Nirmala Memorial Foundation College of Commerce and Science

(Autonomous)

Re-accredited by NAAC with B++, ISO 9001-2015 Certified, Recognised under section 2(f) & 12(B) of the UGC Act 1956

Permanently Affiliated to the University of Mumbai,

Syllabus

Bachelor of Science in Computer Science

(B.Sc. Computer Science)

(Semester V and VI)

(Programme Code: UGCS02)

As Per New Education Policy (NEP) 2020

(To be implemented from the Academic Year 2026-2027)

Approved in the Academic Council Meeting held on ***8th June 2026***

INTRODUCTION OF THE PROGRAMME

1. Introduction

In the era of Information and Communication Technology (ICT), the transformative impact of computers on society is undeniable. The pervasive applications of computing across diverse fields have given rise to dynamic industries, evolving in tandem with the swift pace of technological change. As the landscape of the computing field continues to advance, it becomes imperative for students to cultivate a robust foundation that not only facilitates their current skills but also empowers them to adapt to the evolving nature of the field.

In line with the National Education Policy (NEP) 2020, our revised Computer Science program is designed to instill in students the ability to navigate the ever-changing technological terrain. Recognizing that specific languages and platforms may undergo transformations, the curriculum places a strong emphasis on fostering adaptability. Students will not only be exposed to a diverse array of programming languages, tools, paradigms, and technologies but will also delve into the fundamental principles that underpin the realm of computer science.

The core of our program encompasses essential courses such as programming languages, data structures, computer architecture and organization, algorithms, database systems, operating systems, and software engineering. Complementing these foundational elements are specialized courses in areas such as artificial intelligence, computer-based communication networks, distributed computing, information security, graphics, human-computer interaction, multimedia, scientific computing, web technology, and other cutting-edge topics in computer science.

Key Philosophy of the Program:

- **Form Strong Foundations:** Lay the groundwork for a comprehensive understanding of Computer Science.
- **Nurture Skills:** Develop programming, analytical, and design skills to tackle real-world problems effectively.
- **Introduce Gradually:** Familiarize students with emerging trends in a gradual and coherent manner.
- **Prepare for Industry Challenges:** Groom students to meet the challenges of the ICT industry with confidence and competence.

In acknowledgement of the evolving aspirations of students, our program not only prepares them for careers in the industry but also opens doors to research opportunities. The primary goal is to deliver a modern curriculum that equips graduates with both theoretical depth and practical acumen, empowering them to excel in the workplace while fostering a mindset of lifelong learning.

This program not only paves the way for a successful career in the software industry but also inspires students to pursue further studies and research opportunities. Graduates can seamlessly transition into postgraduate programs in Computer Science, leading to research and development roles, employment in IT industries, or even a career in business management.

As we unveil this syllabus, we invite students on a journey of exploration, learning, and innovation, ensuring they are not only prepared for the present but also poised to shape the future of Computer Science.

2. Aims and Objectives

Understanding and Knowledge Base: Develop a profound understanding and knowledge of the fundamental theories, systems, and applications that form the bedrock of Computer Science. This includes establishing a strong foundation in theoretical concepts and cultivating expertise in the practical application of Computer Science theories.

Analytical Abilities and Problem Solving: Foster essential skills and analytical abilities required for devising computer-based solutions to real-life problems. This involves developing critical thinking skills for problem identification and analysis, as well as cultivating the ability to design and implement effective solutions using computational tools.

Training in Emerging Technologies: Provide training in emergent computing technologies, facilitating the development of innovative solutions for both industry and academia. This includes exposing students to cutting-edge technologies and their applications, as well as encouraging exploration and experimentation with emerging tools and platforms.

Preparation for Post-Graduate Studies: Develop the necessary study skills and knowledge for students to pursue further post-graduate study in Computer Science or related fields. This involves equipping students with the academic rigor required for advanced studies and fostering a passion for continuous learning and research in the field.

Professional Skillset Development: Develop the professional skillset required for a successful career in an information technology-oriented business or industry. This includes providing practical exposure to industry-relevant tools and practices, as well as instilling a sense of professional ethics and responsibility.

Independent and Collaborative Work: Enable students to work independently and collaboratively, communicate effectively, and become responsible, competent, confident, insightful, and creative users of computing technology. This involves cultivating independence in problem-solving and project execution, as well as enhancing communication and collaboration skills for effective teamwork.

PROGRAM OUTCOME

At the end of three-year Bachelor of Computer Science, the students will be able:

- Formulate, model, and design solutions and procedures, utilizing software tools to address real-world problems effectively.
- Design and develop computer programs and computer-based systems in diverse areas such as networking, web design, security, cloud computing, IoT, data science, and other emerging technologies.
- Familiarize themselves with modern-day trends in industry and research-based settings, fostering the ability to innovate novel solutions to existing problems.
- Apply concepts, principles, and theories related to computer science to new and challenging situations.
- Demonstrate proficiency in using current techniques, skills, and tools essential for computing practice.
- Apply standard Software Engineering practices and strategies in real-time software project development.
- Pursue higher studies of specialization and confidently enter technical employment.
- Work independently or collaboratively as effective team members on substantial software projects, showcasing project management and teamwork skills.
- Communicate and present their work effectively and coherently, both in oral and written formats.
- Display ethical conduct in the usage of the Internet and Cyber systems, understanding and adhering to ethical standards in computing practices.
- Engage in independent and life-long learning, adapting to the rapidly changing IT industry and staying abreast of evolving technologies.

SEMESTER V

COURSE STRUCTURE

VERTICAL		COURSE CODE	SUBJECT	CREDIT
Vertical 1	Major	**NUCS501**	Artificial Intelligence	**2**
	Major	**NUCS502**	Information Security	**2**
	Major	**NUCS503**	Indian Knowledge Systems in Cryptography	**2**
	Major	**NUCS504**	CS Practical V - Artificial Intelligence and Information Security- Lab	**2**
	Major	**NUCS505**	Mini Project - I	**2**
	Elective **(Any One)**	**NUCS506** **NUCS508** **NUCS510**	1. Software Testing and Quality Engineering 2. Wireless Sensor Networks. 3. Robotics Process Automation	**2**
	Elective	**NUCS507** **NUCS509** **NUCS511**	1. Software Testing and Quality Engineering **- Lab** 2. Wireless Sensor Networks **- Lab** 3. Robotic Process Automation **- Lab**	**2**
Vertical 2	Minor	**NUCS512**	DS_Deep Learning	**2**
	Minor	**NUCS513**	DS_Deep Learning **- Lab**	**2**
Vertical 4	VSC	**NUCS514**	Full Stack Web Development	**2**
Vertical 6	FP	**NUFP501**	Field Project / CEP	**2**
			Total Credits	**22**

SEMESTER VI

COURSE STRUCTURE

VERTICAL		COURSE CODE	SUBJECT	CREDIT
Vertical 1	Major	**NUCS601**	Data Science	**2**
	Major	**NUCS602**	Big Data Analytics	**2**
	Major	**NUCS603**	Cybersecurity and Threat Management	**2**
	Major	**NUCS604**	CS Practical VI - Data Science-& Big Data Analytics -Lab	**2**
	Major	**NUCS605**	Mini Project -II	**2**
	Elective	**NUCS606** **NUCS608** **NUCS610**	1. Information Retrieval 2. Ethical Hacking 3. Blockchain Technology & Applications	**2**
	Elective **(Any One)**	**NUCS607** **NUCS609** **NUCS610**	1. Information Retrieval - Lab 2. Ethical Hacking - Lab 3. Blockchain Technology & Applications -Lab	**2**
Vertical 2	Minor	**NUCS612**	DS_Machine Learning Algorithms	**2**
	Minor	**NUCS613**	DS_Machine Learning Algorithms - **Lab**	**2**
Vertical 4	VSC		****	**
Vertical 6	FP	**NUOJ601**	OJT	**4**
			Total Credits	**22**

Semester – V

Vertical – I

Mandatory Courses

Name of the Course: Artificial Intelligence

Sr. No.	Heading	Particulars
1	**Description the course:**	**Introduction:** This course introduces students to the foundations of Artificial Intelligence, intelligent agents, problem-solving techniques, knowledge representation, reasoning, and machine learning methods. Students will gain theoretical understanding along with practical skills in AI applications.
2	**Vertical:**	**Major**
3	**Type:**	**Theory**
4	**Credits:**	**2 credits**
5	**Hours Allotted:**	**30 Hours**
6	**Marks Allotted:**	**50 Marks**
7	**Course Objectives:** 1. Understand AI concepts, history, and applications. 2. Design intelligent agents and solve problems using search strategies. 3. Apply knowledge representation, reasoning, and machine learning techniques.	
8	**Course Outcomes:** **CO1.** AI Foundations & History: Understand the evolution and current state of AI, including intelligent agents and their environments. **CO2.** Problem-Solving Strategies: Explore uninformed and informed search techniques for effective problem-solving. **CO3.** Knowledge Representation & Reasoning: Learn methods for representing and reasoning with knowledge to solve complex problems. **CO4.** Machine Learning Techniques: Gain insight into classification, regression, and ensemble learning for data-driven decision-making.	
9	**Modules: -** **Module 1 (15 hours):**	
	● **What is AI:** Foundations, history, and current state-of-the-art AI ● **Intelligent Agents:** ○ Agents and environments ○ Types and characteristics of environments ○ Structure of agents ● **Problem Solving by Searching:** ○ Problem-solving agents ○ Uninformed search strategies (BFS, DFS, Uniform Cost) ○ Informed (heuristic) search strategies (A*, Greedy Best-First Search)	

<table>
<tr><td></td><td colspan="2">Module 2 (15 hours):</td></tr>
<tr><td></td><td colspan="2">

Knowledge Representation and Reasoning:

Knowledge representation techniques and forms
Reasoning and planning
Handling uncertainty in knowledge
Introduction to Fuzzy Logic and Fuzzification

Machine Learning for AI:

Forms of learning: Supervised, Unsupervised, Reinforcement
Parametric & Non-Parametric Models
Classification & Regression
Decision Trees, SVM, Artificial Neural Networks
Ensemble Learning, Boosting
K-NN, Gradient Descent

</td></tr>
<tr><td>10</td><td colspan="2">Reference Books
1. Artificial Intelligence: Foundations of Computational Agents, David L Poole,Alan K. Mackworth, 2nd Edition, Cambridge University Press ,2017.
2. Artificial Intelligence, Kevin Knight and Elaine Rich, 3rd Edition, 2017
3. The Elements of Statistical Learning, Trevor Hastie, Robert Tibshirani and Jerome Friedman, Springer, 2013</td></tr>
<tr><td>11</td><td>Internal Continuous Assessment: 40%</td><td>Semester End Examination: 60%</td></tr>
<tr><td>12</td><td>Continuous Evaluation through:
Class test of 15 marks
Quizzes/ Presentations/ Assignments: 5 marks
Total: 20 marks</td><td>Format of Question Paper: External Examination (30 Marks)– 1 hr duration</td></tr>
<tr><td>13</td><td colspan="2">Format of Question Paper: (Semester End Examination: 30 Marks. Duration:1 hour)
<table>
<tr><th>Questions</th><th>Based On</th><th>Options</th><th>Marks</th></tr>
<tr><td>Q.1</td><td>Module 1</td><td>Any 2 out of 4</td><td>10</td></tr>
<tr><td>Q.2</td><td>Module 2</td><td>Any 2 out of 4</td><td>10</td></tr>
<tr><td>Q.3</td><td>Module 1 & 2</td><td>Any 2 out of 4</td><td>10</td></tr>
</table>
</td></tr>
</table>

Name of the Course: Information Security

Sr. No.	Heading	Particulars
1	**Description of the course:**	**Introduction**: This course provides an in-depth understanding of the principles and techniques used in computer and network security. Students will explore various security topics, including encryption techniques, public-key cryptography, key management, message authentication, digital signatures, authentication protocols, network security, web security, intrusion detection, malicious software, and firewall design principles. Through theoretical learning and practical exercises, students will develop the necessary knowledge and skills to analyze, design, and implement secure systems and protect against security threats.
2	**Vertical:**	Major
3	**Type:**	Theory
4	**Credits:**	2 credits (1 credit = 15 Hours for Theory)
5	**Hours Allotted:**	30 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives:** 1. Familiarize students with the fundamental principles, models, and mechanisms of computer and network security. 2. Explore various encryption techniques, including symmetric and public-key cryptography, and understand their strengths, weaknesses, and real-world applications. 3. Examine different authentication and key management methods to ensure secure communication and protect against unauthorized access. 4. Understand the concepts and techniques of message authentication, digital signatures, and authentication protocols used in secure communication systems.	
8	**Course Outcomes:** CO1. Security Trends & Mechanisms: Analyze security trends, attacks, and propose solutions based on the OSI security architecture. CO2. Classical Encryption: Apply encryption techniques like substitution and transposition ciphers, and evaluate their security implications. CO3. Public-Key Cryptography: Implement RSA and other algorithms to securely exchange keys and establish secure communication channels. CO4. Authentication & Security Measures: Design authentication mechanisms (e.g., MACs, digital signatures) and evaluate security protocols (IPsec, SSL/TLS) to safeguard systems and networks.	
9	**Modules** **Module 1 (15 hours):**	
	1. **Introduction to Security:** ○ Security trends and challenges ○ OSI Security Architecture ○ Security attacks, services, and mechanisms	

	2. **Classical Symmetric Encryption Techniques:** ○ **Substitution Ciphers:** ■ Caesar cipher ■ Monoalphabetic cipher ■ Playfair cipher – digraph substitution ■ Polyalphabetic cipher ○ **Transposition Ciphers:** ■ Rail fence cipher ■ Columnar transposition cipher ○ **Steganography** 3. **Block and Stream Ciphers (Symmetric Algorithms):** ○ Principles of block ciphers ○ Data Encryption Standard (DES) ○ Strength of DES ○ Advanced Encryption Standard (AES) – conceptual overview ○ Multiple encryption and Triple DES ○ Block cipher modes of operation (ECB, CBC, CFB, OFB) ○ Stream ciphers	
	Module 2 (15 hours):	
	1. **Public-Key (Asymmetric) Cryptography:** ○ Principles of asymmetric encryption ○ RSA Algorithm (conceptual understanding) 2. **Key Management:** ○ Public-key cryptosystems ○ Key management techniques ○ Diffie-Hellman key exchange 3. **Message Authentication and Hash Functions:** ○ Authentication requirements and functions ○ Message Authentication Codes (MAC) ○ Hash functions, SHA, and HMAC 4. **Digital Signatures and Authentication:** ○ Principles of digital signatures ○ Applications for authentication	
10	**Reference Books** 1. Cryptography and Network, 2nd edition, Behrouz A Fourouzan, Debdeep Mukhopadhyay, TMH. 2. Atul Kahate, “Cryptography and Network Security”, Tata McGraw-Hill.	
11	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
12	**Continuous Evaluation through:** Class test of 15 marks Quizzes/ Presentations/ Assignments: 5 marks Total: 20 marks	**Format of Question Paper: External Examination (30 Marks)– 1 hr duration**

13	**Format of Question Paper: (Semester End Examination: 30 Marks. Duration:1 hour)**

Questions	Based On	Options	Marks
Q.1	Module 1	Any 2 out of 4	10
Q.2	Module 2	Any 2 out of 4	10
Q.3	Module 1 & 2	Any 2 out of 4	10

Name of the Course: Indian Knowledge Systems in Cryptography

Sr. No.	Heading	Particulars
1	**Description of the course:**	**Introduction:** This course introduces students to the foundations of Indian Knowledge Systems (IKS) with special emphasis on ancient Indian cryptography, secure communication practices, strategic intelligence, and their relevance to modern cyber security. The course explores India's contributions to logic, mathematics, coding systems, secret communication, and knowledge protection mechanisms described in ancient texts such as Arthashastra and classical Indian scholarly traditions. It further examines modern cryptographic techniques, cyber security principles, blockchain, secure communication systems, and ethical computing through Indian philosophical perspectives. The course aims to bridge traditional Indian security concepts with contemporary information and cyber security technologies.
2	**Vertical:**	Major
3	**Type:**	Theory
4	**Credits:**	2 credits
5	**Hours Allotted:**	30 hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives:** 1. To introduce the concepts and significance of Indian Knowledge Systems in the context of information security. 2. To understand ancient Indian approaches to cryptography, strategic intelligence, and secure communication. 3. To examine Indian contributions to mathematics, logic, and computational thinking relevant to cryptography. 4. To familiarize students with modern cryptographic techniques and cyber security fundamentals. 5. To explore the role of emerging technologies in secure communication and information protection. 6. To develop ethical and responsible computing practices inspired by Indian philosophical traditions.	
8	**Course Outcomes:** After successful completion of the course, learners will be able to: **CO1.** Explain the foundations and significance of Indian Knowledge Systems related to cryptography and security. **CO2.** Describe ancient Indian methods of secure communication, strategic intelligence, and knowledge protection.	

	CO3. Relate Indian mathematical and logical traditions to modern cryptographic and cyber security concepts. **CO4.** Analyze the applications of cryptography, blockchain, and cyber security technologies in information protection. **CO5.** Evaluate ethical and secure computing practices from Indian philosophical and strategic perspectives.
9	**Modules: -** **Module 1 :** Foundations of Indian Knowledge Systems and Ancient Cryptography (15 Hours)
	1. Introduction to Indian Knowledge Systems (IKS) • Meaning, scope, and significance of IKS • Characteristics of Indian Knowledge Traditions **2. Sources of Indian Knowledge** • Vedas and Upavedas • Arthashastra and strategic intelligence • Ancient Indian communication systems **3. Indian Knowledge Traditions in Security and Governance** • Ancient Indian administrative and intelligence systems • Role of spies and coded communication in Arthashastra • Knowledge protection practices in ancient India **4. Indian Mathematical and Computational Heritage** • Concept of Zero and Decimal Number System • Aryabhata and Bhaskaracharya • Vedic Mathematics and logical patterns **5. Indian Logic and Computational Thinking** • Panini's Grammar and rule-based systems • Nyaya Logic and reasoning techniques **6. Relevance of IKS in Modern Information Security** • Traditional security concepts and modern computing • Indigenous approaches to secure communication
	Module 2: Cryptography and Cyber Security Technologies (15 Hours)
	1. Ancient Indian Cryptography and Secret Communication • Concept of cryptography in ancient India • Secret communication methods in ancient kingdoms • Coded language and symbolic communication systems • Role of cryptography in warfare and administration **2. Arthashastra and Intelligence Systems** • Kautilya's Arthashastra and espionage techniques • Spy networks and intelligence gathering • Use of coded messages and secret symbols • Strategic communication and information protection **3. Indian Mathematical Foundations of Cryptography** • Number systems and cryptographic relevance • Vedic Mathematics and pattern recognition • Logical structures in Indian computational traditions • Contributions of Aryabhata and Bhaskaracharya **4. Traditional Knowledge Protection Systems** • Preservation and protection of manuscripts • Knowledge transfer methods in Gurukul systems

<table>
<tr><td></td><td colspan="2">

Palm leaf manuscripts and coded documentation
Indigenous methods of safeguarding information

5. Modern Cryptography and Cyber Security

Fundamentals of Encryption and Decryption
Symmetric and Asymmetric Cryptography
Hash Functions and Digital Signatures
Authentication and Access Control

6. Emerging Technologies in Information Security

Blockchain for secure information management
Artificial Intelligence in Cyber Security
Ethical hacking and cyber defense
Ethical computing from Indian philosophical perspectives

</td></tr>
<tr><td>10</td><td colspan="2">Reference Books

Mahadevan, B., Bhat, V. R., & Nagendra, P. (2022). Introduction to Indian Knowledge System. PHI Learning Pvt. Ltd.
Kangle, R. P. (2010). Kautilya's Arthashastra. Motilal Banarsidass Publishers.
Kapoor, K., & Singh, A. K. (2017). Indian Knowledge Systems: An Introduction. D.K. Printworld.
Rao, S. B. (2019). Computing Science in Ancient India. Bharatiya Vidya Bhavan.
Rajaraman, V. (2012). History of Computing in India: From Ancient Times to the Present Era. PHI Learning.
Stallings, W. (2020). Cryptography and Network Security: Principles and Practice. Pearson Education.
Paar, C., & Pelzl, J. (2010). Understanding Cryptography. Springer.
Forouzan, B. A. (2017). Cryptography and Network Security. McGraw Hill Education.
</td></tr>
<tr><td>11</td><td>Internal Continuous Assessment: 40%</td><td>Semester End Examination: 60%</td></tr>
<tr><td>12</td><td>Continuous Evaluation through:
Class test of 15 marks

Quizzes/ Presentations/ Assignments: 5 marks
Total: 20 marks</td><td>Format of Question Paper: External Examination (30 Marks)– 1 hr duration</td></tr>
<tr><td>13</td><td colspan="2">Format of Question Paper: (Semester End Examination: 30 Marks. Duration:1 hour)
<table>
<tr><th>Questions</th><th>Based On</th><th>Options</th><th>Marks</th></tr>
<tr><td>Q.1</td><td>Module 1</td><td>Any 2 out of 4</td><td>10</td></tr>
<tr><td>Q.2</td><td>Module 2</td><td>Any 2 out of 4</td><td>10</td></tr>
<tr><td>Q.3</td><td>Module 1 & 2</td><td>Any 2 out of 4</td><td>10</td></tr>
</table></td></tr>
</table>

Name of the Course: CS Practical V - Artificial Intelligence and Information Security Lab

Sr. No.	Heading	Particulars
1	**Description the course:**	This practical course in Artificial Intelligence (AI) and Information Security provides hands-on experience with key AI techniques, algorithms, and security concepts using Python. Students will explore foundational topics such as search strategies, knowledge representation, fuzzy logic, machine learning, optimization techniques, and essential information security practices. Through practical exercises, real-world problem-solving, and the development of intelligent and secure systems, learners will gain a strong understanding of how AI and security methods are applied in practical scenarios.
2	**Vertical:**	Major
3	**Type:**	Practical
4	**Credits:**	2 credits (1 credit = 30 Hours of Practical work in a semester)
5	**Hours Allotted:**	60 hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives:** • To introduce the fundamental concepts of Artificial Intelligence and Information Security. • To develop practical skills in implementing AI techniques and algorithms using Python. • To provide hands-on experience in solving problems using machine learning, search strategies, and optimization methods. • To familiarize students with knowledge representation, fuzzy logic, and basic cybersecurity practices. • To enhance analytical thinking and the application of AI and security concepts in real-world scenarios.	
8	**Course Outcomes:** After successful completion of this course, learners will be able to: CO1. Implement basic Artificial Intelligence algorithms and techniques using Python. CO2. Apply machine learning, search strategies, and optimization methods to practical problems. CO3. Demonstrate understanding of knowledge representation and fuzzy logic concepts. CO4. Identify common information security threats and apply basic security measures. CO5. Develop simple AI-based and secure computing applications for real-world use.	
9	**Modules:-** **Module 1 (30 hours): Artificial Intelligence – Practical**	
	Practical 1: Problem-Solving Agent Simulation • Implement a simple problem-solving agent that can navigate a predefined environment (like a grid world). • Demonstrate agent decisions based on current state and goal.	

	Practical 2: Uninformed Search Implementation ● Implement **Breadth-First Search (BFS)** and **Depth-First Search (DFS)** in Python. ● Test the algorithms on a sample maze or pathfinding problem. **Practical 3: Informed (Heuristic) Search Implementation** ● Implement **A*** or **Greedy Best-First Search** algorithm. ● Apply it to a pathfinding problem and compare with BFS/DFS. **Practical 4: Knowledge Representation using Logic** ● Represent simple facts and rules using **propositional logic**. ● Implement **forward and backward chaining** to deduce new facts. **Practical 5: Fuzzy Logic Application** ● Design a simple **fuzzy logic system** (e.g., temperature control or speed controller). ● Define fuzzy sets, membership functions, and rules. **Practical 6: Classification using Decision Trees** ● Implement a **decision tree classifier** on a sample dataset (e.g., Iris dataset). ● Visualize the decision tree and evaluate accuracy. **Practical 7: Support Vector Machine (SVM) Classification** ● Implement **SVM** for classification tasks. ● Compare results with decision tree classifier. **Practical 8: K-Nearest Neighbors (K-NN) Classification** ● Implement **K-NN algorithm** on a dataset. ● Experiment with different values of K and measure accuracy. **Practical 9: Gradient Descent for Linear Regression** ● Implement **linear regression** using gradient descent. ● Predict outputs for a sample dataset and plot results. **Practical 10: Mini Project: AI Application** ● Combine techniques learned: ○ Example: Maze-solving agent using A* + knowledge representation ○ Example: Classification project using decision tree or SVM ● Prepare a report with methodology, results, and conclusions **Preferred Software for practicals** : **Python 3.x** – Primary programming language for AI. **Recommended IDEs:** ● **Jupyter Notebook** – Great for step-by-step coding and visualization. ● **VS Code** – Lightweight, good for larger scripts.
	Module 2 (30 hours): Information Security – Practical
	Practical 1: Caesar Cipher (Substitution Cipher) ● Implement **Caesar cipher encryption and decryption** using Python. ● Test with sample messages. **Practical 2: Playfair Cipher (Substitution Cipher)**

	• Implement **Playfair cipher** for encryption and decryption. • Use a keyword to generate the 5×5 key matrix. **Practical 3: Monoalphabetic & Polyalphabetic Ciphers** • Implement **monoalphabetic substitution** and **Vigenère cipher (polyalphabetic)**. • Compare security with Caesar cipher. **Practical 4: Transposition Ciphers** • Implement **Rail Fence Cipher** and **Columnar Transposition Cipher**. • Encrypt and decrypt sample messages. **Practical 5: DES and AES Encryption (Symmetric Block Ciphers)** • Encrypt/decrypt text using **DES** and **AES** using Python libraries (PyCryptodome or cryptography). • Explore **CBC/ECB modes**. **Practical 6: RSA Encryption and Decryption (Asymmetric Cipher)** • Implement **RSA algorithm** for encryption/decryption. • Generate public/private keys and test with messages. **Practical 7: Diffie-Hellman Key Exchange** • Implement **Diffie-Hellman key exchange** to securely share a secret key between two parties over an insecure channel. **Practical 8: Message Authentication Codes (MACs)** • Implement MAC generation and verification using Python. • Use HMAC with SHA algorithms to ensure message integrity. **Practical 9: Digital Signatures (RSA-based)** • Implement **RSA-based digital signatures**. • Sign a message digitally and verify its authenticity and integrity. **Practical 10: Mini Project: Secure Communication Simulation** • Combine techniques learned: ○ Symmetric encryption for message content (AES/DES) ○ Asymmetric encryption for key exchange (RSA/Diffie-Hellman) ○ MAC or digital signature for authentication • Simulate a secure message transfer between two entities. **Preferred Software for practicals** : JDK1,8 or above, Netbeans, Jupyter Notebook, VS Code	
10	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
11	**Continuous Evaluation through:** Practical journal submission, viva, assignments	**A semester end practical examination of 2 hours duration for 30 marks as the paper pattern given below.**

	• Journal Submission : 10 Marks • Assignments: 10 Marks • Total: 20 marks	*Its compulsory to carry certified journal at the time of practical exam*
12	Format of Question Paper: (Semester End Practical Examination: 30 Marks. Duration:2 hours) Q1: Module 1 (12 marks) Q2: Module 2 (12 marks) Q.3 Viva (06 Marks)	

Name of the Course : Mini Project - I

Sr. No.	Heading	Particulars
1	**Description the course:**	Mini Project – I is a structured, development-oriented course that enables students to design and implement a complete real-world software solution. The course guides students through problem identification, requirement engineering, system modeling, architecture design, implementation, testing, and deployment. It emphasizes professional documentation, structured development practices, and industry-standard tools.
2	**Vertical:**	Major
3	**Type:**	Practical
4	**Credits:**	2 credits (1 credit = 30 Hours of Practical work in a semester)
5	**Hours Allotted:**	60 hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives:** 1. To enable students to identify and analyze real-world problems and conduct feasibility studies. 2. To develop skills in requirement engineering, UML-based system modeling, and architecture design. 3. To provide hands-on experience in application development, testing, and deployment. 4. To inculcate best practices in version control, documentation, and project management. 5. To prepare students for professional software	
8	**Course Outcomes:** After successful completion of this course, students would be able to - CO1. Identify and analyse a real-world problem and prepare structured project documentation including SRS and feasibility reports. CO2. Design complete system models using UML diagrams and develop suitable frontend, backend, and database architectures. CO3. Implement and integrate application components with authentication, validation, and error handling mechanisms. CO4. Test and deploy applications using systematic testing approaches and version control tools such as GitHub. CO5. Evaluate and present the developed solution through technical documentation, demonstration, and performance/security validation.	
9	**Modules:-** **Module 1 (30 hours)**	
	Problem Identification, Requirement Engineering & System Design Phase ○ Problem Identification & Feasibility Study: Identification of a real-world problem (industry / social / institutional), Problem justification and scope definition, ○ Stakeholder identification, Feasibility analysis (technical, economic, operational) ○ Requirement Engineering: Functional requirements specification, Non-functional requirements, Use-case analysis, Requirement prioritization, Constraints and assumptions ○ Software Development Life Cycle (SDLC) Planning: Selection of SDLC model, Work Breakdown Structure (WBS), Project timeline (Gantt Chart), Resource planning ○ System Modeling using UML: Use Case Diagram, Class Diagram, Sequence Diagram, Activity Diagram, ER Diagram, Deployment Diagram	

	○ System Architecture Design: Frontend architecture, Backend architecture, Database schema design, API structure, Security considerations • Documentation Deliverables at the End of Module 1 ○ Project Proposal ○ SRS Document ○ Complete UML Set ○ Architecture Design Document	
	Module 2 (30 hours)	
	Implementation, Testing, Deployment & Evaluation Phase • Application Development: Frontend implementation, Backend implementation, Database integration, Authentication & validation, Error handling • Integration & System Testing: Unit testing, Black-box testing, Integration testing, Test case preparation, Bug tracking • Deployment: Cloud deployment / Local hosting, APK build, Server configuration (if applicable), Version control using GitHub (Mandatory) • Performance & Security Testing: Basic load testing, Input validation checks, Security validation • Final Documentation: Technical Report, User Manual, Screenshots, Source Code Documentation Final Deliverables at the End of Module 2 o Working Application o GitHub Repository o Final Report o Presentation & Demonstration	
10	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
	Internal Evaluation : (To be assessed by the Project Guide) **Components :** • Problem Identification & Project Proposal – 5 Marks • System Design (SRS, UML Diagrams, Architecture) – 5 Marks • Development Progress & Code Review– 5 Marks • Internal Presentation / Review– 5 Marks	
	External Evaluation : (To be assessed by External Examiner) Components : • Working Application Demonstration -10 Marks • Technical Design & Implementation - 10 Marks • Project Report Evaluation – 5 Marks • Viva Voce – 5 Marks	

Elective Courses

Name of the Course: Software Testing and Quality Engineering

Sr. No.	Heading	Particulars
1	**Description of the course:**	**Introduction:** This course provides an in-depth understanding of software testing principles, techniques, and quality assurance practices. Students will learn the concepts and methodologies required to ensure the quality and reliability of software systems. The course covers various aspects of software testing, including test case design, verification and validation, defect management, software quality assurance, and quality improvement techniques. Students will gain hands-on experience with industry-standard tools and techniques used in software testing and quality assurance.
2	**Vertical:**	**Elective**
3	**Type:**	**Theory**
4	**Credits:**	**2 credits**
5	**Hours Allotted:**	**30 Hours**
6	**Marks Allotted:**	**50 Marks**
7	**Course Objectives:** 1. Understand the significance of software testing in ensuring software quality and reliability. 2. Familiarize with the fundamental concepts and principles of software quality assurance. 3. Learn different software testing techniques and methodologies for effective test case design. 4. Explore the role of verification and validation in software development and testing processes. 5. Gain practical experience in using software testing tools and frameworks for automated testing	
8	**Course Outcomes:** **CO1.** Explain the significance of software testing and its role in ensuring software quality. **CO2.** Apply suitable testing techniques to detect, analyze, and reduce software defects. **CO3.** Design and execute effective test cases to validate functionality, reliability, and performance of software systems. **CO4.** Apply verification and validation principles with automated testing tools and frameworks to ensure software quality, efficiency, and effectiveness.	
9	**Modules: -** **Module 1 (15 hours):**	
	Introduction to Software Testing a. Nature of errors, need for testing b. Quality concepts: Quality, Quality Assurance (QA), Quality Control (QC), Quality Management (QM), Software Quality Assurance (SQA) 2. **SDLC and Testing** a. Phases of SDLC and testing role in each phase b. Software quality factors and impact on testing	

3. **Verification & Validation (V&V)**
    a. Definition, significance, mechanisms
    b. Reviews, inspections, walkthroughs
4. **Testing Techniques**
    a. Test process fundamentals, test case design principles
    b. Black Box Testing: Equivalence Partitioning, Boundary Value Analysis, Decision Table Testing, State Transition Testing
    c. White Box Testing: Statement Testing, Branch Testing
    d. Experience-based Testing: Error guessing, Exploratory, Checklist-based

Module 2 (15 hours):

Software Testing Strategies

a. Unit Testing: purpose, techniques, best practices
b. Integration Testing: approaches, challenges
c. Validation & System Testing: adherence to requirements, end-to-end testing
d. Software Metrics: concepts, types, complexity metrics

2. **Defect Management**
    a. Definition, lifecycle, reporting & tracking
    b. Defect metrics for process improvement
3. **Software Quality Assurance**
    a. SQA concepts, activities, and approaches
    b. Formal Technical Reviews, Statistical Quality Assurance, Software Reliability
    c. ISO 9000 Quality Standards in software development
4. **Quality Improvement Techniques**
    a. Quality costs in decision-making
    b. Tools: Pareto Diagram, Cause-effect Diagram, Scatter Diagram, Run Charts

10 **Reference Books**

- Quality Management, Donna C. S. Summers, 5th Edition, Prentice-Hall.
- Software Testing and Quality Assurance Theory and Practice, Kshirsagar Naik, Priyadarshi Tripathy , John Wiley & Sons, Inc. , Publication.

11	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
12	**Continuous Evaluation through:** Class test of 15 marks Quizzes/ Presentations/ Assignments: 5 marks Total: 20 marks	**Format of Question Paper: External Examination (30 Marks)– 1 hr duration**

13 **Format of Question Paper: (Semester End Examination: 30 Marks. Duration:1 hour)**

Questions	Based On	Options	Marks
Q.1	Module 1	Any 2 out of 4	10
Q.2	Module 2	Any 2 out of 4	10
Q.3	Module 1 & 2	Any 2 out of 4	10

Name of the Course: Wireless Sensor Network

Sr. No.	Heading	Particulars
1	**Description of the course:**	**Introduction:** This course provides a comprehensive understanding of Wireless Sensor Networks (WSNs) and their applications. It covers the fundamental concepts, architectural elements, advantages, and challenges of WSNs. Students will explore sensor node technology, network architecture, optimization goals, and design principles for WSNs. The course also delves into wireless transmission, telecommunication systems, and introduces the concepts of WSN operating systems and ad-hoc networks. Through practical examples and case studies, students will gain hands-on experience in medium access control protocols, routing strategies, transport control protocols, and WSN middleware architecture.
2	**Vertical:**	**Elective**
3	**Type:**	**Theory**
4	**Credits:**	**2 credits (1 credit = 15 Hours for Theory)**
5	**Hours Allotted:**	**30 Hours**
6	**Marks Allotted:**	**50 Marks**
7	**Course Objectives:** ● Provide students with a comprehensive understanding of Wireless Sensor Networks (WSNs), including their basic architectural elements, advantages, and challenges. ● Introduce students to the key technologies and protocols used in WSNs, such as medium access control (MAC) protocols, routing strategies, and transport control protocols. ● Familiarize students with wireless transmission principles and telecommunication systems relevant to WSNs, including frequency, signals, antennas, and satellite systems. ● Develop students' practical skills in designing and implementing WSN solutions by exploring WSN operating systems, ad-hoc networks, and optimization goals.	
8	**Course Outcomes:** ● Understand WSN fundamentals, architecture, components, and operating environments. ● Explain WSN operating systems, ad-hoc networks, and energy efficiency aspects. ● Analyze MAC protocols, routing strategies, and transport mechanisms in WSNs. ● Apply wireless transmission principles and describe telecom, cellular, satellite, and broadcast systems (GSM, DECT, UMTS, etc.).	
9	**Modules** **Module 1 (15 hours):**	
	1. Introduction to WSN ● Basic sensor network architecture and elements ● Advantages, challenges, and applications ● Sensor node technology and taxonomy ● WSN operating environment, radio technology, network architecture ● Optimization goals, figures of merit, and design principles ● Service interfaces and gateway concepts	

2. WSN Operating Systems & Ad-hoc Networks

- Overview and examples of WSN Operating Systems
- Ad-hoc networks: characteristics, challenges, energy efficiency
- Security and privacy considerations in ad-hoc networks

3. MAC and Routing in WSN

1. Fundamentals of Medium Access Control (MAC) protocols
2. Case study: Sensor-MAC
3. Routing challenges and design issues
4. IEEE 802.15.4 LR-WPANs standard
5. Routing strategies in WSNs
6. Transport control protocols and design issues
7. WSN Middleware architecture

Module 2 (15 hours):

1. Wireless Transmission Fundamentals

- Frequency for radio transmission, signal propagation, and antennas
- Multiplexing, modulation, spread spectrum
- Cellular systems basics

2. Telecommunication, Satellite, and Broadcast Systems

CO5. GSM: Mobile services, system architecture, radio interface, protocols, localization, handover, security, new data services
CO6. DECT: System and protocol architecture
CO7. ETRA, UMTS, and IMT-2000 overview
CO8. Satellite systems: GEO, LEO, MEO, routing, localization, handover
CO9. Broadcast systems basics and applications

10 Reference Books

- Fundamentals of Wireless Sensor Networks, Theory and Practice, Waltenegus Dargie, Christian Poellabauer , Wiley Series on wireless Communication and Mobile Computing, 2011
- Networking Wireless Sensors, Bhaskar Krishnamachari , Cambridge University Press, 2005

11	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
12	**Continuous Evaluation through:** Class test of 15 marks Quizzes/ Presentations/ Assignments: 5 marks Total: 20 marks	**Format of Question Paper: External Examination (30 Marks)– 1 hr duration**

13 Format of Question Paper: (Semester End Examination: 30 Marks. Duration:1 hour)

Questions	Based On	Options	Marks
Q.1	Module 1	Any 2 out of 4	10
Q.2	Module 2	Any 2 out of 4	10
Q.3	Module 1 & 2	Any 2 out of 4	10

Name of the Course: Robotics Process Automation

Sr. No.	Heading	Particulars
1	**Description of the course:**	**Introduction:** This course provides a comprehensive introduction to Robotic Process Automation (RPA) using UiPath, focusing on the development of intelligent automation solutions for business and industry applications. Students will gain practical knowledge of UiPath Studio, Orchestrator, and Robot components while learning to design, develop, and deploy automation workflows efficiently. The course covers essential topics such as workflow design, data manipulation, screen and web scraping, PDF and Excel automation, exception handling, email automation, and bot deployment. Learners will also explore advanced automation techniques, debugging methods, queue management, and real-world case studies across domains such as banking, healthcare, logistics, education, and accounting. Through hands-on practice and project-based learning, students will develop the skills required to automate repetitive tasks, improve operational efficiency, and implement reliable RPA solutions in modern organizations.
2	**Vertical:**	Elective
3	**Type:**	Theory
4	**Credits:**	2 credits
5	**Hours Allotted:**	30 hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives:** 1. UiPath Fundamentals: Learn UiPath RPA basics, including Studio and Orchestrator. 2. Workflow Design: Develop automation workflows efficiently with UiPath Studio. 3. Data Manipulation: Extract, transform, and integrate data using UiPath. 4. Error Management: Handle exceptions and ensure reliable automation. 5. Orchestrator Management: Configure and monitor Orchestrator for robot management.	
8	**Course Outcomes:** 1. Understand WSN fundamentals, architecture, design principles, and applications. 2. Explore sensor node technology, wireless transmission, and telecommunication systems. 3. Examine WSN operating systems, ad-hoc networks, and middleware architecture. 4. Gain hands-on experience with MAC protocols, routing, and transport control through case studies.	
9	**Modules: -** **Module 1 (15 hours)**	
	1. Introduction to RPA ○ Definition, history, benefits, and use cases ○ RPA tools overview: UiPath, Automation Anywhere, Blue Prism	

<table>
<tr><td></td><td>

RPA lifecycle and components (Studio, Orchestrator, Robot)

2. Workflow Design in UiPath

Sequences and Flowcharts
Variables and Data Types: Number, String, Boolean, Array, Data Table, Generic
Decision making and loops: If, Switch, For Each, Do While, Delay activities
Recording techniques: Basic, Desktop, Web

3. Data Handling Basics

Excel automation: Reading, writing, appending, sorting, filtering
Screen and web scraping: Full Text, Native, OCR methods
PDF automation: Reading, merging, extracting data

</td></tr>
<tr><td></td><td>Module 2 (15 hours)</td></tr>
<tr><td></td><td>

Advanced Automation Techniques

Controls and event handling: Attach Window, Anchor Base, Element Exists, Find Children
Keyboard and mouse automation: Click, Double-click, Hover, Type Into, Type Secure Text
Exception handling and debugging: Try Catch, Retry Scope, Logging

Email and Process Automation

Email configuration, reading, sending, replying, and saving attachments
Automating repetitive tasks and real-world applications: Banking, Healthcare, Logistics, Education, Accounting

Orchestrator and Deployment

Deploy bots and run jobs in Orchestrator
Queue management and job scheduling
Case studies of RPA implementation in different domains

</td></tr>
<tr><td>11</td><td>Internal Continuous Assessment: 40%</td><td>Semester End Examination: 60%</td></tr>
<tr><td>12</td><td>Continuous Evaluation through:
Class test of 15 marks

Quizzes/ Presentations/ Assignments: 5 marks
Total: 20 marks</td><td>Format of Question Paper: External Examination (30 Marks)– 1 hr duration</td></tr>
<tr><td>13</td><td colspan="2">Format of Question Paper: (Semester End Examination: 30 Marks. Duration:1 hour)
<table>
<tr><th>Questions</th><th>Based On</th><th>Options</th><th>Marks</th></tr>
<tr><td>Q.1</td><td>Module 1</td><td>Any 2 out of 4</td><td>10</td></tr>
<tr><td>Q.2</td><td>Module 2</td><td>Any 2 out of 4</td><td>10</td></tr>
<tr><td>Q.3</td><td>Module 1 & 2</td><td>Any 2 out of 4</td><td>10</td></tr>
</table>
</td></tr>
</table>

Name of the Course: Software Testing and Quality Engineering - *LAB*

Sr. No.	Heading	Particulars
1	**Description the course:**	**Introduction:** This course introduces students to software test automation using Selenium IDE, WebDriver, and Selenium RC. It covers test suite creation, data-driven testing, and automation of web interactions such as login, form handling, and object identification. Students will also gain hands-on experience with performance testing using JMeter and defect tracking through Bugzilla, enabling them to build efficient and reliable automated testing solutions.
2	**Vertical:**	Elective
3	**Type:**	Practical
4	**Credits:**	2 credits (1 credit = 15 Hours of Practical work in a semester)
5	**Hours Allotted:**	30 hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives:** 1. Understand Selenium IDE, WebDriver, and RC for web automation. 2. Design and execute automated test suites across different web formats and websites. 3. Apply data-driven testing and automate UI element handling. 4. Perform load testing with JMeter and defect tracking with Bugzilla.	
8	**Course Outcomes:** CO1. Install, configure, and execute automated test suites using Selenium IDE and WebDriver. CO2. Automate login, form filling, object identification, and UI element handling in web applications. CO3. Implement data-driven testing with Excel for record updates and analysis. CO4: Conduct load testing with JMeter and manage bug tracking using Bugzilla.	
9	**Modules: -** **Module 1 (15 hours):**	
	1. **Selenium IDE Installation & Test Suite** o Install Selenium IDE and create a test suite with at least 4 test cases for different web page formats (HTML, XML, JSON, etc.). ● **Test Suite Execution on Multiple Websites** o Conduct a test suite for two different websites using Selenium IDE, performing actions such as clicking links, filling forms, and verifying content. ● **Selenium Server (RC) Demonstration** o Install Selenium Server (RC) and execute an automation script in Java or PHP to demonstrate browser interaction ● **Login Automation using Selenium WebDriver** o Write a Selenium WebDriver program to automate the login process on a web page and verify successful login with assertions ● **Excel Data Update Automation** o Use Selenium WebDriver to update 10 student records in an Excel file and perform data manipulation with verification.	

	Module 2 (15 hours):	
	6. **Student Marks Analysis** o Automate a script using Selenium WebDriver to identify students scoring more than 60 in one or all subjects, with data extraction and analysis. 7. **Object Identification and Counting** o Write a Selenium WebDriver program to detect and count the total number of objects on a web page. 8. **List/Combo Box Item Count** o Automate counting of items in a list or combo box on a web page using Selenium WebDriver. 9. **Checkbox Handling and Counting** o Write a Selenium WebDriver program to count all checkboxes on a web page, distinguishing between checked and unchecked. 10. **Load Testing & Bug Tracking** ● Perform load testing on a web application using JMeter, generate and analyze load scenarios, and use Bugzilla for logging and tracking software defects. **Preferred Software for practical's**: Selenium IDE, Eclipse	
10	**Reference Books:** 1. Quality Management, Donna C. S. Summers, 5th Edition, Prentice-Hall. 2. Software Testing and Quality Assurance Theory and Practice, Kshirsagar Naik, Priyadarshi Tripathy, John Wiley & Sons, Inc., Publication.	
11	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
12	**Continuous Evaluation through:** Practical journal submission, viva, assignments 3. Journal Submission : 10 Marks 4. Assignments: 10 Marks 5. Total: 20 marks	**A semester end practical examination of 2 hours duration for 30 marks as the paper pattern given below.** *Its compulsory to carry certified journal at the time of practical exam*
13	Format of Question Paper: (Semester End Practical Examination: 30 Marks. Duration:2 hours) Q1: Module 1 (12 marks) Q2: Module 2 (12 marks) Q.3 Viva (06 Marks)	

Name of the Course: Wireless Sensor Network - *LAB*

Sr. No.	Heading	Particulars
1	**Description of the course:**	**Introduction:** This course provides hands-on exploration of Wireless Sensor Networks (WSNs) through hardware and software simulations. Students will study sensor node components, TinyOS concepts, and nesC programming. Using tools like TOSSIM and NS2/NS3, they will simulate communication, routing, and MAC protocols in WSNs, as well as advanced scenarios such as MANETs, directional antennas, and cellular networks. The course emphasizes both foundational understanding and practical implementation of wireless networking principles.
2	**Vertical:**	Elective
3	**Type:**	Practical
4	**Credits:**	2 credits (1 credit = 15 Hours of Practical work in a semester)
5	**Hours Allotted:**	30 hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives:** 1. Introduce the hardware and software components of sensor nodes and their programming models. 2. Provide practical exposure to TinyOS, nesC, and simulation tools (TOSSIM, NS2/NS3). 3. Enable students to simulate and analyze WSN communication, routing, and MAC protocols. 4. Explore advanced wireless simulations including MANETs, directional antennas, and cellular networks.	
8	**Course Outcomes:** CO1. Identify and analyze sensor node hardware, TinyOS concepts, and nesC programming components. CO2. Simulate mote-to-mote, mote-to-PC, and ad-hoc network communication using TOSSIM. CO3. Implement and analyze routing, MAC protocols, and WSN communication through simulations. CO4. Design and simulate advanced wireless networks such as MANETs, directional antenna systems, and cellular networks.	
9	**Modules: -** **Module 1 (15 hours):**	
	● **Sensor Node Hardware Exploration** ○ Understand the components of a sensor node (sensor, mote, base station) and its graphical user interface. ● **TinyOS Concepts** ○ Explore TinyOS computational concepts: Events, Commands, Tasks. ○ Understand the nesC programming model and nesC components.	

	• **TOSSIM Simulation Basics** ○ Simulate mote-to-mote radio communication. ○ Simulate mote-to-PC serial communication. • **Ad-hoc Network Simulation** ○ Create and simulate a simple ad-hoc network. • **Routing Table Analysis** ○ Understand, read, and analyze the routing table of a WSN.	
	Module 2 (15 hours):	
	• **MANET Simulation** ○ Implement a basic Mobile Ad-hoc Network (MANET) and visualize packet animation and trace. • **Wireless Sensor Network Simulation** ○ Simulate a basic WSN with multiple nodes and base station communication. • **MAC Protocol Simulation** ○ Implement and simulate a MAC protocol for a WSN. • **Directional Antenna Simulation** ○ Simulate a mobile ad-hoc network using directional antennas. • **Cellular Network Simulation** ○ Create a mobile network using a cell tower, central office server, web browser, and web server. ○ Simulate connections and data communication between components.	
10	**Reference Books**	
11	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
12	**Continuous Evaluation through:** Practical journal submission, viva, assignments 5. Journal Submission : 10 Marks 6. Assignments: 10 Marks 7. Total: 20 marks	**A semester end practical examination of 2 hours duration for 30 marks as the paper pattern given below.** *Its compulsory to carry certified journal at the time of practical exam*
13	Format of Question Paper: (Semester End Practical Examination: 30 Marks. Duration:2 hours) Q1: Module 1 (12 marks) Q2: Module 2 (12 marks) Q.3 Viva (06 Marks)	

Name of the Course: Robotics Process Automation - *LAB*

Sr. No.	Heading	Particulars
1	**Description the course:**	**Introduction:** This course introduces Robotic Process Automation (RPA) using UiPath, focusing on automation of repetitive business tasks through sequences, flowcharts, and advanced activities. Students will gain hands-on experience with decision-making, looping, recording techniques, Excel and PDF automation, email automation, and exception handling. The course also covers advanced topics like screen/web scraping, bot deployment, and real-world mini projects, preparing students to design and implement RPA solutions across various domains.
2	**Vertical:**	Elective
3	**Type:**	Practical
4	**Credits:**	2 credits (1 credit = 15 Hours of Practical work in a semester)
5	**Hours Allotted:**	30 hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives:** 1. Understand the fundamentals of RPA and UiPath workflows using sequences and flowcharts. 2. Develop automation solutions for decision-making, looping, and data handling in Excel, PDFs, and emails. 3. Gain skills in advanced automation techniques such as web scraping, exception handling, and event-driven bots. 4. Apply RPA concepts in real-world domains through mini projects and bot deployment on Orchestrator.	
8	**Course Outcomes:** CO1. Design and implement automation workflows using sequences, flowcharts, and control activities in UiPath. CO2. Automate business tasks involving Excel, PDFs, web scraping, and email handling. CO3. Apply decision-making, looping, recording, and exception handling to create reliable automation solutions. CO4. Deploy, manage, and demonstrate RPA bots in practical domains such as banking, healthcare, logistics, or education.	
9	**Modules: -** **Module 1 (15 hours):**	
	1. **Sequences and Flowcharts Automation** a. Create a sequence-based and a flowchart-based project. b. Automate basic number operations (subtraction, multiplication, division) using variables of different types (number, datetime, Boolean, generic, array, data table).	

	2. **Decision Making and Looping** a. Count names starting with a specific letter in an array. b. Demonstrate switch, if, Do..While, and For Each activity. c. Use Delay activity to separate actions. 3. **Recording Techniques** a. Perform Basic Recording using Toolbar and Notepad. b. Perform Desktop Recording and Web Recording (e.g., extract movie rating from IMDb). 4. **Excel Automation** a. Read, write, append, sort, and filter data in Excel using UiPath. b. Extract data from Excel into DataTables and vice versa. 5. **Controls and Element Handling** a. Implement Attach Window, Anchor Base, Element Exists, Find Children, Find Relative, and Get Ancestor activities.
	Module 2 (15 hours):
	6. **Keyboard and Mouse Events Automation** a. Demonstrate Mouse (click, double-click, hover), Type Into, and Type Secure Text. b. Use Element, Image, and System triggering events. c. Automate launching a bot on a keyboard event. 7. **Screen Scraping and Web Scraping** a. Implement Full Text, Native, and OCR screen scraping. b. Extract data from web pages, PDFs, and images; display in Message Box. 8. **PDF Automation and Exception Handling** a. Read PDF using OCR, merge PDFs, extract page count using Regex. b. Extract data from PDFs or Excel and populate databases/spreadsheets with filtering, sorting, and validation. c. Demonstrate Exception Handling in UiPath. 9. **Email Automation** a. Configure and automate reading, sending (with attachment), saving, and replying to emails. 10. **Mini Project / RPA Applications** a. Deploy bots to Orchestrator, run jobs, manage queues. b. Implement a project automating tasks in any domain (e.g., Banking, Healthcare, Logistics, Education, Accounting). c. Examples: invoice data extraction, expense reporting, inventory management, sales order processing, data entry automation **Preferred Software for practical's : UI Path**
10	**Reference Books** 1. Learning Robotic Process Automation — Alok Mani Tripathi, Packt Publishing. 2. Robotic Process Automation using UiPath StudioX — Nandan Mullakara and Lakshmi Ramakrishnan, Packt Publishing. 3. Hands-On Intelligent Agents with OpenAI Gym — Praveen Palanisamy, Packt Publishing.

	4. Robotic Process Automation: Guide to Building Software Robots, Automate Repetitive Tasks & Become an RPA Consultant — Srikanth Merianda. 5. Intelligent Automation with Blue Prism — Colin Dowsett, Packt Publishing.	
11	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
12	**Continuous Evaluation through:** Practical journal submission, viva, assignments 11. Journal Submission: 10 Marks 12. Assignments: 10 Marks 13. Total: 20 marks	**A semester end practical examination of 2 hours duration for 30 marks as the paper pattern given below.** *Its compulsory to carry certified journal at the time of practical exam*
13	Format of Question Paper: (Semester End Practical Examination: 30 Marks. Duration: 2 hours) Q1: Module 1 (12 marks) Q2: Module 2 (12 marks) Q.3 Viva (06 Marks)	

Vertical – II

Name of the Course: DS-Deep Learning

Sr. No.	Heading	Particulars
1	**Description the course:**	**Introduction:** This course provides a comprehensive overview of Deep Learning, covering the fundamentals of Artificial Neural Networks (ANNs), Deep Neural Networks (DNNs), and their applications in computer vision, NLP, healthcare, and robotics. Students will learn key architectures like CNNs, RNNs, autoencoders, and GANs, along with optimization, regularization, and practical model deployment considerations.
2	**Vertical:**	**Minor**
3	**Type:**	**Theory**
4	**Credits:**	**2 credits**
5	**Hours Allotted:**	**30 Hours**
6	**Marks Allotted:**	**50 Marks**
7	**Course Objectives:** 1. Understand the principles, architectures, and applications of Deep Learning and Artificial Neural Networks. 2. Explore key network types such as CNNs, RNNs, autoencoders, and generative models. 3. Learn optimization, regularization, and evaluation techniques for building effective deep learning models. 4. Gain practical skills in using popular deep learning frameworks (TensorFlow, Keras, PyTorch) for model development and deployment.	
8	**Course Outcomes:** CO1. Explain the concepts of Deep Learning, ANNs, and DNNs, differentiating them from traditional Machine Learning. CO2. Design and implement CNNs, RNNs, and autoencoder-based models for real-world tasks. CO3. Apply optimization, regularization, and evaluation techniques to improve model performance. CO4. Develop, train, and deploy deep learning models using frameworks like TensorFlow, Keras, and PyTorch.	
9	**Modules: -** **Module 1 (15 hours):**	
	1. Introduction to Deep Learning ● Definition and scope of Deep Learning ● Difference between Machine Learning and Deep Learning ● Applications of Deep Learning in computer vision, NLP, healthcare, robotics, etc. **2. Artificial Neural Networks (ANNs)** 1. Biological inspiration and structure of neurons	

	2. Architecture of ANN: Input, hidden, output layers 3. Activation functions: Sigmoid, ReLU, Tanh, Softmax 4. Forward propagation and backpropagation concepts **3. Deep Neural Networks (DNNs)** ● Concept of depth in networks ● Vanishing and exploding gradient problem ● Initialization, optimization, and regularization techniques **4. Deep Learning Frameworks** ● Overview of popular frameworks: TensorFlow, Keras, PyTorch ● Basic workflow for building and training models	
	Module 2 (15 hours):	
	1. Convolutional Neural Networks (CNNs) ● Architecture: Convolution, Pooling, Fully connected layers ● Applications in image classification, object detection, and recognition **2. Recurrent Neural Networks (RNNs) and Variants** ● Introduction to RNNs, LSTM, GRU ● Applications in time series analysis, speech recognition, and NLP **3. Autoencoders and Generative Models** ● Autoencoders: Structure, encoding-decoding, applications ● Introduction to Generative Adversarial Networks (GANs) **4. Deep Learning Optimization and Regularization** ● Loss functions, learning rate, gradient descent variants ● Dropout, batch normalization, early stopping	
10	**Reference Books** 1. **Deep Learning**, Charu C. Aggarwal, Springer. 2. **Fundamentals of Artificial Intelligence and Machine Learning**, Suresh Samudrala, Notion Press. 3. **Machine Learning and Deep Learning with Python**, Vinita Silaparasetty, BPB Publications. 4. **Artificial Intelligence with Python**, Prateek Joshi, Packt Publishing. 5. **Neural Networks and Deep Learning: A Textbook**, Shivendra Nandan, BPB Publications.	
11	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
12	**Continuous Evaluation through:** Class test of 15 marks	**Format of Question Paper: External Examination (30 Marks)– 1 hr duration**

	Quizzes/ Presentations/ Assignments: 5 marks Total: 20 marks	

13 **Format of Question Paper: (Semester End Examination: 30 Marks. Duration:1 hour)**

Questions	Based On	Options	Marks
Q.1	Module 1	Any 2 out of 4	10
Q.2	Module 2	Any 2 out of 4	10
Q.3	Module 1 & 2	Any 2 out of 4	10

Name of the Course: DS-Deep Learning - LAB

Sr. No.	Heading	Particulars
1	**Description the course:**	**Introduction:** This course provides hands-on experience in Deep Learning using frameworks like TensorFlow, Keras, and PyTorch. Students will implement feedforward neural networks, CNNs, RNNs, LSTMs, GRUs, and autoencoders. The course emphasizes practical skills in model training, backpropagation, regularization, data augmentation, evaluation, and hyperparameter tuning for real-world datasets.
2	**Vertical:**	**Minor**
3	**Type:**	**Theory**
4	**Credits:**	**2 credits (1 credit = 15 Hours for Theory)**
5	**Hours Allotted:**	**30 Hours**
6	**Marks Allotted:**	**50 Marks**
7	**Course Objectives:** 1. Install deep learning frameworks and implement basic feedforward neural networks. 2. Explore activation functions, forward and backward propagation, and DNN training techniques. 3. Develop CNN, RNN, LSTM, GRU, and autoencoder models with regularization and data augmentation. 4. Evaluate model performance and optimize hyperparameters for improved accuracy and efficiency.	
8	**Course Outcomes:** CO1. Build and train basic and deep neural networks on standard datasets using TensorFlow, Keras, or PyTorch. CO2. Apply different activation functions, backpropagation, and network depth techniques to optimize learning. CO3. Implement CNNs, RNNs, LSTMs, GRUs, and autoencoders for image and sequence data tasks. CO4. Evaluate models using metrics and tune hyperparameters to enhance performance and generalization.	
9	**Modules** **Module 1 (15 hours):**	
	1. **Setup and Basic Neural Network** ○ Install TensorFlow/Keras or PyTorch. ○ Create a simple feedforward neural network for a small dataset (e.g., Iris or MNIST subset). 2. **Activation Functions and Forward Propagation** ○ Implement and compare different activation functions (Sigmoid, ReLU, Tanh) in a simple neural network. 3. **Backpropagation Demonstration** ○ Train a small neural network and visualize weight updates using backpropagation.	

4. **Deep Neural Network (DNN) Training**
    - Build a multi-layer deep neural network.
    - Observe effects of network depth on training and accuracy.
5. **Convolutional Neural Network (CNN) for Image Classification**
    - Build a CNN model using a dataset like MNIST or CIFAR-10.
    - Train, evaluate, and visualize feature maps.

Module 2 (15 hours):

6. **Image Data Augmentation and Regularization**
    - Apply data augmentation techniques.
    - Implement dropout and batch normalization in a CNN.
7. **Recurrent Neural Network (RNN) for Sequence Data**
    - Implement a simple RNN for time series prediction or sequence classification.
8. **LSTM and GRU Networks**
    - Compare LSTM and GRU models on a sequence prediction dataset.
9. **Autoencoder for Dimensionality Reduction**
    - Build an autoencoder to compress and reconstruct image data.
    - Visualize reconstructed outputs.
10. **Model Evaluation and Hyperparameter Tuning**
    - Evaluate models using metrics like accuracy, precision, recall, F1-score.
    - Experiment with learning rate, batch size, and optimizer variations.

Preferred Software: PyTorch

11 Reference Books

1. Deep Learning, Charu C. Aggarwal, Springer.
2. Fundamentals of Artificial Intelligence and Machine Learning, Suresh Samudrala, Notion Press.
3. Neural Networks and Deep Learning: A Textbook, Shivendra Nandan, BPB Publications.

12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
13	**Continuous Evaluation through:** Class test of 15 marks Quizzes/ Presentations/ Assignments: 5 marks Total: 20 marks	**Format of Question Paper: External Examination (30 Marks)– 1 hr duration**

14 Format of Question Paper: (Semester End Examination: 30 Marks. Duration:1 hour)

Questions	Based On	Marks
Q.1	Module 1	12
Q.2	Module 2	12
Q.3	Viva	06

Vertical – IV

Name of the Course: Full Stack Web Development - LAB

Sr. No.	Heading	Particulars
1	**Description the course:**	**Introduction:** Full Stack Web Development involves designing, developing, and deploying complete web applications using frontend, backend, and database technologies. The MERN Stack consists of MongoDB, Express.js, React.js, and Node.js, which together provide a powerful JavaScript-based development environment. This course introduces students to modern web application development concepts and hands-on implementation of full-stack solutions. It enables learners to build responsive, data-driven, and scalable web applications using industry-standard open-source technologies.
2	**Vertical:**	**VSC**
3	**Type:**	**Practical**
4	**Credits:**	**2 credits**
5	**Hours Allotted:**	**30 Hours**
6	**Marks Allotted:**	**50 Marks**
7	**Course Objectives:** 1. To understand the architecture and components of the MERN Stack framework. 2. To develop dynamic and interactive web applications using React.js and Node.js. 3. To implement RESTful APIs and perform database operations using Express.js and MongoDB. 4.To integrate frontend, backend, and database technologies for full-stack application development.	
8	**Course Outcomes:** Upon successful completion of the course, students will be able to: CO1: Design and develop responsive web interfaces using React.js. CO2: Build server-side applications and REST APIs using Node.js and Express.js. CO3: Perform CRUD operations and manage data using MongoDB and Mongoose. CO4: Develop and deploy complete MERN Stack web applications following industry practices.	
9	**Modules: -** **Module 1 (15 hours):**	
	Practical 1: Setting Up MERN Development Environment Aim	

	To understand the basic structure of Full Stack Web Development and configure the MERN Stack development environment using Node.js, npm, VS Code, and MongoDB Atlas for building web applications. Practical 2: Creating a Simple Node.js Server Aim To create a basic web server using Node.js that responds to client requests and displays a message in the browser. Practical 3: Creating First Express Application Aim To develop a simple web application using Express.js framework and understand routing concepts. Practical 4: Express Routing Example Aim To implement multiple routes using Express.js and understand URL-based request handling. Practical 5: Building REST API Using Express Aim To create a simple REST API that returns JSON data and understand API development concepts.
	Module 2 (15 hours):
	Practical 6: Connect MongoDB Atlas and Create a Student Schema Aim To connect a Node.js application with MongoDB Atlas and create a Mongoose schema and model for storing student information. Practical 7: Implement Express Middleware and Route Parameters Aim To understand and implement middleware in Express.js and use route parameters to handle dynamic user requests. Practical 8: Perform Create and Read Operations in MongoDB Using Mongoose Aim To perform Create (Insert) and Read (Retrieve) operations in MongoDB using Mongoose and understand basic database interaction in a MERN application. Practical 9: Simple React Application Setup Aim To create a basic React application using Create React App and display dynamic content using React components. Practical 10: Simple MERN Student Management Application Aim To develop a basic Full Stack MERN application that accepts student details through a web form and stores them in MongoDB database.

	Preferred Software : Node.js & npm/yarn – Free and open-source. Visual Studio Code – Free, open-source IDE. Express.js – Free, open-source framework. MongoDB – Free Community Edition; MongoDB Atlas has a free tier for cloud databases. Postman – Free tier available for API testing (sufficient for lab exercises). Angular CLI / React (Create React App or Vite) – Free, open-source. bcrypt.js, jsonwebtoken (JWT), Multer – Free, open-source Node.js packages. Browsers (Chrome/Edge) – Free. Git & GitHub – Git is free; GitHub offers free public and private repositories. Heroku / Railway / Vercel – Free tiers available for small projects and labs. Swagger – Free version for API documentation. Note: Theory to be covered before practicals	
10	**Reference Books** 1. Full Stack Development with MERN, Shama Hoque, Packt Publishing. 2. Learning React: Modern Patterns for Developing React Apps – Alex Banks and Eve Porcello. 3. MongoDB: The Definitive Guide – Shannon Bradshaw, Kristina Chodorow, and Eoin Brazil. 4. Full Stack React Projects – Shama Hoque.	
11	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
12	**Continuous Evaluation through:** Practical journal submission, viva, assignments 1. Journal Submission: 10 Marks 2. Assignments: 10 Marks 3. Total: 20 marks	**A semester end practical examination of 2 hours duration for 30 marks as the paper pattern given below.** *Its compulsory to carry certified journal at the time of practical exam*
13	Format of Question Paper: (Semester End Practical Examination: 30 Marks. Duration: 2 hours) Q1: Module 1 (12 marks) Q2: Module 2 (12 marks) Q.3 Viva (06 Marks)	

Semester – VI

SEMESTER VI
COURSE STRUCTURE

VERTICAL		COURSE CODE	SUBJECT	CREDIT
Vertical 1	Major		Data Science	2
	Major		Big Data Analytics	2
	Major		Cybersecurity and Threat Management	2
	Major		CS Practical VI - Data Science-& Big Data Analytics -Lab	2
	Major		Mini Project -II	2
	Elective	(Any One)	1. Information Retrieval 2. Ethical Hacking 3. Blockchain Technology & Applications	2
	Elective	(Any One)	1. Information Retrieval - Lab 2. Ethical Hacking - Lab 3. Blockchain Technology & Applications -Lab	2
Vertical 2	Minor		Machine Learning Algorithms	2
	Minor		Machine Learning Algorithms - **Lab**	2
Vertical 4	VSC		****	**
Vertical 6	FP		OJT	4
			Total Credits	**22**

Mandatory Courses

Name of the Course: Data Science

Sr. No.	Heading	Particulars
1	**Description the course:**	This course introduces the fundamentals of Data Science, its scope, and applications in relation to AI, ML, BI, and Data Warehousing. Students will learn to handle different data types, apply preprocessing and exploratory data analysis, and use statistical and visualization techniques to gain insights. The course further covers basic machine learning methods for classification, regression, and clustering, along with model evaluation and optimization, using Python libraries such as Pandas, NumPy, and Scikit-learn.
2	**Vertical:**	**Major**
3	**Type:**	**Theory**
4	**Credits:**	**2 credits**
5	**Hours Allotted:**	**30 Hours**
6	**Marks Allotted:**	**50 Marks**
7	**Course Objectives:** ● To provide foundational knowledge of Data Science concepts, scope, and its relationship with AI, ML, BI, and Data Warehousing/Data Mining. ● To enable students to identify various data types and sources, and apply preprocessing techniques such as cleaning, transformation, encoding, scaling, and feature selection. ● To develop skills in performing exploratory data analysis (EDA) and applying statistical and visualization methods for meaningful data insights. ● To introduce fundamental machine learning techniques for classification, regression, clustering, and dimensionality reduction using standard tools and libraries. ● To impart knowledge of model evaluation techniques, performance metrics, and optimization strategies to build effective and reliable data-driven solutions.	
8	**Course Outcomes:** By the end of this course, students will be able to: 7. **Explain** the scope of Data Science, its applications, and its relation to AI, ML, BI, and Data Warehousing/Data Mining. 8. **Identify and preprocess** different types and sources of data using techniques such as cleaning, transformation, encoding, scaling, and feature selection. 9. **Perform** exploratory data analysis (EDA) and apply basic statistical and visualization methods to derive insights from data. 10. **Implement and evaluate** basic machine learning algorithms for classification, regression, and clustering using standard tools and libraries.	

9	**Modules: -**
	Module I: Foundations of Data Science and Preprocessing 4. **Introduction to Data Science** a. Definition and scope of Data Science b. Applications and domains of Data Science c. Comparison with BI, AI, ML, DW-DM 5. **Data Types and Sources** a. Structured, semi-structured, and unstructured data b. Data sources: databases, files, APIs, web scraping, sensors, social media 6. **Data Preprocessing** a. Data cleaning: handling missing values, outliers, duplicates b. Data transformation: scaling, normalization, encoding categorical variables c. Feature selection and data merging 7. **Data Wrangling and Feature Engineering** a. Reshaping, pivoting, aggregating b. Creating new features, handling time-series data c. Dummification (categorical to binary indicators) d. Feature scaling: standardization and normalization 8. **Tools and Libraries** a. Overview of Pandas, NumPy, Scikit-learn
	Module 2 :
	Module II: Data Analysis and Machine Learning (15 Hours) ● **Exploratory Data Analysis (EDA)** ○ Visualization: histograms, scatter plots, box plots, etc. ○ Descriptive statistics: mean, median, mode, SD ○ Hypothesis testing: t-tests, chi-square, ANOVA ● **Machine Learning Basics** ○ Supervised learning: classification & regression ○ Unsupervised learning: clustering & dimensionality reduction ○ Bias-variance tradeoff, underfitting, overfitting ● **Regression Analysis** ○ Simple & multiple regression ○ Logistic regression for classification ● **Model Evaluation & Selection** ○ Accuracy, precision, recall, F1-score, confusion matrix, ROC curve ○ Cross-validation: k-fold, stratified ○ Hyperparameter tuning ● **Machine Learning Algorithms (Overview)** ○ Decision Trees, Random Forests ○ SVM, K-NN, ANN ○ Ensemble methods: Bagging & Boosting ○ Gradient Descent optimization

<table>
<tr><td>10</td><td colspan="2">Reference Books
1. Data Science and Big Data Analytics – P. N. Murthy & G. Girija, PHI Learning, 1st Edition, 2019.
2. Big Data and Analytics – Seema Acharya & Subhasini Chellappan, Wiley India, 2nd Edition, 2019.
3. Foundations of Data Science Using Python – B. M. Harwani, BPB Publications, 1st Edition, 2020.
4. Statistics for Machine Learning – Pratap Dangeti, Packt Publishing (Indian Reprint), 1st Edition, 2017.
5. Mining of Massive Datasets – Anand Rajaraman, Jure Leskovec, Jeffrey D. Ullman, Cambridge University Press (Indian Edition available), 3rd Edition, 2020.</td></tr>
<tr><td>11</td><td>Internal Continuous Assessment: 40%</td><td>Semester End Examination: 60%</td></tr>
<tr><td>12</td><td>Continuous Evaluation through:
Class test of 15 marks
Quizzes/ Presentations/ Assignments: 5 marks
Total: 20 marks</td><td>Format of Question Paper: External Examination (30 Marks)– 1 hr duration</td></tr>
<tr><td>13</td><td colspan="2">Format of Question Paper: (Semester End Examination: 30 Marks. Duration:1 hour)
<table>
<tr><th>Questions</th><th>Based On</th><th>Options</th><th>Marks</th></tr>
<tr><td>Q.1</td><td>Module 1</td><td>Any 2 out of 4</td><td>10</td></tr>
<tr><td>Q.2</td><td>Module 2</td><td>Any 2 out of 4</td><td>10</td></tr>
<tr><td>Q.3</td><td>Module 1 & 2</td><td>Any 2 out of 4</td><td>10</td></tr>
</table></td></tr>
</table>

Name of the Course: Big Data Analytics

Sr. No.	Heading	Particulars
1	**Description the course:**	This course provides an introduction to Big Data fundamentals, analytics concepts, and the technologies that enable large-scale data processing. Students will explore the evolution, characteristics, and applications of Big Data, along with traditional Business Intelligence comparisons. The course covers analytical methods including clustering, regression, classification, time series, and text analytics, as well as practical exposure to Hadoop, Spark, HDFS, MapReduce, and related tools. Emphasis is placed on building data pipelines, implementing distributed analytics, and understanding the end-to-end data science lifecycle in Big Data environments.
2	**Vertical:**	Major
3	**Type:**	Theory
4	**Credits:**	2 credits (1 credit = 15 Hours for Theory)
5	**Hours Allotted:**	30 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives:** 1. To introduce students to the fundamentals of Big Data, its evolution, characteristics, challenges, and real-world applications. 2. To provide an understanding of the differences between traditional Business Intelligence and Big Data analytics, including the roles and responsibilities in Big Data ecosystems. 3. To familiarize students with analytical methods such as clustering, association rules, regression, classification, time series, and basic text analytics. 4. To develop practical knowledge of Big Data technologies, including Hadoop, Spark, HDFS, MapReduce, and high-level tools like Hive, HBase, Pig, and Spark SQL. 5. To equip students with the skills to design and implement data pipelines, data products, and distributed analytics solutions in a Big Data environment.	
8	**Course Outcomes:** By the end of this course, students will be able to: ● Explain the evolution, characteristics, and challenges of Big Data, and compare it with traditional Business Intelligence and Data Warehousing approaches. ● Apply basic analytical methods such as clustering, regression, classification, and text analytics to derive insights from large datasets. ● Demonstrate the use of Big Data technologies and frameworks such as Hadoop, HDFS, MapReduce, and Spark for large-scale data processing. ● Construct simple data pipelines using ingestion and querying tools (Hive, Sqoop, Flume, Pig, Spark SQL) to manage and analyze structured and unstructured data	
9	**Modules**	

	Module I: Introduction to Big Data and Analytics (15 Hours) 5. **Big Data Fundamentals** a. Evolution, definition, and characteristics of Big Data b. Challenges with Big Data, importance and applications c. Traditional Business Intelligence vs Big Data d. Data Warehouse environment and state of practice in analytics e. Roles in Big Data ecosystems and real-world examples 6. **Big Data Analytics Concepts** a. Introduction to analytics (descriptive, predictive, prescriptive) b. Data Analytics Life Cycle c. Challenges in Big Data Analytics d. Responsibilities in Data Science, consistency concepts (soft state, eventual consistency) 7. **Analytical Methods (Overview)** a. Clustering and Association rules (Apriori basics, applications) b. Regression: linear, logistic (introductory concepts) c. Classification basics: decision trees, Naïve Bayes d. Introduction to Time Series (concept + ARIMA overview) e. Basics of Text Analytics: TF-IDF, sentiment analysis	
	Module II: Big Data Ecosystem and Technologies (15 Hours) CO1. **Big Data Infrastructure** a. Hadoop ecosystem: architecture, HDFS, distributed computation b. Hadoop MapReduce concepts, Python with Hadoop Streaming c. Spark basics: in-memory computing, PySpark, simple applications CO2. **Data Products and Pipelines** a. Data science pipeline with Hadoop/Spark b. Building data products at scale CO3. **Distributed Analysis and Tools** a. Data mining and warehousing overview b. Structured queries with Hive, basics of HBase c. Data ingestion: Sqoop (relational data), Flume (streaming data) d. High-level APIs: Pig, Spark SQL	
10	**Reference Books** 1. **Big Data and Analytics** – Seema Acharya & Subhasini Chellappan, *Wiley India*, 1st Edition. 2. **Data Analytics with Hadoop: An Introduction for Data Scientists** – Benjamin Bengfort & Jenny Kim, *O'Reilly Media*, 1st Edition, 2016. 3. **Big Data and Hadoop** – V. K. Jain, *Khanna Publishing*, 1st Edition. 4. **Big Data: Principles and Best Practices of Scalable Realtime Data Systems** – Nathan Marz & James Warren, *Manning Publications*, 1st Edition, 2015. 5. **Hadoop: The Definitive Guide** – Tom White, *O'Reilly Media*, 4th Edition, 2015.	
11	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
12	**Continuous Evaluation through:** Class test of 15 marks Quizzes/ Presentations/ Assignments: 5 marks	**Format of Question Paper: External Examination (30 Marks)– 1 hr duration**

	Total: 20 marks	

13 **Format of Question Paper: (Semester End Examination: 30 Marks. Duration:1 hour)**

Questions	Based On	Options	Marks
Q.1	Module 1	Any 2 out of 4	10
Q.2	Module 2	Any 2 out of 4	10
Q.3	Module 1 & 2	Any 2 out of 4	10

Name of the Course: Cybersecurity & Threat Management

Sr. No.	Heading	Particulars
1	**Description the course:**	This course introduces the fundamentals of cybersecurity, covering key principles, secure communication protocols, and common cyber threats. Students will learn to analyze vulnerabilities, detect intrusions, and respond to malware and attacks using industry-standard tools. The course emphasizes hands-on Red Team and Blue Team exercises, including penetration testing, threat monitoring, and incident response. Through real-world scenarios and case studies, students gain practical skills to implement effective cybersecurity strategies and defend digital assets.
2	**Vertical:**	Major
3	**Type:**	Theory
4	**Credits:**	2 credits
5	**Hours Allotted:**	30 hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives:** **2)** To provide students with a foundational understanding of cybersecurity principles, including the CIA triad, common cyber threats, and security standards and frameworks. **3)** To develop the ability to analyze network and system vulnerabilities, detect intrusions, and identify malicious software using standard tools and techniques. **4)** To equip students with practical skills in defensive strategies, including firewall configuration, network monitoring, secure communication, and incident response. **5)** To introduce Red Team (offensive) and Blue Team (defensive) techniques, enabling students to simulate attacks, detect threats, and implement mitigation strategies. **6)** To provide an understanding of cybersecurity best practices, risk assessment, and real-world case studies to prepare students for practical security challenges.	
8	**Course Outcomes:** By the end of this course, students will be able to: ● Explain the fundamentals of cybersecurity, including secure communication protocols, web and email security, and common cyber threats. ● Analyze network and system vulnerabilities, intrusion attempts, and malware activities using standard tools and techniques. ● Demonstrate defensive strategies, including firewall configuration, network monitoring, and incident response in a simulated environment. ● Apply Red Team (offensive) and Blue Team (defensive) techniques to simulate attacks, detect threats, and implement mitigation strategies effectively.	
9	**Modules:-**	
	Module I: Cybersecurity Foundations and Secure Communication (15 Hours) 1. **Introduction to Cybersecurity** a. Goals: Confidentiality, Integrity, Availability (CIA triad)	

	b. Cyber threats: internal vs external c. Cybersecurity standards & frameworks (ISO 27001, NIST) 2. **Electronic Mail Security** a. PGP, S/MIME, digital signatures b. Phishing, spoofing, and countermeasures 3. **IP & Web Security** a. IPSec (AH, ESP, key management) b. SSL/TLS, secure web transactions (SET) c. Common web vulnerabilities (SQLi, XSS, CSRF)	
	Module II: Threat Management, Red Team / Blue Team, and Defense (15 Hours) 1. **Intrusion and Malicious Software** a. Intruders and intrusion techniques b. Malware: viruses, worms, trojans, ransomware c. DDoS attacks and mitigation 2. **Firewalls and Network Security** a. Firewall design principles and types b. Network segmentation and perimeter defense 3. **Red Team / Blue Team Exercises (Hands-On Focus)** a. **Red Team**: simulate attacks, penetration testing, vulnerability exploitation b. **Blue Team**: monitor, detect, and respond to attacks c. Tools: Wireshark, Nmap, Metasploit, Snort, Security Information and Event Management (SIEM) basics d. Incident response exercises and log analysis 4. **Security Best Practices and Case Studies** a. Risk assessment, incident response plans b. Real-world attack scenarios and mitigation strategies	
10	**Reference Books** ● **Computer Security: Principles and Practice** – William Stallings & Lawrie Brown, *Pearson*, 4th Edition, 2017 ● **Hacking: The Art of Exploitation** – Jon Erickson, *No Starch Press*, 2nd Edition, 2008 ● **The Web Application Hacker's Handbook** – Dafydd Stuttard & Marcus Pinto, *Wiley*, 2nd Edition, 2011 ● **Metasploit: The Penetration Tester's Guide** – David Kennedy, Jim O'Gorman, Devon Kearns, Mati Aharoni, *No Starch Press*, 1st Edition, 2011 ● **Cyber Security: Understanding Cyber Crimes, Computer Forensics and Legal Perspectives** – Nina Godbole & Sunit Belapure, *Wiley India*, 1st Edition, 2012	
11	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
12	**Continuous Evaluation through:** Class test of 15 marks Quizzes/ Presentations/ Assignments: 5 marks Total: 20 marks	**Format of Question Paper: External Examination (30 Marks)– 1 hr duration**

13	**Format of Question Paper: (Semester End Examination: 30 Marks. Duration:1 hour)**

Questions	**Based On**	**Options**	**Marks**
Q.1	Module 1	Any 2 out of 4	10
Q.2	Module 2	Any 2 out of 4	10
Q.3	Module 1 & 2	Any 2 out of 4	10

Name of the Course: CS Practical VI - Data Science & Big Data Analytics -Lab

Sr. No.	Heading	Particulars
1	**Description the course:**	This course provides a comprehensive introduction to Data Science, Machine Learning, and Big Data Analytics, focusing on the collection, preprocessing, analysis, visualization, and interpretation of structured and unstructured data. Students will learn to implement supervised and unsupervised machine learning techniques such as regression, classification, clustering, feature engineering, and model evaluation using Python libraries including Pandas, NumPy, Scikit-learn, and Matplotlib. The course also introduces Big Data concepts and technologies such as Hadoop, HDFS, MapReduce, Spark, Hive, Sqoop, and Flume for handling and analyzing large-scale datasets through distributed computing and data pipelines. Through practical applications and hands-on exercises, learners will develop the skills required to derive meaningful insights and build data-driven solutions for real-world problems.
2	**Vertical:**	Major
3	**Type:**	Practical
4	**Credits:**	2 credits (1 credit = 30 Hours of Practical work in a semester)
5	**Hours Allotted:**	60 hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives:**	● To introduce students to the fundamentals of Data Science, its scope, applications, and relationship with AI, ML, BI, and Data Warehousing. ● To develop skills for handling, preprocessing, and transforming various types of data from multiple sources. ● To teach students how to perform exploratory data analysis (EDA) using statistical and visualization techniques to derive insights. ● To enable students to implement and evaluate machine learning algorithms for regression, classification, and clustering. ● To provide hands-on experience with Python libraries and tools such as Pandas, NumPy, Scikit-learn, and Matplotlib for real-world data analysis and modeling.
8	**Course Outcomes:**	By the end of this course, students will be able to: 8. Explain the scope of Data Science, its applications, and its relation to AI, ML, BI, and Data Warehousing/Data Mining. 9. Identify and preprocess different types and sources of data using techniques such as cleaning, transformation, encoding, scaling, and feature selection. 10. Perform exploratory data analysis (EDA) and apply statistical and visualization methods to derive meaningful insights. 11. Implement basic machine learning algorithms for classification, regression, and clustering using standard tools and libraries. 12. Evaluate models using performance metrics such as accuracy, precision, recall, F1-score, confusion matrix, ROC curve, and cross-validation techniques.
9	**Modules:-**	

	Module 1 (30 hours):
	Module I: Data Science Foundations & Preprocessing, Data Analysis & Machine Learning 1. Python Libraries for Data Science ○ Write a Python program to create NumPy arrays and perform basic arithmetic operations. ○ Using Pandas, create a DataFrame, select specific rows and columns, and visualize the data using Matplotlib. 2. Working with Different Data Types and Sources ○ Import a dataset from a CSV/Excel/JSON file and display its summary (head, info, describe). ○ Fetch data from a public API or web page, load it into a Pandas DataFrame, and display basic statistics. 3. Data Cleaning ○ Identify missing values in a dataset and fill them using mean/median/mode imputation. ○ Remove duplicate rows from a dataset. ○ Detect outliers using statistical methods and remove or replace them. 4. Data Transformation ○ Normalize and standardize numeric columns in a dataset. ○ Encode categorical variables using label encoding and one-hot encoding. 5. Feature Engineering & Data Wrangling ○ Create new features from existing data (e.g., extracting year/month from date columns). ○ Reshape the dataset using pivot or melt operations. ○ Convert categorical variables into dummy/binary variables. 6. Exploratory Data Analysis (EDA) ○ Generate histograms, scatter plots, and box plots for a dataset. ○ Calculate mean, median, standard deviation, and interpret the results. 7. Hypothesis Testing ○ Perform a t-test to compare means of two groups in a dataset. ○ Conduct a chi-square test to examine relationships between categorical variables ○ Apply ANOVA to compare means across multiple groups. 8. Regression Models ○ Implement simple linear regression on a dataset and interpret the coefficients. ○ Apply multiple linear regression and evaluate the model using R^2 and error metrics. 9. Classification Models ○ Build a logistic regression model for a binary classification problem and evaluate its performance. ○ Create a Decision Tree Classifier for a dataset and visualize the tree structure. 10. Model Evaluation and Validation ○ Compute confusion matrix, accuracy, precision, recall, and F1-score for classification models. ○ Plot the ROC curve and calculate the AUC for a classifier. Perform k-fold cross-validation on a model and report the average performance metrics.

	Module 2 (30 hours):
	Module II: Big Data Concepts & Analytics 1. **Exploring Big Data with Python** ○ Write a Python program using **NumPy** and **Pandas** to load large structured and unstructured datasets. ○ Summarize the dataset using descriptive statistics (`.info()`, `.describe()`) and basic visualization. 2. **Data Analytics Life Cycle** ○ Given a mini case study, apply the **data analytics life cycle**: data collection, cleaning, preparation, analysis, and reporting. ○ Document each step and visualize insights using charts or tables. 3. **Clustering** ○ Apply **k-Means clustering** on a dataset (e.g., customer segmentation or sales data). ○ Visualize the clusters and interpret the results. 4. **Association Rule Mining** ○ Implement the **Apriori algorithm** on a transaction dataset to find frequent itemsets. ○ Generate association rules and evaluate their support, confidence, and lift. 5. **Regression & Classification** ○ Apply **linear regression** to a dataset (e.g., housing prices) and predict target values. ○ Implement **logistic regression** or a **decision tree classifier** on a classification dataset and evaluate model performance using accuracy, confusion matrix, and ROC curve. 6. **Text Analytics** ○ Perform **TF-IDF vectorization** on a sample text corpus. ○ Conduct **sentiment analysis** on tweets or product reviews using Python libraries such as NLTK or TextBlob. 7. **HDFS Operations** ○ Store, retrieve, and manage files in **Hadoop Distributed File System (HDFS)** using command-line operations. ○ Verify file contents and directory structures in HDFS. 8. **MapReduce with Python** ○ Write a **MapReduce program** (e.g., word count) using **Hadoop Streaming with Python**. ○ Analyze the output and explain the mapping and reducing steps. 9. **Introduction to Spark with PySpark** ○ Load a dataset in **PySpark** and perform **basic RDD operations** (map, filter, reduce). ○ Apply transformations and actions to extract meaningful insights from the data. 10. **Hive / Sqoop / Flume** ○ Use **Sqoop** to import relational data into Hadoop. ○ Run basic queries on **Hive tables** and analyze the results. Demonstrate **Flume** for streaming data ingestion from a log file or sample source
11	**Reference Books** ● **Data Science and Big Data Analytics** – P. N. Murthy & G. Girija, *PHI Learning*, 1st Edition, 2019. ● **Big Data and Analytics** – Seema Acharya & Subhasini Chellappan, *Wiley India*, 2nd Edition, 2019. ● **Foundations of Data Science Using Python** – B. M. Harwani, *BPB Publications*, 1st Edition, 2020.

	• **Statistics for Machine Learning** – Pratap Dangeti, *Packt Publishing (Indian Reprint)*, 1st Edition, 2017. • **Mining of Massive Datasets** – Anand Rajaraman, Jure Leskovec, Jeffrey D. Ullman, *Cambridge University Press (Indian Edition available)*, 3rd Edition, 2020. • **Big Data and Analytics** – Seema Acharya & Subhasini Chellappan, *Wiley India*, 1st Edition. • **Data Analytics with Hadoop: An Introduction for Data Scientists** – Benjamin Bengfort & Jenny Kim, *O'Reilly Media*, 1st Edition, 2016. • **Big Data and Hadoop** – V. K. Jain, *Khanna Publishing*, 1st Edition. • **Big Data: Principles and Best Practices of Scalable Realtime Data Systems** – Nathan Marz & James Warren, *Manning Publications*, 1st Edition, 2015. • **Hadoop: The Definitive Guide** – Tom White, *O'Reilly Media*, 4th Edition, 2015	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
13	**Continuous Evaluation through:** Practical journal submission, viva, assignments • Journal Submission : 10 Marks • Assignments: 10 Marks • Total: 20 marks	**A semester end practical examination of 2 hours duration for 30 marks as the paper pattern given below.** *Its compulsory to carry certified journal at the time of practical exam*
14	Format of Question Paper: (Semester End Practical Examination: 30 Marks. Duration:2 hours) Q1: Module 1 (12 marks) Q2: Module 2 (12 marks) Q.3 Viva (06 Marks)	

Name of the Course : Mini Project – II

(Research-Oriented Project with Implementation)

Sr. No.	Heading	Particulars
1	**Description the course:**	Mini Project – II is a research-focused course designed to develop students' analytical, experimental, and scholarly skills through systematic investigation and implementation. The course guides students from research problem identification and literature survey to methodology design, experimentation, result analysis, and research paper writing in IEEE format.
2	**Vertical:**	Major
3	**Type:**	Practical
4	**Credits:**	2 credits (1 credit = 30 Hours of Practical work in a semester)
5	**Hours Allotted:**	60 hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives:** 1. To develop skills in research problem identification and structured literature review. 2. To enable design of experimental methodology and selection of appropriate models and tools. 3. To provide hands-on experience in implementing and evaluating research prototypes. 4. To cultivate analytical skills for result interpretation and hypothesis validation. 5. To train students in professional research documentation and IEEE-format paper writing.	
8	**Course Outcomes:** After successful completion of this course, students would be able to - CO4. Formulate and justify a research problem supported by literature review and gap analysis. CO5. Design and implement an experimental framework including dataset selection, model development, and evaluation metrics. CO6. Conduct and analyze experiments using appropriate performance comparisons and baseline studies. CO7. Interpret results critically, identify limitations, and propose future research directions. CO8. Prepare and present a structured research paper and working prototype following IEEE standards	
9	**Modules:-** **Module 1 (30 hours)**	
	Research Planning, Literature Survey & Methodology Design Phase • **Research Problem Identification:** Selection of topic, Problem statement formulation, Research objectives, Hypothesis (if applicable) • **Literature Review:** Study of minimum 5–8 research papers, Identification of research gap, Comparative analysis, Summary of existing techniques	

	• **Research Methodology Design:** Dataset selection or data collection, Algorithm/model selection, Tool selection, Experimental setup planning, Definition of evaluation metrics • **Research Design Documentation:** Research Proposal, Literature Review Matrix, System Model / Architecture Diagram, Experimental Design Plan	
	Module 2 (30 hours)	
	Implementation, Experimentation & Evaluation Phase • **System Implementation:** Model building, Algorithm implementation, Experimental setup, Code optimization • **Experimental Evaluation:** Model training/testing, Performance metric calculation, Baseline comparison • **Result Analysis:** Graphical representation, Interpretation of results, Validation of hypothesis, Identification of limitations • **Research Paper Writing:** Abstract, Introduction, Related Work, Methodology, • Results, Conclusion & Future Scope **Final Deliverables** • Working prototype • Research Paper (IEEE format, 6–8 pages minimum) • GitHub Repository • Presentation	
10	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
	Internal Evaluation : (To be assessed by the Project Guide) Components : • Research Problem Identification **– 5 Marks** • Literature Review **– 5 Marks** • Methodology & Implementation Progress **– 5 Marks** • Internal Seminar / Review **— 5 Marks**	
	External Evaluation : (To be assessed by External Examiner) **Components :** Prototype / Experimental Implementation **– 10 Marks** Result Analysis & Interpretation **-10 Marks** Research Report / Paper – **5 Marks** Viva Voce **– 5 Marks**	

Elective Courses

Name of the Course: Information Retrieval

Sr. No.	Heading	Particulars	
1	**Description the course:**	**Introduction:** This course provides a comprehensive foundation in the principles, techniques, and applications of Information Retrieval (IR). It explores how systems like search engines retrieve relevant information from large datasets in response to user queries. Students will learn both classical and modern IR models, evaluation methods, and advanced topics such as web search, machine learning-based ranking, and recommender systems.	
2	**Vertical:**	**Elective**	
3	**Type:**	**Theory**	
4	**Credits:**	**2 credits**	
5	**Hours Allotted:**	**30 Hours**	
6	**Marks Allotted:**	**50 Marks**	
7	**Course Objectives:** 1. Understand the architecture and core components of Information Retrieval systems. 2. Learn techniques for indexing, term weighting, and efficient document retrieval. 3. Explore different retrieval models including Boolean, vector space, and probabilistic approaches. 4. Study advanced applications such as web search, ranking algorithms, and recommender systems.		
8	**Course Outcomes:** CO1. Design and implement basic IR systems using appropriate indexing and retrieval techniques. CO2. Apply retrieval models to match user queries with relevant documents. CO3. Evaluate IR system performance using metrics like precision, recall, and F1-score. CO4. Utilize machine learning approaches in text classification, clustering, and ranking tasks.		
9	**Modules: -** **Module 1 (15 hours):**		

	Introduction to Information Retrieval: Definition and goals of IR systems, Components of an IR system, Applications and challenges **Document Indexing and Storage:** Inverted index construction, Term weighting (TF-IDF), Storage and retrieval of indexed documents **Retrieval Models:** Boolean model (AND, OR, NOT), Vector space model (cosine similarity, query-document matching), Probabilistic model basics (Bayesian retrieval, relevance feedback) **Spelling Correction:** Challenges of spelling errors in queries, Edit distance and string similarity, Basic correction techniques **Performance Evaluation:** Evaluation metrics: precision, recall, F1-score, average precision, Test collections and relevance judgments		
	Module 2 (15 hours):		
	Text Categorization and Clustering: Text classification algorithms: Naive Bayes, SVM, Feature selection and dimensionality reduction, Text clustering: k-Means, hierarchical clustering, Applications: query expansion, result grouping **Web Information Retrieval:** Web search architecture and challenges, Crawling and indexing web pages, Link analysis: PageRank and HITS algorithms **Learning to Rank:** Supervised ranking techniques: RankSVM, RankBoost (introductory overview), Evaluation metrics for ranking **Selected Advanced IR Applications:** Recommender systems: collaborative and content-based filtering, Text summarization (extractive methods), Question answering (introductory approaches)		
10	**Reference Books** 1. Ricci, F, Rokach, L. Shapira, B. Kantor, —Recommender Systems Handbook‖, First Edition. 2. Bruce Croft, Donald Metzler, and Trevor Strohman, Search Engines: Information Retrieval in Practice, Pearson Education. 3. Stefan Buttcher, Charlie Clarke, Gordon Cormack, Information Retrieval: Implementing and Evaluating Search Engines, MIT Press.		
11	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**	
12	**Continuous Evaluation through:** Class test of 15 marks Quizzes/ Presentations/ Assignments: 5 marks Total: 20 marks	**Format of Question Paper: External Examination (30 Marks)– 1 hr duration**	

13	**Format of Question Paper: (Semester End Examination: 30 Marks. Duration:1 hour)** **Questions** | **Based On** | **Options** | **Marks** Q.1 | Module 1 | Any 2 out of 4 | 10 Q.2 | Module 2 | Any 2 out of 4 | 10 Q.3 | Module 1 & 2 | Any 2 out of 4 | 10	

Name of the Course: Ethical Hacking

Sr. No.	Heading	Particulars
1	**Description the course:**	**Introduction:** This course introduces ethical hacking and penetration testing fundamentals. It covers reconnaissance, scanning, exploitation techniques, wireless and web application security, and defensive best practices. Emphasis is placed on legal and ethical considerations, hands-on methodologies, and producing actionable penetration-test reports to improve organizational security.
2	**Vertical:**	**Elective**
3	**Type:**	**Theory**
4	**Credits:**	**2 credits (1 credit = 15 Hours for Theory)**
5	**Hours Allotted:**	**30 Hours**
6	**Marks Allotted:**	**50 Marks**
7	**Course Objectives:** 1. Teach core concepts, phases, and ethics of ethical hacking and penetration testing. 2. Develop practical skills in footprinting, scanning, sniffing, exploitation, and mitigation techniques. 3. Examine system, web, and wireless vulnerabilities and demonstrate safe testing approaches. 4. Train students to follow a structured penetration-testing methodology and produce professional reports with remediation guidance.	
8	**Course Outcomes:** CO1. Conduct responsible information gathering, scanning, and enumeration to identify security weaknesses. CO2. Demonstrate understanding of common attacks (social engineering, SQLi, buffer overflows, DoS) and apply appropriate countermeasures. CO3. Perform basic, ethically-approved exploitation and vulnerability validation in controlled environments and document findings. CO4. Produce structured penetration-test reports including risk assessment, remediation recommendations, and adherence to legal/ethical standards.	
9	**Modules** **Module 1 (15 hours):**	
	Introduction & Ethics: Terminology, hacker types, ethical hacking phases, legal/ethical considerations, skills required. **Footprinting & Reconnaissance:** Information gathering methodology, competitive intelligence, DNS enumeration, WHOIS/ARIN lookups, DNS record types, traceroute, email tracking. **Social Engineering:** Common attack types (phishing, pretexting, baiting, tailgating), basic countermeasures and awareness. **Scanning & Enumeration:** Port/network scanning fundamentals, Nmap basics and common scan types (SYN, FIN, XMAS, NULL, ACK), ping sweep, vulnerability scanning, SNMP enumeration, enumeration steps. **Sniffing & Session Attacks:** Passive vs active sniffing, ARP poisoning, MAC flooding, DNS spoofing, basics of session hijacking (concepts & prevention).	

	Denial of Service Fundamentals: DoS/DDoS overview, common techniques (SYN flood, Smurf), botnets, basic mitigation principles and defensive best practices.
	Module 2 (15 hours):
	System Hacking Essentials: Password cracking concepts and techniques, keyloggers/spyware overview, privilege escalation, rootkits (detection & mitigation). **Web Application Hacking:** Web app threats overview, common vulnerabilities (OWASP top issues), Google dorking overview, basic countermeasures and secure coding principles. **SQL Injection & Buffer Overflows:** SQLi types and exploitation steps, defensive measures; buffer overflow basics (stack overflows, mitigation concepts). **Wireless Security:** WEP/WPA/WPA2 authentication basics, common cracking techniques at conceptual level, rogue APs, wireless sniffing and defenses. **Web- Based Password Attacks & Tools:** Authentication types, password cracking tools (conceptual safe demo), password hardening best practices. **Penetration Testing Methodology & Reporting:** Pen- test phases, methodology, automation vs manual testing, common tools (conceptual), deliverables and reporting, remediation recommendations.
10	**Reference Books** 1. Certified Ethical Hacker: Michael Gregg, Pearson Education 2. Certified Ethical Hacker: Matt Walker, TMH.
11	**Internal Continuous Assessment: 40%** / **Semester End Examination: 60%**
12	**Continuous Evaluation through:** Class test of 15 marks Quizzes/ Presentations/ Assignments: 5 marks Total: 20 marks / **Format of Question Paper: External Examination (30 Marks)– 1 hr duration**
13	**Format of Question Paper: (Semester End Examination: 30 Marks. Duration:1 hour)**

Questions	Based On	Options	Marks
Q.1	Module 1	Any 2 out of 4	10
Q.2	Module 2	Any 2 out of 4	10
Q.3	Module 1 & 2	Any 2 out of 4	10

Name of the Course: Blockchain Technology & Applications

Sr. No.	Heading	Particulars
1	**Description the course:**	**Introduction:** This course introduces the foundational concepts of blockchain technology, its architecture, and practical applications. Students will explore key blockchain platforms like Bitcoin and Ethereum, learn how smart contracts work, and build basic decentralized applications (DApps) using Solidity. The course also provides a high-level understanding of NFTs, cryptoeconomics, and permissioned blockchains such as Hyperledger.
2	**Vertical:**	Elective
3	**Type:**	Theory
4	**Credits:**	2 credits
5	**Hours Allotted:**	30 hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives:** 1. Introduce the core principles of blockchain technology, decentralization, and distributed consensus. 2. Explore the Bitcoin and Ethereum networks, including transactions, wallets, and smart contracts. 3. Teach the basics of Solidity to write and deploy smart contracts on test networks. 4. Provide an overview of emerging blockchain applications such as DApps, NFTs, and permissioned systems.	
8	**Course Outcomes:** CO1. Explain how blockchain works and differentiate between centralized, decentralized, and distributed systems. CO2. Describe how transactions and consensus mechanisms operate in Bitcoin and Ethereum. CO3. Write and deploy basic smart contracts using Solidity on Ethereum test networks. CO4. Understand the concepts of cryptoeconomics, NFTs, and the role of permissioned blockchains in enterprise use cases.	
9	**Modules:-** **Module 1 (15 hours)**	
	Introduction to Blockchain: What is blockchain?, Centralized vs decentralized systems, Key features and uses of blockchain **How Blockchain Works:** Blockchain transactions, Distributed consensus (basic idea), Cryptography basics for blockchain **Bitcoin Overview:** Bitcoin blockchain and wallets, How transactions work **Ethereum Overview:** Ethereum blockchain and accounts, Smart contract basics, Gas and transaction cost (simple idea)	
	Module 2 (15 hours)	

	Smart Contracts & Solidity Basics: Simple Solidity programming (variables, loops, functions), Writing a basic smart contract **Decentralized Applications (DApps):** Deploying a smart contract on a test network, Interacting with smart contracts **Permissioned Blockchains:** Public vs private blockchain, Hyperledger basic overview **Cryptoeconomics & NFTs:** Simple explanation of cryptoeconomics, NFTs: creating, minting, and marketplaces
10	**Reference Books** 6. NPTEL courses: a. Blockchain and its Applications, b. Blockchain Architecture Design and Use Cases 7. www.swayam.gov.in/ 8. www.coursera.org 9. https://ethereum.org/en/ 10. https://www.trufflesuite.com/tutorials 11. https://hyperledger-fabric.readthedocs.io/en/release-2.2/whatis.h
11	**Internal Continuous Assessment: 40%** / **Semester End Examination: 60%**
12	**Continuous Evaluation through:** Class test of 15 marks Quizzes/ Presentations/ Assignments: 5 marks Total: 20 marks **Format of Question Paper: External Examination (30 Marks)– 1 hr duration**
13	**Format of Question Paper: (Semester End Examination: 30 Marks. Duration:1 hour)**

Questions	Based On	Options	Marks
Q.1	Module 1	Any 2 out of 4	10
Q.2	Module 2	Any 2 out of 4	10
Q.3	Module 1 & 2	Any 2 out of 4	10

Name of the Course: Information Retrieval - LAB

Sr. No.	Heading	Particulars
1	**Description the course:**	**Introduction:** This practical module complements the theoretical aspects of Information Retrieval by guiding students through the implementation of core IR techniques. Students will gain hands-on experience in building IR systems, working with retrieval models, evaluating system performance, and exploring advanced applications such as classification, clustering, and web search. The goal is to bridge the gap between IR concepts and their real-world implementation.
2	**Vertical:**	Elective
3	**Type:**	Practical
4	**Credits:**	2 credits (1 credit = 15 Hours of Practical work in a semester)
5	**Hours Allotted:**	30 hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives:** 1. Build and evaluate a functional Information Retrieval system using indexing and retrieval techniques. 2. Apply classical and modern retrieval models including Boolean, vector space, and TF-IDF. 3. Integrate spelling correction and implement evaluation metrics for system performance.	
8	**Course Outcomes:** CO1. Construct an inverted index and retrieve documents using Boolean and vector space models. CO2. Compute TF-IDF weights and implement cosine similarity for document ranking. CO3. Develop and evaluate spelling correction modules and IR system performance using precision and recall. CO4. Implement web crawlers, PageRank, classifiers (Naive Bayes/SVM), clustering (k-Means), and basic learning-to-rank and summarization techniques.	
9	**Modules:-** **Module 1 (15 hours):**	
	Document Indexing and Retrieval ○ Implement an inverted index construction algorithm for a small set of documents. ○ Build a simple document retrieval system that takes a query and returns matching documents using the constructed index. 2. **Boolean Retrieval Model** ○ Implement Boolean queries (AND, OR, NOT) on the document collection. ○ Process sample queries and return relevant documents based on Boolean logic. 3. **Vector Space Model (TF-IDF)** ○ Compute TF-IDF weights for terms in a document collection.	

	○ Implement cosine similarity to rank documents for a given query. 4. **Spelling Correction** ○ Develop a spelling correction module using edit distance (Levenshtein distance). ○ Integrate it into the document retrieval system to handle misspelled queries. 5. **Evaluation Metrics for IR Systems** ○ Calculate precision, recall, and F1-score for retrieval results. ○ Use an evaluation toolkit (or Python scripts) to compute average precision and other metrics.	
	Module 2 (15 hours):	
	6. Text Categorization ○ Implement a Naive Bayes or SVM classifier for text documents. ○ Train the classifier on a labeled dataset and evaluate performance using accuracy, precision, and recall. 7. **Clustering for IR** ○ Implement a clustering algorithm (K-Means or hierarchical). ○ Apply it to a document collection and evaluate clusters using silhouette score or purity metrics. 8. **Web Crawling and Indexing** ○ Develop a basic web crawler to fetch and index a set of web pages. ○ Handle challenges like robots.txt, delays, and simple dynamic content. 9. **Link Analysis and PageRank** ○ Implement the PageRank algorithm on a small web graph. ○ Analyze and rank pages based on link structure. 10. **Learning to Rank & Summarization** ○ Implement a simple learning-to-rank algorithm (RankSVM or RankBoost) on a small dataset. ○ Implement an extractive text summarization algorithm to generate summaries from documents.	
10	**Reference Books:** 1. Linux Administration: A Beginner's Guide, Wale Soyinka, Seventh Edition, McGraw-Hill Education, 2016 2. Ubuntu Server Guide, Ubuntu Documentation Team, 2016	
11	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
12	**Continuous Evaluation through:** Practical journal submission, viva, assignments ● Journal Submission : 10 Marks ● Assignments: 10 Marks ● Total: 20 marks	**A semester end practical examination of 2 hours duration for 30 marks as the paper pattern given below.** *Its compulsory to carry certified journal at the time of practical exam*
13	Format of Question Paper: (Semester End Practical Examination: 30 Marks. Duration:2 hours) Q1: Module 1 (12 marks) Q2: Module 2 (12 marks) Q.3 Viva (06 Marks)	

Name of the Course: Ethical Hacking - LAB

Sr. No.	Heading	Particulars
1	**Description the course:**	**Introduction:** These hands-on labs teach practical reconnaissance, cracking, network analysis, web-app exploitation, and exploitation workflows in safe, controlled environments. Emphasis is on responsible, legal practice—all activities are performed on instructor-provided targets or isolated VMs and focus on learning defensive controls as well as attack techniques.
2	**Vertical:**	Elective
3	**Type:**	Practical
4	**Credits:**	2 credits (1 credit = 15 Hours of Practical work in a semester)
5	**Hours Allotted:**	30 hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives:** 1. Practice open-source intelligence and footprinting techniques using Google operators and WHOIS to map target surfaces. 2. Demonstrate password handling, encryption concepts, and controlled password-cracking methods to highlight authentication weaknesses. 3. Perform network and application security exercises (scanning, ARP poisoning, packet analysis, XSS, SQLi, session manipulation) in isolated labs. 4. Execute an end-to-end, ethical exploitation workflow using safe vulnerable VMs and document findings for remediation.	
8	**Course Outcomes:** CO1. Effectively gather public intelligence on a domain using Google dorks and WHOIS, and interpret the results for attack surface analysis. CO2. Explain stream cipher concepts (RC4 demo) and demonstrate the risks of weak passwords by safely running dictionary attacks on lab hashes. CO3. Use tools such as Nmap, Wireshark, Burp Suite, and Metasploit in controlled environments to identify, exploit, and mitigate common network and web vulnerabilities. CO4. Produce clear, ethical lab reports that describe methods, evidence, risk impact, and practical remediation steps while adhering to legal and safety constraints.	

9	**Modules:-** **Module 1 (15 hours):**
	1. **Google & WHOIS Reconnaissance** *Objective:* Use Google advanced operators and WHOIS lookups to gather public intelligence about a target domain. *Tools:* Browser, whois. 2. **Password Encryption & Decryption (CrypTool demo)** *Objective:* Demonstrate RC4 (stream cipher) encryption and decryption to illustrate password handling and cryptographic concepts. *Tools:* CrypTool (educational). 3. **Password Cracking — Controlled Demo** *Objective:* Show dictionary/wordlist attacks against instructor- provided hashes/accounts to demonstrate weak password risks. *Tools:* John the Ripper / Hashcat (lab hashes only). 4. **Linux Network Basics & ARP Poisoning (Isolated VM)** *Objective:* Practice ip/ifconfig, ping, netstat, traceroute, and observe ARP spoofing effects in an isolated virtual network. Tools: Linux VM, ettercap / arping (host- only network). 5. Port Scanning with Nmap Objective: Run SYN, FIN, NULL, XMAS and ACK scans against lab hosts and interpret the results. *Tools:* Nmap / Zenmap.
	Module 2 (15 hours):
	6. **Network Traffic Capture & Analysis (Wireshark)** *Objective:* Capture packets, filter traffic, and analyze protocol fields to identify suspicious patterns. *Tools:* Wireshark, sample PCAPs. 7. **Persistent Cross- Site Scripting (XSS) — Safe Lab** *Objective:* Exploit a persistent XSS vulnerability on a deliberately vulnerable app and study impacts and mitigations. *Tools:* DVWA / OWASP Juice Shop (local). 8. **Session Manipulation (Tamper / Intercept Tools)** *Objective:* Intercept and modify HTTP requests to demonstrate session impersonation risks and the need for secure session management. *Tools:* Burp Suite Community / browser tamper add- ons, local test app. 9. **SQL Injection — Exploit & Fix (Demo)** *Objective:* Demonstrate SQL injection on a lab app, extract sample data, then apply parameterized queries to remediate. *Tools:* DVWA / Juice Shop, local DB. 10. **Introductory Exploitation Workflow (Metasploit on Lab VM)** *Objective:* Use Metasploit against an intentionally vulnerable VM to demonstrate exploit identification, execution, and responsible reporting. *Tools:* Kali VM (Metasploit), Metasploitable or vuln VM.
10	**Reference Books:** 1. CEH official Certfied Ethical Hacking Review Guide, Wiley India Edition. 2. Certified Ethical Hacker: Michael Gregg, Pearson Education. 3. Certified Ethical Hacker: Matt Walker, TMH.

11	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
12	**Continuous Evaluation through:** Practical journal submission, viva, assignments ● Journal Submission : 10 Marks ● Assignments: 10 Marks ● Total: 20 marks	**A semester end practical examination of 2 hours duration for 30 marks as the paper pattern given below.** *Its compulsory to carry certified journal at the time of practical exam*
13	Format of Question Paper: (Semester End Practical Examination: 30 Marks. Duration:2 hours) Q1: Module 1 (12 marks) Q2: Module 2 (12 marks) Q.3 Viva (06 Marks)	

Name of the Course: Blockchain Technology & Application - LAB

Sr. No.	Heading	Particulars
1	**Description the course:**	**Introduction:** This practical module introduces students to the hands-on aspects of blockchain, smart contracts, NFTs, and decentralized applications. Using test networks and simulators, students will explore how transactions are validated, how wallets and contracts operate, and how DApps connect to the blockchain. The labs are designed to provide real-world exposure in a safe, sandboxed environment.
2	**Vertical:**	Elective
3	**Type:**	Practical
4	**Credits:**	2 credits (1 credit = 15 Hours of Practical work in a semester)
5	**Hours Allotted:**	30 hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives:** 1. Demonstrate the core structure and behavior of blockchain through simulators and wallet setups. 2. Practice writing, deploying, and interacting with basic smart contracts using Ethereum testnets. 3. Explore NFT creation, smart contract interaction, and Ether transactions on test networks. 4. Introduce permissioned blockchain concepts using Hyperledger Fabric and simulate basic mining.	
8	**Course Outcomes:** CO1. Visualize how blockchain links blocks and transactions through simulators. CO2. Create and manage Bitcoin and Ethereum test wallets, and execute test transactions. CO3. Write, deploy, and interact with smart contracts and NFTs on Ethereum test networks. CO4. Understand permissioned blockchain workflows via Hyperledger and simulate blockchain mining and rewards.	
9	**Modules:-** **Module 1 (15 hours):**	
	1. **Explore Blockchain Basics** ○ Use an online blockchain simulator to create a few blocks and see how transactions are linked. 2. **Bitcoin Wallet Demo** ○ Set up a Bitcoin wallet on a testnet and check balances. 3. **Ethereum Wallet Demo** ○ Create an Ethereum wallet on a test network and explore account addresses. 4. **Simple Solidity Contract**	

	○ Write a “Hello World” smart contract in Remix IDE and deploy it on a test network. 5. **Read & Write Contract Values** ○ Add a function in your smart contract to store and retrieve a value.	
	Module 2 (15 hours):	
	6. **Send Test Ether** ○ Transfer small amounts of Ether between test accounts on a test network. 7. **Mint a Simple NFT** ○ Create a basic NFT using a test network and view its ownership. 8. **Interact with a DApp** ○ Connect a deployed smart contract to a simple webpage using Web3.js and perform a transaction. 9. **Hyperledger Fabric Chaincode Demo** ○ Deploy a simple chaincode on a local Hyperledger Fabric test environment. 10. **Simple Mining Simulation** ● Use an online simulator or Python notebook to demonstrate block creation and rewards.	
10	**Reference Books** 1. Mastering Blockchain: Inner workings of blockchain, from cryptography and decentralized identities, to DeFi, NFTs and Web3, 4th Edition 2. Blockchain and the Supply Chain: Concepts, Strategies and Practical Applications, Kogan Page, Nick Vyas, Aljosja Beije and Bhaskar Krishnamachari	
11	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
12	**Continuous Evaluation through:** Practical journal submission, viva, assignments ● Journal Submission : 10 Marks ● Assignments: 10 Marks ● Total: 20 marks	**A semester end practical examination of 2 hours duration for 30 marks as the paper pattern given below.** *Its compulsory to carry certified journal at the time of practical exam*
13	Format of Question Paper: (Semester End Practical Examination: 30 Marks. Duration:2 hours) Q1: Module 1 (12 marks) Q2: Module 2 (12 marks) Q.3 Viva (06 Marks)	

Vertical – II

Name of the Course: Machine Learning Algorithms

Sr. No.	Heading	Particulars
1	**Description the course:**	This course builds on foundational machine learning concepts, focusing on supervised, unsupervised, and reinforcement learning techniques. Students will learn to implement regression and classification algorithms, evaluate model performance, and optimize models using cross-validation and hyperparameter tuning. The course also covers clustering, dimensionality reduction, and reinforcement learning algorithms such as Q-Learning and SARSA. Through practical applications, including market segmentation, anomaly detection, recommendation systems, and resource optimization, students gain hands-on experience applying machine learning techniques to real-world problems.
2	**Vertical:**	**Minor**
3	**Type:**	**Theory**
4	**Credits:**	**2 credits**
5	**Hours Allotted:**	**30 Hours**
6	**Marks Allotted:**	**50 Marks**
7	**Course Objectives:**	4. To provide students with a deep understanding of supervised learning algorithms, including regression, classification, and model evaluation techniques. 5. To develop practical skills in implementing decision trees, random forests, k-NN, and Naïve Bayes classifiers, along with optimization using cross-validation and hyperparameter tuning. 6. To introduce students to unsupervised learning methods such as clustering (k-Means, hierarchical) and dimensionality reduction (PCA) for exploratory data analysis. 7. To familiarize students with the fundamentals of reinforcement learning, including agents, environment, reward systems, and basic algorithms like Q-Learning and SARSA. 8. To enable students to apply machine learning techniques to real-world problems, including market segmentation, anomaly detection, recommendation systems, and resource optimization.

8	**Course Outcomes:** By the end of this course, students will be able to: 4. Apply supervised learning algorithms (regression and classification) to real-world datasets and evaluate their performance. 5. Implement unsupervised learning techniques such as clustering and dimensionality reduction for pattern discovery and data summarization. 6. Simulate and explain basic reinforcement learning algorithms (e.g., Q-Learning, SARSA) in simple environments. 7. Analyze and optimize machine learning models using appropriate evaluation metrics, cross-validation, and hyperparameter tuning.	
9	**Modules: -**	
	Module I: Supervised Learning (12 Hours) 1. **Regression Algorithms** a. Linear Regression, Multiple Regression b. Logistic Regression for classification c. Model evaluation: MSE, RMSE, R^2 for regression; confusion matrix, accuracy, precision, recall, F1-score for classification 2. **Classification Algorithms** a. Decision Trees, Random Forest b. k-Nearest Neighbors (k-NN) c. Naïve Bayes 3. **Model Optimization** a. Overfitting vs underfitting, bias-variance tradeoff b. Cross-validation and hyperparameter tuning	
	Module 2 (15 hours):	
	Module II: Unsupervised and Reinforcement Learning (18 Hours) **1. Unsupervised Learning (9 Hours)** 8. Clustering: k-Means, Hierarchical Clustering 9. Dimensionality Reduction: Principal Component Analysis (PCA) 10. Applications: Market segmentation, anomaly detection, visualization **2. Reinforcement Learning (9 Hours)** 1. Introduction to RL: Agent, Environment, Reward, Policy 2. Q-Learning, SARSA (simple tabular examples) 3. Applications: Game playing, recommendation systems, resource optimization 4. Exploration vs Exploitation, reward shaping	
10	**Reference Books** ● **Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow** – Aurélien Géron, *O'Reilly Media*, 2nd Edition, 2019 ● **Introduction to Machine Learning with Python** – Andreas C. Müller & Sarah Guido, *O'Reilly Media (Indian Edition available)*, 1st Edition, 2016 ● **Reinforcement Learning: An Introduction** – Richard S. Sutton & Andrew G. Barto, *MIT Press*, 2nd Edition, 2018 ● **Python Machine Learning** – Sebastian Raschka & Vahid Mirjalili, *Packt Publishing*, 3rd Edition, 2019	
11	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**

<table>
<tr><td>12</td><td>Continuous Evaluation through:
Class test of 15 marks
Quizzes/ Presentations/
Assignments: 5 marks
Total: 20 marks</td><td>Format of Question Paper: External Examination (30 Marks)– 1 hr duration</td></tr>
<tr><td>13</td><td colspan="2">Format of Question Paper: (Semester End Examination: 30 Marks. Duration:1 hour)
<table>
<tr><th>Questions</th><th>Based On</th><th>Options</th><th>Marks</th></tr>
<tr><td>Q.1</td><td>Module 1</td><td>Any 2 out of 4</td><td>10</td></tr>
<tr><td>Q.2</td><td>Module 2</td><td>Any 2 out of 4</td><td>10</td></tr>
<tr><td>Q.3</td><td>Module 1 & 2</td><td>Any 2 out of 4</td><td>10</td></tr>
</table></td></tr>
</table>

Name of the Course: Machine Learning algorithms - Lab

Sr. No.	Heading	Particulars
1	**Description the course:**	This course focuses on practical applications of advanced machine learning techniques. Students will gain hands-on experience in supervised learning algorithms such as regression, classification, decision trees, random forests, k-NN, and Naïve Bayes, including model evaluation and optimization. The course also covers unsupervised learning techniques like clustering and dimensionality reduction for exploratory data analysis. Additionally, students will explore reinforcement learning algorithms such as Q-Learning and SARSA, applying them to simple grid-world environments and real-world simulation tasks. Emphasis is placed on using Python libraries such as Scikit-learn and NumPy to implement end-to-end machine learning solutions.
2	**Vertical:**	**Minor**
3	**Type:**	**Practical**
4	**Credits:**	**2 credits (1 credit = 15 Hours for Theory)**
5	**Hours Allotted:**	**30 Hours**
6	**Marks Allotted:**	**50 Marks**
7	**Course Objectives:** 1. To develop practical skills in implementing supervised learning algorithms such as regression, classification, decision trees, random forests, k-NN, and Naïve Bayes. 2. To enable students to evaluate, optimize, and tune machine learning models using cross-validation, hyperparameter tuning, and learning curve analysis. 3. To introduce students to unsupervised learning techniques, including clustering and dimensionality reduction, for exploratory data analysis. 4. To provide hands-on experience with reinforcement learning algorithms, including Q-Learning and SARSA, and their applications in simple environments. 5. To equip students to apply machine learning and reinforcement learning techniques to solve real-world problems using Python and standard libraries.	
8	**Course Outcomes:** By the end of this course, students will be able to: 9. Implement supervised learning algorithms for regression and classification problems and evaluate their performance using appropriate metrics. 10. Optimize machine learning models using cross-validation, hyperparameter tuning, and techniques to address overfitting and underfitting. 11. Apply unsupervised learning methods such as k-Means clustering, hierarchical clustering, and PCA for data exploration and visualization. 12. Implement basic reinforcement learning algorithms (Q-Learning and SARSA) and analyze the agent's behavior in a simulated environment. 13. Design and execute end-to-end practical machine learning workflows for real-world datasets, integrating supervised, unsupervised, and reinforcement learning methods.	

9	**Modules** **Module 1 (15 hours):**
	Module I: Supervised Learning Practicals ● **Linear Regression** ○ Implement **simple linear regression** on a dataset (e.g., housing prices) using Python/Scikit-learn. ○ Evaluate model performance using MSE, RMSE, and R^2. ● **Multiple Regression** ○ Perform **multiple linear regression** with multiple predictor variables. ○ Interpret coefficients and evaluate model accuracy. ● **Logistic Regression** ○ Apply **logistic regression** for a binary classification problem (e.g., predicting loan approval). ○ Evaluate using confusion matrix, accuracy, precision, recall, and F1-score. ● **Decision Tree Classifier** ○ Build a **decision tree** on a classification dataset. ○ Visualize the tree structure and analyze feature importance. ● **Random Forest Classifier** ○ Implement **random forest** for classification. ○ Compare performance with a single decision tree using evaluation metrics. ● **k-Nearest Neighbors (k-NN)** ○ Apply **k-NN classifier** on a dataset. ○ Experiment with different values of k and evaluate performance. ● **Naïve Bayes Classifier** ○ Implement **Gaussian/Multinomial Naïve Bayes** for a classification dataset. ○ Evaluate model performance using standard metrics. ● **Model Optimization** ○ Perform **cross-validation** on one of the above models. ○ Apply **hyperparameter tuning** (GridSearchCV/RandomizedSearchCV) and compare results. ○ Analyze overfitting and underfitting using learning curves.
	● **Module 2 (15 hours):**
	Module II: Unsupervised and Reinforcement Learning Practicals ● **Clustering and Dimensionality Reduction** ○ Apply **k-Means clustering** and **hierarchical clustering** on a dataset (e.g., customer segmentation). ○ Reduce dimensionality using **PCA** and visualize clusters in 2D/3D. ● **Reinforcement Learning (Q-Learning / SARSA)** ○ Implement a **tabular Q-Learning** algorithm on a simple grid-world environment. ○ Compare with **SARSA**, analyze rewards, and discuss exploration vs exploitation. ○ Optionally, implement a basic recommendation or resource allocation simulation using RL principles.
11	**Reference Books** 5. **Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow** – Aurélien Géron, *O'Reilly Media*, 2nd Edition, 2019 6. **Introduction to Machine Learning with Python** – Andreas C. Müller & Sarah Guido, *O'Reilly Media (Indian Edition available)*, 1st Edition, 2016 7. **Reinforcement Learning: An Introduction** – Richard S. Sutton & Andrew G. Barto, *MIT Press*, 2nd Edition, 2018 8. **Python Machine Learning** – Sebastian Raschka & Vahid Mirjalili, *Packt Publishing*, 3rd Edition, 2019

12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
13	**Continuous Evaluation through:** Practical journal submission, viva, assignments ● Journal Submission : 10 Marks ● Assignments: 10 Marks ● Total: 20 marks	**A semester end practical examination of 2 hours duration for 30 marks as the paper pattern given below.** *Its compulsory to carry certified journal at the time of practical exam*
14	Format of Question Paper: (Semester End Practical Examination: 30 Marks. Duration:2 hours) Q1: Module 1 (12 marks) Q2: Module 2 (12 marks) Q.3 Viva (06 Marks)	

Letter Grades and Grade Points:

Semester GPA/ Program CGPA Semester/ Program	% of Marks	Alpha-Sign / Letter Grade Result		Grade Points
9.00-10.00	90.0-100	O	(Outstanding)	10
8.00-<9.00	80.0-<90.0	A+	(Excellent)	9
7.00-<8.00	70.0-<80.0	A	(Very Good)	8
6.00-<7.00	60.0-<70.0	B+	(Good)	7
5.50-<6.00	55.0-<60.0	B	(Above Average)	6
5.00-<5.50	50.0-<55.0	C	(Average)	5
4.00-<5.00	40.0-<50.0	P	(Pass)	4
Below 4.00	Below 40	F	(Fail)	0
Ab (Absent)	-	AB	(Absent)	0

Signature of BOS Chairman	***Signature of Faculty Member***	***Signature of I/C Principal***
Dr. Bhakti Chaudhari	***Ms. Jeenal Jain***	***Dr. Swiddle D'Cunha***
BOS (Computer Science)	*(Computer Science)*	