature
- Draw diagram

Assignment 3: Web Security Report and Cloud security Analysis

Module 4:

1. Assignment 1: Cyber Laws in India
2. Activity 1: Digital Forensics Process
3. Activity 2: Different case stud y analysis and presentation Case Study Analysis and report

Text Books:

1. Atul Kahate, Cryptography and Network Security, McGraw Hill

Reference Books:

1. Cybersecurity Fundamentals: A Real-World Perspective

2. CRYPTOGRAPHY AND INFORMATION SECURITY, THIRD EDITION, PACHGHARE,
3. V. K. Eastern Economy Edition, 2019.

4. Kaufman, C., Perlman, R. & Speciner, M., Network Security, Private Communication
5. in a Public world, 2nd ed., Prentice Hall PTR, 2002
6. Stallings, W., Cryptography and Network Security: Principles and Practice, 3rd ed.,
7. Prentice Hall PTR., 2003.
8. Stallings, W., Network Security Essentials: Applications and Standards, Prentice
9. Hall, 2000
10. A Course in Cryptography, By Heiko Knospe, The Sally Series, AMS.

Assessment:
Internal Assessment: (50 Marks) Evaluation Scheme:

Internal 25 marks Written Test

Internal 25 marks Practical/Project (rubrics as follows)

Component	**Activity**	**Marks**
1	Assignment/Activity/Case study	5
2	Cipher Technique Analysis	5
3	Algorithm implementation/ Activity/Analysis	5
4	Case study Presentation	5
5	Case study report	5
Total		**25**

External Assessment: (50 Marks)
End Semester examination of 50 marks for 2 hours duration will be conducted

Semester – VI

.6.3 A. Major (Elective)

Course Title	**Machine Learning (Major-Elective II)** **[2 theory + 2 practicals]**
Course Credits	**4 Credits**
Course Outcomes	**After going through the course, learners will be able to**
	Understand and explain fundamental concepts of Machine Learning
	Perform data preprocessing, feature engineering, and prepare datasets suitable for Machine Learning models.
	Implement and apply supervised learning algorithms such as Linear Regression, KNN, Decision Trees, and SVM to solve real-world problems.
	Apply unsupervised learning techniques like K-Means clustering and basic dimensionality reduction methods for data analysis.
Module 1(Credit 1) - Introduction to ML & Data Preparation	
Learning Outcomes	Explain Machine Learning concepts, types, and workflow.
	Perform data preprocessing and feature engineering on datasets.
Content Outline	Introduction to Machine Learning and its applications **Types of ML:** Supervised, Unsupervised, Reinforcement Learning ML lifecycle and workflow **Feature Engineering :** Steps in feature engineering **Data preprocessing:** handling missing values(mean, median and mode imputation), Data cleaning, handling outliers, noise removal and feature scaling **Data transformation:** normalization, standardization, encoding categorical variables(Label encoding and one hot encoding) Feature selection and Train-test split
Module 2(Credit 1) -Supervised Learning Techniques	
Learning Outcomes	Apply regression and classification algorithms to solve problems.
	Analyse model performance and handle overfitting using validation techniques.

Content Outline	**Regression:** Linear Regression- Simple and Multiple Linear regression, Logistic regression **Classification algorithms:** ● K-Nearest Neighbors (KNN) ● Decision Tree ● Random Forest ● Support Vector Machine (SVM) Overfitting and underfitting, Bias-Variance tradeoff Cross-validation techniques - K-fold cross validation, performance evaluation and model selection
Module 3(Credit 1) -Unsupervised Learning & Advanced Concepts	
Learning Outcomes	Implement clustering techniques for pattern discovery in data.
	Apply basic dimensionality reduction methods for data analysis.

Content Outline	Clustering: ● K-Means ● Hierarchical Clustering Dimensionality Reduction (PCA – basic concept) Introduction to Association Rules (basic idea) Applications of unsupervised learning
Module 4(Credit 1) -Model Evaluation, Ensembles & Applications	
Learning Outcomes	Evaluate ML models using performance metrics and interpret results
	Develop ML-based solutions using ensemble techniques and understand ethical considerations.
Content Outline	Confusion Matrix Performance metrics: Accuracy, Precision, Recall, F1-score ROC Curve and AUC Ensemble learning: Bagging, Boosting, Stacking (basic idea) Introduction to AI Ethics and bias Case studies and real-world ML applications
Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)	
Module 1: 1. Write a short report on applications of Machine Learning in real life 2. Explain differences between supervised and unsupervised learning with examples 3. Perform data preprocessing on a given dataset (handle missing values, normalization) 4. Implement train-test split using Python (Scikit-learn)	

Module 2: 1. Compare KNN, Decision Tree, and SVM algorithms (advantages & limitations) 2. Solve a problem using Linear Regression (manual calculation or Python) 3. Implement KNN or Decision Tree using Scikit-learn 4. Perform cross-validation and analyze overfitting/underfitting
Module 3: 1. Explain K-Means clustering algorithm with example 2. Write a short note on PCA and its importance 3. Implement K-Means clustering on a dataset (e.g., customer segmentation) 4. Visualize clusters and interpret results
Module 4: 1. Explain performance metrics (Accuracy, Precision, Recall, F1-score) with examples 2. Write a short report on ensemble learning techniques (Bagging, Boosting, Stacking) 3. Evaluate a model using confusion matrix and performance metrics 4. Mini case study: Compare two ML models and justify the better one

Text Books:

1. Mitchell, T. M. (1997). *Machine learning*. McGraw-Hill.
2. Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow* (2nd ed.). O'Reilly Media.
3. Müller, A. C., & Guido, S. (2016). *Introduction to machine learning with Python: A guide for data scientists*. O'Reilly Media.
4. Kelleher, J. D., Namee, B. M., & D'Arcy, A. (2015). *Fundamentals of machine learning for predictive data analytics: Algorithms, worked examples, and case studies*. MIT Press.

Reference Books:

1. Burkov, A. (2019). *The hundred-page machine learning book*. Andriy Burkov.
2. Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.
3. Murphy, K. P. (2012). *Machine learning: A probabilistic perspective*. MIT Press.
4. Raschka, S., & Mirjalili, V. (2017). *Python machine learning* (2nd ed.). Packt Publishing.
5. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Assessment:
Internal Assessment: (50 Marks) Evaluation Scheme:
Internal written exam for 25 marks
Internal 25 marks for project/practicals (rubrics as follows)

Component	Activity	Marks
1	Assignment (Concept + Analytical Questions)	5
2	Practical Activity-Selection criteria	5

3	Case Study: Mini Project	5
4	Presentation/Viva	5
5	Case study report	5
Total		**25**

Mini ML Project: Students can take up any relevant project like House price prediction, Customer Churn prediction, Student performance prediction etc and the project should include :

- **Problem definition**
- **Data preprocessing**
- **Model building**
- **Model Evaluation/ Model tuning**
- **Result interpretation**
- **Comparison of all models**
- **Final model selection for deployment**

Tools used : Anaconda Jupyter Notebook/ PowerBI/Tableau/KNIME?R-Programming Data set : Available on Kaggle site

External Assessment: (50 Marks)

End Semester examination of 50 marks for 2 hours duration will be conducted

Semester – VI

.6.3 B. Major (Elective)

Course Title	**Blockchain Technology (Major Elective)** **[2 theory + 2 practicals]**
Course Credits	**4 Credits**
Course Outcomes	After going through the course, learners will be able to
	CO1 – Understanding Blockchain Fundamentals
	CO2 – Analysing Block Structure and Operations
	CO3 – Applying Cryptographic Techniques
	CO4 – Implementing Digital Signatures and Blockchain Security
	CO5 – Understanding Blockchain Networking and Cryptocurrency
Module 1(Credit 1) - Introduction to Blockchain and overview of Blocks	
Learning Outcomes	Understand the fundamental concepts of blockchain
	Apply their knowledge to illustrate the structure of a blockchain block by identifying its key components such as the header, previous hash, timestamp, nonce, and transaction data.
	Analyse the impact of distributed systems and Moore's Law on blockchain performance and scalability.
Content Outline	What is a blockchain? Origin of blockchain, Blockchain Technology, Distributed systems, History of blockchain and Bitcoin, Types of blockchain. Overview of Blocks: (Block attributes, Structure of block, block header, Linking block) Influence of Moore's Law on Blockchain technology.
Module 2(Credit 1) - Cryptography and Hashing Algorithms	
Learning Outcomes	Understand the fundamental concepts of cryptography
	Apply symmetric and asymmetric cryptographic techniques
	Analyse hashing algorithms and cryptographic methods
Content Outline	Cryptography: Concepts, Classical Cryptography, Cryptographic Primitives, Symmetric key cryptography: (Stream, Cipher, Block cipher, Data Encryption Standard (DES), Advanced Encryption Standard (AES), Implementation of AES Asymmetric key cryptography: (Prime Factorization, Discrete logarithm, Eliptic – curve, RSA algorithm). Hashing Algorithms: Message Digest, SHA, Hashing example using an SHA -256 algorithm
Module 3(Credit 1) - Digital Signatures and Blockchain Security	
Learning Outcomes	Understand The principles and working of digital signatures in blockchain

	Apply the concepts of digital signatures and blockchain wallets to illustrate asset transfer, transaction transmission.
	Analyse blockchain security models and associated vulnerabilities,
Content Outline	Digital Signatures: Working, Signing Process, Verification Process, creating an identity, Signature in transaction, asset ownership in blockchain – (Transferring an asset, Transmitting the transaction, claiming the asset), Blockchain Wallet. Blockchain Security: Transaction Security Model: (Risks of the Security model) Decentralized Security Model: (Centralization due to cryptocurrency exchange, Centralization in mining pool) Attacks on the Blockchain: (Double-spend attacks, 51% attack, Eclipse attacks)
Module 4(Credit 1) -Networking in Blockchain and Blockchain Optimization	
Learning Outcomes	Understand the fundamentals of blockchain networking
	Apply the concepts of cryptocurrency and blockchain networks to describe keys and addresses, transactions, mining
	Analyse blockchain optimization techniques, including transaction exchange, off-chain transactions, and block size improvements
Content Outline	Networking in Blockchain: Peer – to – Peer (P2P) networking: (History of P2P network, P2P networking architecture), Network Discovery, Block Synchronization Cryptocurrency: (Basics, Key and address, Transactions, Mining and Consensus, Block Structure and Merkle Trees, Blockchain Networks viz. Testnet, Regtest and Bitcoin hard forks and altcoins). Blockchain Optimization: Transaction Exchange, Off-chain Transactions, Block size improvements
Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)	
Module1: ● Draw and label the structure of a blockchain block, including block header, previous hash, timestamp, nonce, and transaction data ● Create a timeline showing the history and evolution of blockchain and Bitcoin with key milestones. ● Prepare a table comparing types of blockchain (public, private, consortium) with features and use cases. ● Explain the influence of Moore's Law on blockchain technology in terms of processing power and scalability. ● Analyze how Bitcoin uses blockchain technology to ensure security, transparency, and decentralization.	

Module2:

- Create a concept map covering cryptography basics, classical cryptography, and cryptographic primitives.
- Prepare a table comparing symmetric key cryptography (DES, AES, stream vs block cipher) and asymmetric key cryptography (RSA, elliptic curve) with features and uses
- Illustrate the working of the AES or RSA algorithm using a step-by-step example.
- Demonstrate hashing by generating a hash using the SHA-256 algorithm and explain its properties.
- Apply cryptographic techniques to secure a simple message by performing both encryption (symmetric or asymmetric) and hashing, then explain the process.

Module3: Mini Project: Digital Signatures & Blockchain Security

- Digital Signatures: Working, Signing & Verification
- Blockchain Wallet & Identity Creation
- Signature in Transactions
- Asset Ownership & Transfer Process
- Transaction Transmission & Claiming Ownership
- Transaction Security Model & Risks
- Decentralized Security Model
- Centralization in Exchanges & Mining Pools
- Double-Spend Attack
- 51% Attack
- Eclipse Attack

Module4:

- Illustrate the steps of block synchronization and network discovery in a blockchain network using a flowchart.
- Write a short note explaining cryptocurrency basics, including keys, addresses, transactions, mining, and consensus.
- Draw the structure of a block with a Merkle Tree and explain how it ensures data integrity.
- Explain different blockchain optimization techniques such as off-chain transactions, transaction exchange, and block size improvements with examples.

References - Text Books:

1. Koshik Raj, Foundations of Blockchain, Packt, Birmingham, Mumbai
2. Atul Kahate, Cryptography and Network Security, McGraw Hill
3. Reference Books:
4. Kaufman, C., Perlman, R.,& Speciner, M., .Network Security, Private Communication in a Public world, 2nd ed., Prentice Hall PTR, 2002
5. Stallings, W., Cryptography and Network Security: Principles and Practice, 3rd ed., Prentice Hall PTR., 2003
6. Stallings, W., Network Security Essentials: Applications and

Standards, Prentice Hall, 2000
7. Van Haren, Introduction to Blockchain Technology, Van Haren Publishing,20 October 2019 by Melanie Swan, Blockchain, O'Reilly; 1 edition, 6 February 2015
8. Clarke, A.C., "Hazards of Prophecy: The Failure of Imagination," from Profiles of the Future: An Inquiry into the Limits of the Possible, 1962.

9. Satoshi Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System", https://bitcoin.org/bitcoin.pdf
10. Gautam N. Ramamoorthy, Samiha Z. Khan, "BITCOIN: BLOCKCHAIN BASED PEER TO PEER PAYMENT SYSTEM",2020 IJRAR March 2020,
Volume 7, Issue 1, http://www.ijrar.org/papers/IJRAR2001834.pdf

Assessment:
Internal Assessment: (50 Marks) Evaluation Scheme:
Internal written exam for 25 marks
Internal 25 marks for project/practicals (rubrics as follows)

Component	**Activity**	**Marks**
1	Assignment (Concept + Analytical Questions)	5
2	Case Study Analysis	5
3	Activity	5
4	Presentation	10
Total		**25**

External Assessment: (50 Marks)
End Semester examination of 50 marks for 2 hours duration will be conducted

Semester – VI

.6.3 C. Major (Elective)

Course Title	**Natural Language Processing (Major Elective)** **[2 Theory + 2 Practical]**
Course Credits	**4 Credits**
Course Outcomes	**After going through the course, learners will be able to**
	Demonstrate understanding and explain core NLP concepts, linguistic structures, and processing techniques
	Apply NLP methods using Python for text preprocessing and analysis
	Analyse syntactic, semantic, and probabilistic models for natural language
	Evaluate and interpret language data for meaning, ambiguity, and discourse structure
	Develop NLP pipelines for text preprocessing, feature extraction, and analysis using Python libraries such as NLTK
Module 1(Credit 1) - Foundations **of NLP & Text Preprocessing**	
Learning Outcomes	**Understand** fundamental NLP concepts and applications
	Explain linguistic structures such as morphology and word formation
	Apply text preprocessing techniques using NLTK
	Analyse textual features using tagging and vectorization methods

Content Outline	• Introduction to NLP
	• Applications and Use Cases
	• Components and Steps in NLP
	• Linguistic Foundations:
	  • Words and their components
	  • Lexemes, Morphemes, Morphology
	  • Morphological Typology
	  • Challenges in Morphological Processing
	• NLP using Python (NLTK):
	  • Word Tokenization
	  • Sentence Tokenization
	  • Stop Word Removal
	  • Stemming & Lemmatization
	  • POS Tagging
	  • Chunking & Chinking
	  • Named Entity Recognition (NER)
	  • TF-IDF

Module 2(Credit 1) - Syntax **Analysis and Parsing**	
Learning Outcomes	**Explain** syntactic structures and parsing techniques
	Apply parsing algorithms to sentence analysis
	Analyse syntactic ambiguity using probabilistic models
	Differentiate between generative and discriminative approaches

Content Outline	• Parsing Natural Language • Treebanks and Data-driven Approaches • Representation of Syntax: • Phrase Structure Trees • Dependency Graphs • Parsing Algorithms: • Shift Reduce Parsing • Chart Parsing (CYK Algorithm) • Ambiguity Resolution: • Probabilistic Context-Free Grammar (PCFG) • Generative Models • Discriminative Models
Module 3 (Credit 1) - Language **Modelling**	

<table>
<tr><td rowspan="4">Learning Outcomes</td><td>Understand concepts of language modeling</td></tr>
<tr><td>Apply N-gram models for text prediction</td></tr>
<tr><td>Analyse model performance using evaluation metrics</td></tr>
<tr><td>Interpret challenges in real-world language modeling</td></tr>
<tr><td>Content Outline</td><td>● Introduction to Language Modeling
● N-Gram Models
● Language Model Evaluation
● Concept of evaluation
● Why evaluation is needed
● Perplexity (only intuition, not heavy derivation)
● Simple accuracy-based comparison
● Parameter Estimation Techniques
● Maximum Likelihood Estimation (MLE) – concept
● Idea of probability calculation from frequency
● Basic smoothing idea (e.g., Laplace smoothing)
● Language Model Adaptation
● Types of Language Models
● Language-Specific Modeling Challenges</td></tr>
<tr><td colspan="2">Module 4 (Credit 1) - Semantic Analysis & Discourse Processing</td></tr>
<tr><td rowspan="4">Learning Outcomes</td><td>Explain semantic interpretation and meaning representation</td></tr>
<tr><td>Apply techniques for word sense and semantic analysis</td></tr>
<tr><td>Analyse sentence meaning using predicate-argument structures</td></tr>
<tr><td>Evaluate discourse coherence and reference resolution</td></tr>
<tr><td>Content Outline</td><td>● Semantic Parsing and Interpretation
● Word Sense Systems
● Predicate-Argument Structure
● Meaning Representation Systems
Discourse Processing:
● Cohesion
● Reference Resolution
● Discourse Structure</td></tr>
<tr><td colspan="2">Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)</td></tr>
</table>

Module 1:

1. **Text Preprocessing Lab (Hands-on)**
   - Perform tokenization, stop-word removal, stemming, lemmatization on a dataset (news/reviews)
   - Compare outputs of stemming vs lemmatization
2. **NLP Pipeline Mini Task**
   - Build a basic preprocessing pipeline using Python (NLTK)
3. **Short Analytical Assignment**
   - Explain morphological concepts with real examples from English or any Indian language
4. **Class Activity**
   - Identify POS tags and named entities manually from a paragraph, then verify using code

Module 2:

1. **Parsing Implementation Task**
   - Generate parse trees (Phrase Structure / Dependency) for given sentences
2. **Algorithm-Based Assignment**
   - Dry run of Shift Reduce Parsing and CYK Algorithm on sample sentences
3. **Case Study**
   - Analyze ambiguity in sentences and explain how PCFG resolves it
4. **Class Activity**
   - Construct parse trees manually and validate with tools

Module 3:

1. **N-Gram Model Development**
   - Build unigram, bigram, trigram models using a dataset
2. **Probability Calculation Task**
   - Compute probabilities of sentences using N-gram models
3. **Model Evaluation**
   - Compare models using perplexity or accuracy
4. **Mini Application**
   - Develop a simple text predictor (next-word prediction)

Module 4:

1. **Semantic Analysis Task**
   - Perform word sense disambiguation on ambiguous words
2. **Mini Project**
   - Build a basic NLP application (choose one):
     - Chatbot
     - Text summarizer
     - Sentiment analyzer
3. **Discourse Analysis**
   - Analyze cohesion and reference resolution in a paragraph
4. **Presentation**
   - Present working model + explanation of approach

Text Books:

- Daniel Jurafsky, & James H. Martin. (2009). *Speech and Language Processing* (2nd ed.). Pearson Education.

- Daniel M. Bikel, & Imed Zitouni. (2012). *Multilingual Natural Language Processing: From Theory to Practice*. Pearson Education.

Reference Books:

- Tanveer Siddiqui and U. S. Tiwary, *Natural Language Processing and Information Retrieval*. Oxford University Press, 2008.

Assessment:
Internal Assessment: (50 Marks) Evaluation Scheme:

Internal 25 marks Written Test Internal 25 marks Practical/Project

Component	**Activity**	**Marks**
1	Assignment (Concept + Analytical Questions)	5
2	Case Study Analysis (Real-world NLP Applications)	5
3	NLP Pipeline Design (Workflow + Architecture)	5
4	Model Development & Evaluation Assignment	5
5	Mini Project + Presentation	5
Total		**25**

Given below are sample project but it is expected to work on similar sort of projects

1. Fake News Detection
2. Resume Screening System
3. Spam Email Classifier

External Assessment: (50 Marks)
End Semester examination of 50 marks for 2 hours duration will be conducted

Semester – VI

.6.4 Minor Stream

Course Title	**IT Project Management [2 Theory]**
Course Credits	**2 Credits**
Course Outcomes	After going through the course, learners will be able to
	Define and **explain** fundamental concepts, phases, and constraints of project management.
	Apply project planning, scheduling, and estimation techniques to manage simple projects.
	Analyze and **evaluate** project risks, team dynamics, and project performance.
	Design and **develop** basic project plans, reports, and closure documents.
Module 1(Credit 1) - Project Fundamentals & Initiation	
Learning Outcomes	**Define** and **explain** project management concepts, life cycle, and triple constraints.
	Differentiate and **analyze** project roles, organizational structures, and team dynamics.
	Apply basic project selection methods and develop simple project charters/proposals.
	Analyze and **evaluate** team conflicts and leadership approaches in project scenarios.
Content Outline	● Definition of Project and Project Management ● Necessity of Project Management ● Triple Constraints (Scope, Time, Cost) ● Project Life Cycle and Phases ● Role and Qualities of Project Manager ● Leadership and Ethics in Project Management ● Organizational Structures in Projects ● Project Initiation Process ● Project Selection Methods (Numeric & Non-Numeric Models – basic idea) ● Project Charter and Project Proposal ● Team Formation and Development ● Conflict and Negotiation Basics

Module 2(Credit 1) - Project Planning, Execution & Closure	
Learning Outcomes	**Explain** and **apply** project planning tools such as WBS, scheduling techniques, and budgeting methods.
	Analyze project risks, resource allocation, and project performance using basic metrics.
	Evaluate project progress using monitoring and control techniques including earned value concepts.
	Design and **develop** basic project plans, reports, and closure documents for small projects.
Content Outline	• Project Planning and Work Breakdown Structure (WBS) • Project Cost Estimation and Budgeting (Top-down & Bottom-up) • Project Scheduling (Gantt Chart, PERT, CPM – basic concepts only) • Resource Allocation and Levelling • Risk Management (Identification, Risk Register, Basic Assessment) • Project Monitoring and Control • Earned Value Management (basic idea) • Change Management and Scope Creep • Project Closure Process • Project Reports and Lessons Learned
Activities towards Comprehensive Continuous Evaluation (CCE)	
Module 1: Activities: • Conceptual Assignment: Students explain project management concepts such as project life cycle, triple constraints, and roles of a project manager with suitable examples. • Project Proposal Preparation: Students develop a simple project proposal including objectives, scope, stakeholders, and a basic project charter. • Class Test / Quiz: Assessment based on fundamental concepts, organizational structures, and project selection methods. • Case-based Discussion: Students analyze basic project scenarios involving team formation, leadership, and conflict situations.	

Module 2:

Activities:

- Project Planning Assignment: Students prepare a Work Breakdown Structure (WBS), basic schedule (Gantt chart), and cost estimation for a small project.
- Risk Identification Activity: Students identify potential risks in a given project scenario and prepare a simple risk register.
- Mini Project / Case Study: Students develop a complete project plan including scheduling, resource allocation, monitoring approach, and reporting.
- Project Report & Closure Document: Students submit a final report including lessons learned and project closure summary.

References – Text Books:

1. Larson, E. W., & Gray, C. F. (2020). Project Management: The Managerial Process (8th ed.). McGraw-Hill Education.
2. Schwalbe, K. (2021). Information Technology Project Management (9th ed.). Cengage Learning.

Reference Books:

1. Kerzner, H. (2017). Project Management: A Systems Approach to Planning, Scheduling, and Controlling (12th ed.). Wiley.
2. Heldman, K. (2018). Project Management JumpStart (4th ed.). Wiley.
3. Meredith, J. R., Shafer, S. M., & Mantel, S. J. (2017). Project Management: A Managerial Approach (10th ed.). Wiley.

Assessment:

External Assessment: (50 Marks)

End Semester examination of 50 marks for 2 hours duration will be conducted

Semester – VI

.6.5 Minor Stream

Course Title	**Big Data Analytics** **[2 theory + 2 practicals]**
Course Credits	**4 Credits**
Course Outcomes	After going through the course, learners will be able to
	Explain the concepts of Big Data, characteristics, architecture and Big Data ecosystem.
	Apply Hadoop ecosystem tools such as HDFS, MapReduce and Spark for big data processing.
	Analyze large datasets using distributed processing frameworks and big data analytics techniques.
	Evaluate big data platforms and data processing methods for different types of data.
Module 1(Credit 1) - Introduction to Big Data	
Learning Outcomes	Understand Big Data concepts and characteristics
	Identify Big Data architecture and ecosystem components
	Understand Applications of Big Data
Content Outline	**Introduction to Big Data** ● Definition of Big Data ● Evolution of Big Data ● Traditional Data vs Big Data **Characteristics of Big Data:** 5 V's of Big Data, Volume Velocity, Variety, Veracity, Value **Big Data Architecture** ● Data sources ● Data storage ● Data processing ● Data analysis and visualization **Big Data Ecosystem** ● Hadoop ecosystem overview ● Tools used in Big Data environment **Applications of Big Data** ● Healthcare ● Banking and finance ● E-commerce ● Social media analytics ● Smart cities

<table>
<tr><td colspan="2">Module 2(Credit 1) - Hadoop Distributed File System (HDFS)</td></tr>
<tr><td rowspan="3">Learning Outcomes</td><td>Understand Hadoop architecture</td></tr>
<tr><td>Understand distributed file systems</td></tr>
<tr><td>Learn to store and manage large datasets using HDFS</td></tr>
<tr><td>Content Outline</td><td>Introduction to Hadoop
● Hadoop architecture overview
● Components of Hadoop ecosystem
Hadoop Distributed File System (HDFS)
● Concept of distributed storage
● HDFS architecture
HDFS Components
● NameNode
● DataNode
● Secondary NameNode
HDFS Operations
● Storing files in HDFS
● Block storage and replication
● Fault tolerance and reliability
HDFS Commands
● File operations in Hadoop
● Uploading and downloading data
● File management operations</td></tr>
<tr><td colspan="2">Module 3(Credit 1) -MapReduce and Distributed Data Processing</td></tr>
<tr><td rowspan="3">Learning Outcomes</td><td>Understand MapReduce programming model</td></tr>
<tr><td>Learn distributed data processing techniques</td></tr>
<tr><td>Apply MapReduce for processing large datasets</td></tr>
</table>

Content Outline	**MapReduce Framework** ● Introduction to MapReduce ● Working of Map and Reduce functions **MapReduce Architecture** ● Job tracker and task tracker ● Data flow in MapReduce **MapReduce Programming Model** ● Mapper function ● Reducer function ● Combiner and partitioner **Applications of MapReduce** ● Word count program ● Log data analysis ● Large scale data processing
Module 4(Credit 1) -Big Data Analytics using Spark	
Learning Outcomes	Understand Apache Spark framework
	Learn big data processing using Spark
	Perform analytics on large datasets
Content Outline	**Introduction to Apache Spark** ● Spark architecture ● Advantages over MapReduce **Spark Components** ● Spark Core ● Spark SQL ● Spark Streaming ● MLlib **Data Processing in Spark** ● Resilient Distributed Datasets (RDD) ● Data transformations and actions **Big Data Analytics** ● Data analysis using Spark ● Machine learning with MLlib ● Real-time analytics
Assignments / Activities towards Comprehensive Continuous Evaluation (CCE)	

Module 1 ● Write a short note on Big Data and its characteristics. ● Identify real world applications of Big Data. ● Case study analysis of Big Data usage in business. ● Compare traditional data processing with Big Data processing.
Module 2 ● Install Hadoop environment (pseudo distributed mode). ● Perform HDFS file operations using commands. ● Upload datasets into HDFS. ● Demonstrate block storage and replication concept.
Module 3 ● Implement simple MapReduce program for word count. ● Perform log data analysis using MapReduce. ● Demonstrate MapReduce workflow. ● Write pseudo code for MapReduce program.
Module 4 ● Install Apache Spark environment. ● Create RDD and perform basic transformations. ● Perform data analysis using Spark SQL. ● Analyse a dataset using Spark MLlib.

Text Book

1. Erl, T., Khattak, W., & Buhler, P. (2016). *Big data fundamentals: Concepts, drivers, and techniques*. Prentice Hall.
2. Demirbaga, Ü., Aujla, G. S., Jindal, A., & Kalyon, O. (2024). *Big data analytics: Theory, techniques, platforms, and applications*. Springer.
3. Kumar, J., Kumar, A., & Kumar, R. (2024). *Big data and analytics*. BPB Publications.
4. Sai Krishna, T. V., Suresh, M., Vivek, K., Mastan Chowdary, C., & Srinu, N. (2024).
   Big data analytics. Royal Book Publishing.

Reference Books

1. White, T. (2015). *Hadoop: The definitive guide* (4th ed.). O'Reilly Media.
2. Chambers, B., & Zaharia, M. (2018). *Spark: The definitive guide: Big data processing made simple*. O'Reilly Media.
3. Acharya, S., & Chellappan, S. (2015). *Big data analytics*. Wiley.
4. Mishra, D., Yang, X.-S., Unal, A., & Jat, D. S. (Eds.). (2025). *Data science and big data analytics: Proceedings of IDBA 2024*. Springer.
5. Dasgupta, A., Kiran, R. U., El Shawi, R., Srirama, S., & Adhikari, M. (Eds.). (2025). *Big data and artificial intelligence: 12th international conference, BDA 2024 proceedings*. Springer.
6. Lin, H., Tang, J., Simic, M., Bhateja, V., & Reddy, V. S. (Eds.). (2024). *Big data analytics and data science: Proceedings of the eighth international conference on information system design and intelligent applications*. Springer.

Assessment:

Internal Assessment (50 Marks) Evaluation Scheme:

Internal 25 marks for written exam
Internal 25 marks for project/practicals (follow the rubrics)

Component	Activity	Marks
1	Concepts and Types of analytics	5
2	Cleaning, transformation	5
3	Charts, dashboards	5
4	Real-world cases Presentation	15
Total		**25**

External Assessment (50 Marks)

End Semester examination of 50 marks for 2 hours duration will be conducted.