



AVANTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Autonomous)

(Approved by A.I.C.T.E., New Delhi & Permanently Affiliated to JNTU-GV, Vizianagaram)

NAAC “A+” Accredited Institute

Cherukupally (Village), Near Tagarapuvalasa Bridge, Vizianagaram (Dist), AP, Pin-531162.

www.aietta.ac.in, principal@aietta.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Institute Vision:

To develop highly skilled professionals with strong ethical and human values.

Institute Mission:

M1: To impart quality education through effective teaching–learning practices, industrial exposure, and professional training.

M2: To produce competent and highly knowledgeable graduates with a positive attitude and problem-solving ability.

M3: To instill self-confidence, leadership qualities, and social responsibility to face professional and societal challenges.

Department Vision:

To emerge as a center of excellence in Computer Science and Engineering education and research, producing competent, industry-ready professionals with innovation, ethical values, and social responsibility.

Department Mission:

M1: To impart strong foundational knowledge in computer science, programming, and emerging technologies through effective teaching–learning practices.

M2: To develop problem-solving, analytical, and research skills by encouraging innovation, projects, and industry interaction.

M3: To produce industry-ready professionals with proficiency in modern tools, teamwork, and professional communication.

M4: To inculcate ethical values, social responsibility, and lifelong learning skills to meet professional and societal challenges.

Program Educational Objectives (PEOs):

PEO1: Graduates will apply strong foundations in computer science, mathematics, and engineering principles to analyze and solve real-world problems.

PEO2: Graduates will design, develop, and deploy software systems and applications using modern Engineering tools, technologies, and industry-standard practices.

PEO3: Graduates will pursue successful careers in industry, entrepreneurship, higher education, or research through continuous learning and skill enhancement.

PEO4: Graduates will demonstrate professional ethics, effective communication, teamwork, and leadership while contributing responsibly to society.



AVANTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Autonomous)

(Approved by A.I.C.T.E., New Delhi & Permanently Affiliated to J.N.T.U-GV, Vizianagaram)

NAAC "A+" Accredited Institute

Cherukupally (Village), Near Tagarapuvalasa Bridge, Vizianagaram (Dist) -531162.

www.aietta.ac.in, principal@aietta.ac.in

Program Specific Outcomes (PSOs):

PSO1: Apply core computing concepts to design efficient software and systems.

PSO2: Use modern tools and emerging technologies to solve real-world problems ethically.

Program Outcomes (POs):

PO1: Engineering Knowledge: Apply knowledge of mathematics, natural science, computing, engineering fundamentals and an engineering specialization as specified in WK1 to WK4 respectively to develop to the solution of complex engineering problems.

PO2: Problem Analysis: Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions with consideration for sustainable development. (WK1 to WK4)

PO3: Design/Development of Solutions: Design creative solutions for complex engineering problems and design/develop systems/components/processes to meet identified needs with consideration for the public health and safety, whole-life cost, net zero carbon, culture, society and environment as required. (WK5)

PO4: Conduct Investigations of Complex Problems: Conduct investigations of complex engineering problems using research-based knowledge including design of experiments, modelling, analysis & interpretation of data to provide valid conclusions. (WK8).

PO5: Engineering Tool Usage: Create, select and apply appropriate techniques, resources and modern engineering & IT tools, including prediction and modelling recognizing their limitations to solve complex engineering problems. (WK2 and WK6)

PO6: The Engineer and The World: Analyze and evaluate societal and environmental aspects while solving complex engineering problems for its impact on sustainability with reference to economy, health, safety, legal framework, culture and environment. (WK1, WK5, and WK7).

PO7: Ethics: Apply ethical principles and commit to professional ethics, human values, diversity and inclusion; adhere to national & international laws. (WK9)

PO8: Individual and Collaborative Team work: Function effectively as an individual, and as a member or leader in diverse/multi-disciplinary teams.

PO9: Communication: Communicate effectively and inclusively within the engineering community and society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations considering cultural, language, and learning differences



AVANTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Autonomous)

(Approved by A.I.C.T.E., New Delhi & Permanently Affiliated to J.N.T.U-GV, Vizianagaram)

NAAC “A+” Accredited Institute

Cherukupally (Village), Near Tagarapuvalasa Bridge, Vizianagaram (Dist) -531162.

www.aietta.ac.in, principal@aietta.ac.in

PO10: Project Management and Finance: Apply knowledge and understanding of engineering management principles and economic decision-making and apply these to one's own work, as a member and leader in a team, and to manage projects and in multidisciplinary environments.

PO11: Life-Long Learning: Recognize the need for, and have the preparation and ability for i) independent and life-long learning ii) adaptability to new and emerging technologies and iii) critical thinking in the broadest context of technological change. (WK8).



AVANTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Autonomous)

(Approved by A.I.C.T.E., New Delhi & Permanently Affiliated to JNTU-GV, Vizianagaram)

NAAC "A+" Accredited Institute

Cherukupally (Village), Near Tagarapuvalasa Bridge, Vizianagaram (Dist), AP, Pin -531162.

www.aietta.ac.in, principal@aietta.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Program: B. Tech Computer Science and Engineering

Regulation: R24

Course Structure: IV Year I Semester (or) Semester-VII

S.No	Category	Course Code	Course Title	Hours per Week			
				Lecture	Tutorial	Practical	Credits
1	PC	R24CSPC25	Neural Network & Deep Learning	3	0	0	3
2	HS	R24HS10	Human Resources & Project Management	2	0	0	2
3	PE-III	R24CSPE11 R24CSPE12 R24CSPE13 R24CSPE14 R24CSPE15	1. Blockchain Technology 2. Big Data Analytics 3. Natural Language Processing 4. Computer Graphics and Multimedia 5. 12-Week MOOC Swayam / NPTEL course recommended by the BoS	3	0	0	3
4	PE-IV	R24CSPE16 R24CSPE17 R24CSPE18 R24CSPE19 R24CSPE20	1. Distributed Systems 2. Soft Computing 3. Foundations of Data Science 4. Information Retrieval systems 5. 12-Week MOOC Swayam / NPTEL course recommended by the BoS	3	0	0	3
5	OE-III	R24CSOE11 R24CSOE12 R24CSOE13 R24CSOE14 R24CSOE15	1. Python Programming 2. Object oriented Analysis and Design 3. Image Processing 4. Web Technologies 5. Fundamentals of Information Security	3	0	0	3
6	SC	R24CSSC05	Data Analytics and Visualization	0	0	2	1
7	SI	R24CSSI01	Evaluation of Industry Internship	0	0	0	2
8	PJ	R24CSPJ01	Project Work Phase-I: Project topic, Problem statement, literature survey	0	0	6	0
9	AC-II	R24AC02	Constitution of India	2	0	0	0
Total				14	00	08	17

Category	Courses	Credits
PC- Professional Core Courses	1	03
PE- Professional Elective Courses	2	06
OE- Open Elective Courses	1	03
SC- Skill Oriented Course	1	01
HS- Humanity Science and Management Course	1	02
PJ- Project Work Phase-I	1	00
AC-Audit Course	1	00
SI-Summer Internship	1	02
Total	10	17



AVANTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Autonomous)

(Approved by A.I.C.T.E., New Delhi & Permanently Affiliated to JNTU-GV, Vizianagaram)

NAAC “A+” Accredited Institute

Cherukupally (Village), Near Tagarapuvalasa Bridge, Vizianagaram (Dist), AP, Pin -531162.

www.aietta.ac.in, principal@aietta.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Program: B. Tech Computer Science and Engineering

Regulation: R24

Course Structure: IV Year II Semester (or) Semester-VIII

S.No	Category	Course Code	Course Title	Hours per Week			
				Lecture	Tutorial	Practical	Credits
1	PE-V	R24CSPE21 R24CSPE22 R24CSPE23 R24CSPE24 R24CSPE25	1. Quantum Computing 2. Principles of Deep Learning 3. UI/UX Design 4. Computer Vision 5. 12-Week MOOC Swayam / NPTEL course recommended by the BoS	3	0	0	3
2	OE-IV	R24CSOE16 R24CSOE17 R24CSOE18 R24CSOE19 R24CSOE20	1. Computer Architecture 2. Computer Forensics and Law Protection 3. Data Mining 4. Network Security 5. Ethical Hacking	3	0	0	3
3	PJ	R24CSPJ02	Internship & Project Work	0	0	24	12
Total				06	00	16	18

Category	Courses	Credits
PE- Professional Elective Courses	1	03
OE- Open Elective Courses	1	03
PJ- Internship & Project Work	1	12
Total	03	18

Note:

1. Student need to do at least ONE MOOC/NPTEL Course (of 3 credits out of 160 credits) to meet the mandatory requirement (11th criteria, as per R24 Regulations); they are allowed to register one semester in advance.
2. Open Elective can be Inter Department Disciplinary Course, Emerging Courses or MOOC. Final decision will be taken by the department.

Chairperson
Board of Studies (CSE)

Course Objectives:

1. To understand fundamentals of artificial neural networks, learning algorithms, and perceptron models.
2. To study deep learning architectures, optimization techniques, and regularization methods.
3. To analyze and implement convolutional and recurrent neural network models for real-world applications.
4. To understand deep generative models such as RBMs, DBNs, and Boltzmann Machines.
5. To develop problem-solving skills for designing intelligent systems using neural network and deep learning techniques.

Course Outcomes:

At the end of the course, students will be able to:

Course Code	Course Outcomes (COs)	Mapping with POs and PSOs												Dok	
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1		PSO2
R24CSPC25 .1	Understand and apply neural network fundamentals, perceptrons, learning rules and back propagation algorithms.	3	2	2	1	2	-	-	-	-	-	-	3	2	L1, L2
R24CSPC25 .2	Analyze deep learning concepts, optimization methods and regularization techniques for training deep neural networks.	3	3	2	2	3	-	-	-	-	-	1	3	2	L2, L3
R24CSPC25 .3	Design and implement convolutional neural networks for image-related applications.	3	3	3	2	3	-	-	-	1	-	-	3	3	L3, L4
R24CSPC25 .4	Apply recurrent neural networks and sequence models for temporal and sequential data processing.	3	3	3	2	3	-	-	-	1	-	-	3	3	L4, L5
R24CSPC25 .5	Understand and apply deep generative models for representation learning and probabilistic modeling.	3	2	2	2	3	-	-	-	-	-	1	3	2	L5, L6

SYLLABUS**UNIT I:****14 Hours**

Introduction to Neural Networks: Introduction to Neuron Model, Neural Network Architecture,

Learning Rules, Perceptrons, Single Layer Perceptrons, Multilayer Perceptrons, Back propagation Networks, Overfitting and Underfitting, Applications of Neural Networks. **CO's –CO1**

Self-Learning Topics: Kohonen's self-organizing networks, Hopfield network

UNIT II:

13 Hours

Introduction to Deep Learning: History of Deep Learning, Applications of Deep learning, Greedy layer wise training, Optimization for training DNNs, Newer optimization methods for neural networks (AdaGrad, RMSProp, Adam), Second order methods for training, Regularization methods (dropout, drop connect, batch normalization) **CO's –CO2**

Self-Learning Topics: Hessian-based optimization, Data augmentation as regularization

UNIT III:

13 Hours

Convolutional Neural Networks: Introduction to CNN, From Fully-Connected Layers to Convolutions, Convolutions for Images, Multiple Input and Multiple Output Channels, Pooling, LeNet, AlexNet, Networks Using Blocks (VGG), Networks with Parallel Concatenations (GoogLeNet), Residual Networks (ResNet), Densely Connected Networks (DenseNet) **CO's –CO3**

Self-Learning Topics: Batch Normalization, Network in Network (NiN)

UNIT IV:

14 Hours

Recurrent Neural Networks: Introduction to RNN, Implementation of Recurrent Neural Networks from Scratch, Back propagation Through Time, Gated Recurrent Units (GRU), Long Short-Term Memory (LSTM), Bidirectional Recurrent Neural Networks. **CO's –CO4**

Self-Learning Topics: Concise Implementation of Recurrent Neural Networks, Sequence to Sequence

UNIT V:

15 Hours

Deep Generative Models: Boltzmann Machines, Restricted Boltzmann Machines (RBMs), Deep Belief Networks (DBNs), Deep Boltzmann Machines (DBMs), Boltzmann Machines for Real-Valued Data, Convolutional Boltzmann Machines, Boltzmann Machines for Structured or Sequential Outputs, Other Boltzmann Machines. **CO's –CO5**

Self-Learning Topics: Back-Propagation through Random Operations, Directed Generative Net

Board of Studies : Computer Science and Engineering

Approved in BoS No : 03

Approved in ACM No: 03

Text Book:

1. Goodfellow, I., Bengio, Y., and Courville, A., Deep Learning, MIT Press, 2016.
2. Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola, Dive into Deep Learning, 2021
3. Learning Deep Learning by Magnus Ekmen, Edison Wisley publishers; 2022
4. Bishop, C., M., Pattern Recognition and Machine Learning, Springer, 2006.

Reference Books:

1. Chollet, Francois, Deep learning with Python, New York: Manning Publications; 2018
2. Richard Wilson, Mastering Deep Learning Fundamentals with Python, 2019
3. Cloud Computing – web-based Applications that change the way you work and collaborate Online – Michael Miller. Pearson Education.
4. Artificial Neural Networks, Yeg nanarayana, B., PHI Learning Pvt. Ltd, 2009.
5. Matrix Computations, Golub, G., H., and Van Loan, C., F, JHU Press, 2013.
6. Neural Networks: A Classroom Approach, Satish Kumar, Tata McGraw-Hill Education,

2004

Internal Assessment Pattern

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
L1	35	--
L2	40	--
L3	25	25
L4	--	35
L5	--	25
L6	--	15
Total (%)	100	100

Sample Short and Long Answers questions of Various Cognitive Levels**L1: Remember**

1. Define artificial neuron and perceptron.
2. List applications of neural networks.
3. Define deep learning and DNN.
4. List optimization methods like AdaGrad, RMSProp and Adam.
5. Define convolution and pooling.
6. List features of Convolutional Neural Network AlexNet and ResNet.
7. Define RNN and BPTT.
8. List differences between GRU and LSTM.
9. Define RBM and DBN.
10. List types of Boltzmann Machines.

L2: Understand

1. Explain learning rules in neural networks.
2. Explain overfitting and underfitting.
3. Explain greedy layer-wise training.
4. Explain dropout and batch normalization.
5. Explain working of CNN architecture.
6. Explain residual learning in ResNet.
7. Explain backpropagation through time.
8. Explain bidirectional recurrent neural networks.
9. Explain Deep Belief Networks.
10. Explain convolutional Boltzmann machines.

L3: Apply

1. Apply the perceptron learning algorithm to solve a classification problem.
2. Illustrate the use of backpropagation for training a multilayer neural network.
3. Apply optimization techniques such as AdaGrad, RMSProp and Adam in training deep neural networks.
4. Demonstrate the use of regularization methods to reduce overfitting.
5. Apply convolution operations in image feature extraction.

6. Implement a CNN model for image classification.
7. Apply Backpropagation Through Time (BPTT) in recurrent neural networks.
8. Use GRU/LSTM models for sequence prediction problems.
9. Apply Restricted Boltzmann Machines for feature learning.
10. Demonstrate the application of Deep Belief Networks in pattern recognition.

L4: Analysing

1. Analyze the differences between single layer and multilayer perceptrons.
2. Compare supervised and unsupervised neural network models.
3. Analyze the performance of AdaGrad, RMSProp and Adam optimizers.
4. Differentiate first-order and second-order optimization methods.
5. Analyze the architectures of LeNet, AlexNet, VGG and ResNet.
6. Compare CNN models such as GoogLeNet and DenseNet.
7. Analyze vanishing and exploding gradient problems in RNNs.
8. Compare GRU and LSTM with suitable analysis.
9. Analyze the structure and functioning of RBM and DBM.
10. Distinguish between directed and undirected generative models.

L5: Evaluating

1. Evaluate the performance of backpropagation networks for classification tasks.
2. Justify the significance of multilayer perceptrons in complex problem solving.
3. Evaluate the effectiveness of optimization methods for deep learning.
4. Assess the impact of dropout and batch normalization on model accuracy.
5. Evaluate different CNN architectures for image recognition tasks.
6. Critically assess the advantages of residual networks over traditional CNNs.
7. Evaluate the performance of recurrent neural networks in sequence modeling.
8. Assess the effectiveness of sequence-to-sequence models in NLP tasks.
9. Evaluate the usefulness of generative models in representation learning.
10. Critically assess the role of Boltzmann machines in deep learning

L6: Create

1. Design a neural network for solving a classification problem.
2. Develop a multilayer perceptron model for a real-world application.
3. Create a deep neural network using suitable optimization and regularization techniques.
4. Design an efficient deep learning model for prediction problems.
5. Develop a CNN architecture for image classification applications.
6. Create a custom convolutional neural network model for object recognition.
7. Design an RNN/LSTM model for sequence prediction applications.
8. Develop a sequence-to-sequence model for machine translation.
9. Design a generative model using RBM/DBN for representation learning.
10. Create a deep generative architecture for anomaly detection or data generation.

**Chairperson
Board of Studies (CSE)**

Course Objectives:

1. To provide knowledge on the fundamental concepts, functions, and practices of Human Resource Management and Project Management.
2. To develop understanding of strategic HR practices, talent management, employee performance, and welfare administration in organizations.
3. To familiarize students with project planning, feasibility analysis, scheduling techniques, and project implementation processes.
4. To impart analytical and managerial skills required for effective decision-making in human resource and project environments.
5. To enable students to apply modern HR practices and project management tools for improving organizational and project performance.

Course Outcomes

At the end of the course, students will be able to:

Course Code	Course Outcomes	Mapping with POs and PSOs											Dok
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	
R24HS10.1	Explain the concepts, functions, and strategic importance of Human Resource Management and Project Management in organizations.	3	2	1	-	-	1	2	1	2	2	2	L1, L2
R24HS10.2	Apply human resource planning, recruitment, training, and talent management practices for organizational effectiveness.	2	2	2	1	1	1	2	3	2	3	2	L2, L3
R24HS10.3	Analyze employee performance management systems, compensation practices, and employee welfare measures in contemporary organizations.	2	3	1	2	1	2	3	3	2	2	2	L3, L4
R24HS10.4	Examine project feasibility, project planning, scheduling techniques, and financial appraisal methods used in project management.	3	3	3	2	2	1	1	2	2	3	2	L4, L5
R24HS10.5	Evaluate project implementation, monitoring, stakeholder management, and modern workplace practices for achieving	2	3	3	3	2	2	3	3	3	3	3	L5, L6

	organizational and project success.												
--	-------------------------------------	--	--	--	--	--	--	--	--	--	--	--	--

SYLLABUS

UNIT – I

16 Hours

Strategic Human Resources Management: Human Resource Management – Meaning, Nature, Scope and Importance – Functions of HRM – Evolution of HRM – Strategic Human Resource Management – Aligning HR Strategy with Organizational Strategy – HR Policies and Ethical Practices – Role and Position of HR Department – Emerging Trends in HRM – Diversity, Equity and Inclusion (DEI) – Digital HR Practices – Employee Engagement – Employee Experience Management – Green HRM Practices. **CO's–CO1**

Self-Learning Topics:

Study how strategic HRM practices, employee engagement, and digital HR systems contribute to organizational effectiveness and employee satisfaction.

UNIT – II

14 Hours

Human Resources Planning and Talent Management: Human Resource Planning – Demand and Supply Forecasting – Recruitment and Selection – Modern Sources of Recruitment – Employer Branding – E-Recruitment and Online Hiring Platforms – Selection Tests and Interview Techniques – Employee Onboarding – Training and Development – E-learning and Digital Training Methods – Job Design and Job Analysis – Talent Management – Human Resource Development Concepts – Career Development and Career Counselling. **CO's–CO2**

Self-Learning Topics:

Develop a simple HR demand and supply forecast for an organization and examine the role of digital recruitment and training methods in effective talent acquisition and employee development.

UNIT – III

12 Hours

Performance Management and Employee Welfare: Importance of Performance Appraisal – Traditional and Modern Methods of Performance Appraisal – Continuous Performance Management – 360-Degree Feedback – Compensation Concepts and Principles – Wage and Salary Administration – Incentives, Rewards and Recognition Systems – Employee Retention Strategies – Work-Life Balance – Employee Well-being Programs – Employee Benefits and Flexible Compensation – Workplace Safety – Employee Health and Wellness Programs – Employee Assistance Programs (EAPs) – Managing Remote and Hybrid Workforce Challenges – Workplace Ethics and Emotional Well-being. **CO's–CO3**

Self-Learning Topics:

Research recent trends in performance management, employee wellness programs, and flexible compensation practices and analyze their impact on employee productivity, motivation, and retention.

UNIT – IV

14 Hours

Project Planning and Feasibility Analysis: Basics of Project Management – Concept and Principles of Project Management – Types of Projects – Project Life Cycle – Project Planning and Monitoring – Project Environment Analysis – Project Appraisal and Project Selection – Causes of Delay in Project Commissioning and Remedies to Avoid Overruns – Identification of Investment Opportunities – Sources of Project Ideas – Preliminary Screening of Projects – Market Feasibility – Market Survey – Demand Forecasting Techniques – Resource Allocation – Sales Projections – Technical Feasibility – Production Technology – Plant Capacity – Site Selection and Plant Layout – Legal Feasibility and Basic Legal Provisions. **CO's–CO4**

Self-Learning Topics:

Study the stages of the project life cycle and analyze the importance of market, technical, and legal feasibility studies in successful project planning and implementation.

UNIT – V**10 Hours**

Project Scheduling, Financial Feasibility and Implementation: Programme Evaluation and Review Technique (PERT) – Construction of PERT – Project Duration and Evaluation – Slack and Critical Activities – Critical Path Method (CPM) – Financial Feasibility – Capital Expenditure – Capital Investment Appraisal Techniques – Risk Analysis – Project Integration Management – Forms of Project Organization – Project Implementation – Monitoring and Controlling Process – Stakeholder Management – Project Review and Performance Evaluation – Continuous Improvement in Projects.

CO's–CO5**Self-Learning Topics:**

Examine the application of PERT and CPM techniques, investment appraisal methods, and stakeholder management practices in effective project implementation and control.

Board of Studies: Master of Business Administration

Approved in BOS No: 03

Approved in ACM No: 03

Text Books

1. Human Resource Management, Gary Dessler, Pearson Education.
2. Personnel Management and Human Resource Management, P. Subba Rao, Himalaya Publishing House.
3. Project Management: A Systems Approach to Planning, Scheduling, and Controlling, Harold Kerzner, Wiley India.
4. Projects: Planning, Analysis, Selection, Financing, Implementation and Review, Prasanna Chandra, McGraw Hill Education.
5. Human Resource and Personnel Management, K. Aswathappa, McGraw Hill Education.

Reference Books

1. Strategic Human Resource Management, Jeffrey A. Mello, Cengage Learning.
2. Human Resource Management, C.B. Mamoria and S.V. Gankar, Himalaya Publishing House.
3. Operations and Project Management, K. Nagarajan, New Age International Publishers.
4. Project Management, Vasant Desai, Himalaya Publishing House.
5. Project Management: The Managerial Process, McGraw Hill Education.

Web References:

1. <https://www.shrm.org>
2. <https://www.pmi.org>
3. <https://www.nhrdn.org>
4. <https://hbr.org/topic/human-resource-management>
5. <https://www.managementstudyguide.com>
6. https://www.mindtools.com/pages/main/newMN_PPM.htm
7. <https://www.coursera.org/browse/business/human-resources>
8. <https://www.coursera.org/browse/business/project-management>
9. <https://nptel.ac.in>
10. <https://swayam.gov.in>

Internal Assessment Pattern

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
L1	17	-

L2	17	-
L3	33	30
L4	33	30
L5	-	28
L6	-	14
Total (%)	100	100

Sample Short and Long Answers to Questions of Various Cognitive Levels

PART – A: 2 MARK QUESTIONS

LEVEL – 1 (Remembering)

1. Define Human Resource Management.
2. What is Strategic HRM?
3. State any two functions of HRM.
4. What is employee engagement?
5. Define recruitment.
6. What is job analysis?
7. Define performance appraisal.
8. What is PERT?
9. Define project feasibility study.
10. What is critical path in CPM?

LEVEL – 2 (Understanding)

1. Explain the importance of HR policies.
2. Describe the role of HR department in organizations.
3. Explain the concept of diversity and inclusion.
4. Discuss the need for human resource planning.
5. Explain the purpose of employee onboarding.
6. Describe the importance of training and development.
7. Explain the significance of employee well-being programs.
8. Discuss the stages of project life cycle.
9. Explain market feasibility analysis.
10. Describe the purpose of project appraisal.

LEVEL – 3 (Applying)

1. Illustrate how e-recruitment is used in organizations.
2. Apply HR forecasting techniques for manpower planning.
3. Demonstrate the process of conducting a performance appraisal.
4. Show how incentives improve employee motivation.
5. Apply work-life balance practices in organizations.
6. Illustrate the use of digital HR practices.
7. Apply demand forecasting methods in project planning.
8. Demonstrate the preparation of a simple PERT chart.
9. Apply CPM techniques to identify project duration.
10. Illustrate risk analysis in project management.

LEVEL – 4 (Analyzing)

1. Differentiate between traditional and modern HR practices.
2. Compare recruitment and selection.
3. Analyze the impact of employee engagement on productivity.

4. Distinguish between training and development.
5. Analyze the importance of compensation policies.
6. Differentiate between monetary and non-monetary incentives.
7. Compare PERT and CPM techniques.
8. Analyze the causes of project delays.
9. Differentiate between market feasibility and technical feasibility.
10. Analyze the role of stakeholder management in projects.

LEVEL – 5 (Evaluating)

1. Evaluate the effectiveness of digital HR practices.
2. Assess the role of DEI initiatives in organizations.
3. Evaluate employee retention strategies.
4. Assess the effectiveness of wellness programs.
5. Evaluate the role of grievance handling systems.
6. Assess the importance of workplace ethics.
7. Evaluate investment appraisal techniques.
8. Assess the importance of project monitoring.
9. Evaluate financial feasibility analysis in projects.
10. Assess the significance of continuous improvement in projects.

LEVEL – 6 (Creating)

1. Design an HR policy for employee welfare.
2. Develop a recruitment plan for a startup company.
3. Prepare a training program for employees.
4. Design a performance appraisal system.
5. Develop an employee engagement strategy.
6. Create a compensation structure for employees.
7. Prepare a project feasibility report.
8. Design a project implementation plan.
9. Construct a simple PERT/CPM network for a project.
10. Develop strategies for successful project completion.

PART – B: ESSAY QUESTIONS

LEVEL – 1 (Remembering)

1. Define HRM and explain its functions.
2. Explain the concept and objectives of Strategic HRM.
3. Describe the stages of the project life cycle.
4. Explain the meaning and importance of project management.
5. Define performance appraisal and explain its objectives.
6. Explain the concept of employee engagement.
7. Define PERT and CPM techniques.
8. Explain the importance of training and development.
9. Define compensation and explain its components.
10. Explain the meaning and objectives of employee welfare.

LEVEL – 2 (Understanding)

1. Explain the role of HR policies in organizational effectiveness.
2. Discuss the significance of diversity, equity, and inclusion in organizations.
3. Explain the process of recruitment and selection.
4. Describe various methods of performance appraisal.

5. Explain the importance of employee retention strategies.
6. Discuss the role of work-life balance in employee satisfaction.
7. Explain the concept of market feasibility analysis.
8. Describe the importance of project feasibility studies.
9. Explain the process of project planning and monitoring.
10. Discuss the importance of workplace ethics and employee well-being.

LEVEL – 3 (Applying)

1. Illustrate how digital HR practices improve organizational efficiency.
2. Apply HR planning techniques for manpower forecasting.
3. Demonstrate how employee onboarding can improve employee performance.
4. Apply compensation and reward systems to motivate employees.
5. Illustrate the role of employee engagement practices in organizations.
6. Apply PERT techniques in project scheduling.
7. Demonstrate the use of CPM for identifying critical activities.
8. Apply market survey methods for project feasibility analysis.
9. Illustrate the use of investment appraisal techniques in projects.
10. Apply stakeholder management techniques in project implementation.

LEVEL – 4 (Analyzing)

1. Analyze the impact of strategic HRM on organizational performance.
2. Differentiate between traditional and modern methods of recruitment.
3. Analyze the role of performance appraisal in employee development.
4. Compare monetary and non-monetary compensation systems.
5. Analyze the impact of employee wellness programs on productivity.
6. Differentiate between PERT and CPM techniques with examples.
7. Analyze the causes of project delays and suggest remedies.
8. Compare market feasibility and financial feasibility analyses.
9. Analyze the role of project monitoring and control.
10. Examine the impact of workplace stress on employee performance.

LEVEL – 5 (Evaluating)

1. Evaluate the effectiveness of digital HR practices in modern organizations.
2. Assess the role of employee engagement in organizational success.
3. Evaluate the importance of DEI practices in HRM.
4. Assess the effectiveness of performance appraisal systems.
5. Evaluate the role of compensation management in employee retention.
6. Assess the importance of project appraisal techniques.
7. Evaluate different investment appraisal methods.
8. Assess the role of stakeholder management in project success.
9. Evaluate the effectiveness of project monitoring systems.
10. Assess the role of workplace ethics in maintaining industrial harmony.

LEVEL – 6 (Creating)

1. Design a strategic HR plan for a manufacturing organization.
2. Develop a recruitment and selection strategy for IT companies.
3. Prepare a comprehensive employee training and development program.
4. Design a modern performance management system.
5. Create an employee wellness and engagement program.
6. Develop a flexible compensation policy for employees.

7. Prepare a detailed project feasibility report for a new business venture.
8. Design a project implementation and monitoring framework.
9. Construct a PERT/CPM network for a construction project.
10. Develop strategies for effective project control and continuous improvement.

**Chairperson
Board of Studies (MBA)**

Course Objectives:

1. Understand the fundamentals of blockchain technology, consensus mechanisms.
2. Analyze the concepts of decentralization, smart contracts and platforms supporting blockchain-based ecosystems.
3. Gain knowledge of Bitcoin architecture, including digital keys, transactions.
4. Understand the Ethereum platform, including smart contracts.
5. Explore Hyperledger frameworks and analyze challenges and opportunities in block chain.

Course Outcomes

At the end of the course, students will be able to:

Course Code	Course Outcomes (COs)	Mapping with POs and PSOs														Dok
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2		
R24CSPE 11.1	Define and explain the fundamentals of Block chain.	3	2	1	–	2	–	–	–	–	–	1	1	-	L1, L2	
R24CSPE 11.2	Understand decentralization and the role of Block chain in it.	3	3	2	1	3	1	–	1	1	–	2	-	1	L2, L3	
R24CSPE 11.3	Understand and analyze Bitcioin Crypto currency and underlying Block chain network.	3	2	2	1	3	–	–	–	1	–	2	-	-	L3, L4	
R24CSPE 11.4	Understand Ethereum currency and platform, and develop applications using Solidity.	3	2	3	1	3	1	–	1	1	1	2	1	-	L4, L5	
R24CSPE 11.5	Understand Hyper ledger project and its components; critically analyze the challenges and future opportunities in Block chain technology.	3	3	2	2	3	2	1	1	1	–	3	-	1	L5, L6	

SYLLABUS**UNIT I:****15 Hours**

Introduction: History and basics, Types of Blockchain, Consensus, CAP Theorem.

Cryptographic Hash Functions: Properties of hash functions, Secure Hash Algorithm, Merkle trees, Patricia trees.

CO's–CO1

Self-Learning Topics: Distributed Systems Basics, Digital Signatures and Public Key Cryptography

UNIT II:**13 Hours**

Decentralization: Decentralization using Blockchain, Methods of decentralization,

decentralization framework, Blockchain and full ecosystem decentralization, Smart contracts, Decentralized Organizations, Platforms for decentralization. **CO's–CO2**

Self-Learning Topics: Tokenization and Digital Assets, Interoperability in Block chain

UNIT III: **13 Hours**

Bitcoin: Introduction to Bitcoin, Digital keys and addresses, Transactions, Block chain, The Bitcoin network, Bitcoin payments, Bitcoin Clients and APIs, Alternatives to Proof of Work, Bitcoin limitations. **CO's–CO3**

Self-Learning Topics: Structure of a Bitcoin Block, Security and Attacks in Bitcoin

UNIT IV: **14 Hours**

Ethereum: Smart Contracts, Introduction to Ethereum, The Ethereum network, Components of the Ethereum ecosystem, Blocks and Blockchain, Fee schedule, Ethereum Development Environment, Solidity. **COs–CO4**

Self-Learning Topics: Account Types in Ethereum, Smart Contract Security Basics

UNIT V: **14 Hours**

Hyperledger: Introduction, Hyperledger Projects, Protocol, Architecture, Hyperledger Fabric, Sawtooth Lake, Corda.

Challenges and Opportunities: Scalability, Privacy, Blockchain for IoT, Emerging trends

CO's–CO5

Self-Learning Topics: Consensus Mechanisms in Hyperledger, Scalability Solutions in Blockchain

Board of Studies : Computer Science and Engineering

Approved in BoS No : 03

Approved in ACM No: 03

Text Book:

1. Mastering Blockchain, Imran Bashir, Second Edition, Packt Publishing.

Reference Books:

1. Mastering Bitcoin: Unlocking Digital Cryptocurrencies, Andreas Antonopoulos, O'Reilly.
2. Blockchain Blueprint for a New Economy, Melanie Swan, O'Reilly.
3. Mastering Bitcoin: Programming the Open Blockchain, Antonopoulos, Andreas M. O'Reilly.
4. Blockchain Technology: Cryptocurrency and Applications, S. Shukla, M. Dhawan, S. Sharma, S. Venkatesan, Oxford University Press.

Internal Assessment Pattern

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
L1	35	--
L2	40	--
L3	25	25
L4	--	35
L5	--	25
L6	--	15
Total (%)	100	100

Sample Short and Long Answers questions of Various Cognitive Levels

L1: Remember

1. Define blockchain. Explain its history and basic concepts.
2. List different types of blockchains and their features.
3. Define cryptographic hash function. State its properties.
4. Define decentralization. List its key characteristics in blockchain systems.
5. What are smart contracts? State their features.
6. Define Bitcoin. Explain its basic features.
7. What are digital keys and Bitcoin addresses?
8. Define Ethereum. List its key features.
9. What are smart contracts? State their characteristics in Ethereum.
10. Define Ethereum and explain its basic features and objectives.
11. What are smart contracts? List their characteristics in Ethereum.

L2: Understand

1. Explain the working of consensus mechanisms in blockchain.
2. Describe CAP theorem and its relevance to blockchain systems.
3. Explain Secure Hash Algorithm (SHA) with suitable example.
4. Explain decentralization using blockchain with suitable examples.
5. Describe different methods of decentralization.
6. Explain the process of a Bitcoin transaction.
7. Describe the structure of the Bitcoin network with a diagram.
8. Explain the architecture of the Ethereum network with a diagram.
9. Describe the working of smart contracts in Ethereum.
10. Describe the lifecycle of a smart contract in Ethereum.
11. Explain the fee (gas) mechanism and its role in Ethereum transactions.

L3: Apply

1. Illustrate how hash functions are used to ensure data integrity in blockchain.
2. Demonstrate the process of building a Merkle tree for a given set of transactions.
3. Apply CAP theorem to analyze a blockchain system scenario.
4. Apply decentralization concepts to design a simple decentralized application model.
5. Demonstrate the working of a decentralized organization using blockchain.
6. Illustrate how Bitcoin transactions are validated and added to the blockchain.
7. Apply the concept of digital keys to explain how secure transactions occur in Bitcoin.
8. Illustrate how a smart contract is deployed and executed in Ethereum.
9. Apply Solidity concepts to write a simple smart contract (outline).
10. Illustrate the deployment and execution of a smart contract in Ethereum.
11. Write the structure (syntax) of a simple Solidity smart contract and explain it.

L4: Analysing

1. Compare public, private, and consortium blockchains.
2. Analyze the properties of cryptographic hash functions and their importance in blockchain security.
3. Differentiate between Merkle trees and Patricia trees.
4. Analyze the decentralization framework and its components.
5. Differentiate between centralized, decentralized, and distributed systems.
6. Analyze the working of the Bitcoin blockchain and its components.

7. Differentiate between Proof of Work and its alternatives.
8. Analyze the structure of Ethereum blocks and blockchain.
9. Differentiate between Ethereum and Bitcoin in terms of functionality.
10. Analyze the structure of Ethereum blocks and how they differ from Bitcoin blocks.
11. Differentiate between Ethereum and Bitcoin with respect to architecture and functionality.

L5: Evaluating

1. Evaluate the effectiveness of different consensus mechanisms in blockchain systems.
2. Assess the limitations of CAP theorem in real-world distributed systems.
3. Justify the use of hash functions in securing blockchain transactions.
4. Evaluate the advantages and limitations of decentralization in blockchain.
5. Assess the effectiveness of smart contracts in automating processes.
6. Evaluate the advantages and limitations of Bitcoin as a digital currency.
7. Assess the effectiveness of Proof of Work in securing the Bitcoin network.
8. Evaluate the advantages of Ethereum over traditional systems.
9. Assess the efficiency of the gas mechanism in Ethereum.
10. Evaluate the advantages and limitations of Ethereum platform.
11. Assess the effectiveness of gas mechanism in optimizing resource usage.

L6: Create

1. Design a simple blockchain model illustrating the use of hash functions and linking of blocks.
2. Develop a conceptual model to demonstrate how digital signatures ensure authentication in blockchain.
3. Construct a Merkle tree for a given dataset and explain its verification process.
4. Design a decentralized application (DAPP) using blockchain concepts.
5. Develop a conceptual model of a decentralized organization (DAO).
6. Design a conceptual model of a Bitcoin transaction system.
7. Develop a simplified architecture of the Bitcoin network.
8. Design a simple decentralized application (DAPP) using Ethereum concepts.
9. Develop a smart contract for a real-world application (e.g., voting system).
10. Design a smart contract for a real-world application such as voting or supply chain management.
11. Develop a conceptual architecture of a decentralized application (DAPP) using Ethereum.

Chairperson
Board of Studies (CSE)

R24CSPE12**BIG DATA ANALYTICS****3 0 0 3**

(Professional Elective-III)

Course Objectives:

1. To understand Big Data concepts and platforms.
2. To learn stream processing and real-time analytics.
3. To study Hadoop and distributed data processing.
4. To use Big Data frameworks like Hive, Pig, and Spark.
5. To apply predictive analytics and visualization techniques.

Course Outcomes:

At the end of the course, students will be able to:

Course Code	Course Outcomes (COs)	Mapping with POs and PSOs														Dok
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2		
R24CSPE 12.1	Understand Big Data fundamentals, characteristics, architectures, and challenges.	3	2	-	-	1	-	-	-	-	-	2	3	2	L1, L2	
R24CSPE 12.2	Analyze and process streaming data using real-time analytics and stream processing techniques.	3	3	2	-	2	-	-	2	-	-	2	3	2	L2, L3	
R24CSPE 12.3	Apply Hadoop ecosystem tools like HDFS and MapReduce for large-scale distributed processing.	3	2	3	-	2	-	-	-	-	-	2	3	3	L3, L4	
R24CSPE 12.4	Utilize Big Data frameworks such as Pig, Hive, HBase, Spark, and PySpark for data handling.	3	2	2	2	3	-	-	-	-	-	2	3	2	L4, L5	
R24CSPE 12.5	Develop predictive analytics models and visualize data using machine learning and visualization tools.	3	3	3	3	3	2	-	-	2	3	3	3	2	L5, L6	

SYLLABUS**UNIT I:****15 Hours**

Introduction to Big Data: Introduction to Big Data Platform, Challenges of Conventional Systems, Intelligent Data Analysis, Nature of Data, Analytic Processes and Tools, , Cloud-based Big Data Platforms, Introduction to NoSQL Databases, Data Governance and Privacy. **CO's-CO1**

Self-Learning Topics: Blockchain for Big Data, Green Computing in Data Centers.

UNIT II:**13 Hours**

Mining Data Streams: Introduction to Streams Concepts, Stream Data Model and Architecture, Stream Computing, Sampling Data in a Stream, Filtering Streams, Counting Distinct Elements in a Stream, Estimating Moments, , Decaying Window, Real-Time Analytics Platform (RTAP) Applications, Introduction to Apache Kafka, Real-Time Data Pipelines, Case Studies – Real-Time

Sentiment Analysis, Stock Market Predictions.

CO's–CO2

Self-Learning Topics: Edge Analytics, Smart City Data Applications.

UNIT III:

13 Hours

Hadoop: History of Hadoop, Hadoop Distributed File System (HDFS), Components of Hadoop, Analyzing the Data with Hadoop, Scaling Out, Hadoop Streaming, Design of HDFS, , Developing a MapReduce Application, How MapReduce Works, Anatomy of a MapReduce Job Run, Failures, Job Scheduling, Shuffle and Sort, Task Execution, MapReduce Types and Formats, Hadoop Environment, YARN Architecture, Hadoop Security Basics.

CO's–CO3

Self-Learning Topics: Container-based Big Data Deployment, Hadoop in Cloud Environments.

UNIT IV:

14 Hours

Frameworks: Applications on Big Data Using Pig and Hive, Hive Services, HiveQL, Querying Data in Hive, Fundamentals of HBase and ZooKeeper, Introduction to Apache Spark, Spark SQL Basics, Introduction to PySpark.

COs–CO4

Self-Learning Topics: Graph Databases, Introduction to Apache Flink.

UNIT V:

13 Hours

Predictive Analytics and Visualizations: Predictive Analytics, Simple Linear Regression, Multiple Linear Regression, Interpretation of Regression Coefficients, Visualizations, Visual Data Analysis Techniques, , Systems and Applications, Introduction to Machine Learning, Classification and Clustering Basics, Interactive Dashboards using Tableau/Power BI.

CO's-CO5

Self-Learning Topics: Auto ML Concepts, Augmented Analytics.

Board of Studies : Computer Science and Engineering

Approved in BoS No : 03

Approved in ACM No: 03

Text Books:

1. Tom White, “Hadoop: The Definitive Guide”, Third Edition, O'reilly Media, Fourth Edition, 2015.
2. Chris Eaton, Dirk DeRoos, Tom Deutsch, George Lapis, Paul Zikopoulos, “Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data”, McGrawHill Publishing, 2012.
3. Anand Rajaraman and Jeffrey David Ullman, “Mining of Massive Datasets”, CUP, 2012

Reference Books:

1. Bill Franks, “Taming the Big Data Tidal Wave: Finding Opportunities in Huge Data Streams with Advanced Analytics”, John Wiley & sons, 2012.
2. Paul Zikopoulos, Dirk deRoos, Krishnan Parasuraman, Thomas Deutsch, James Giles, David Corrigan, “Harness the Power of Big Data: The IBM Big Data Platform”, Tata McGraw Hill Publications, 2012.
3. Arshdeep Bahga and Vijay Madisetti, “Big Data Science & Analytics: A Hands On Approach”, VPT, 2016.
4. Bart Baesens, “Analytics in a Big Data World: The Essential Guide to Data Science and its Applications (WILEY Big Data Series)”, John Wiley & Sons, 2014.

Internal Assessment Pattern

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
L1	35	--

L2	40	--
L3	25	25
L4	--	35
L5	--	25
L6	--	15
Total (%)	100	100

Sample Short and Long Answers questions of Various Cognitive Levels

L1 : Remembering

1. Define Big Data.
2. What are the 5Vs of Big Data?
3. Expand HDFS and YARN.
4. Define NoSQL database.
5. What is stream computing?
6. Define MapReduce.
7. What is Apache Kafka?
8. Define HiveQL.
9. What is predictive analytics?
10. Define clustering.
11. Explain the components of Hadoop.
12. Describe the architecture of HDFS with a neat diagram.
13. Explain the features of Apache Spark.

L2: Understanding

1. Explain the challenges of conventional systems in handling Big Data.
2. Describe the importance of data governance.
3. Explain stream data architecture.
4. Describe Hadoop streaming.
5. Explain Spark SQL basics.
6. Discuss the role of Zookeeper in distributed systems.
7. Explain simple linear regression.
8. Describe classification techniques in machine learning.
9. Explain the working of MapReduce with neat architecture.
10. Discuss cloud-based Big Data platforms and their applications.
11. Explain Hive architecture and querying process.
12. Describe predictive analytics and visualization techniques.

L3: Applying

1. Write a HiveQL query to retrieve filtered records.
2. Demonstrate stream filtering with an example.
3. Apply Kafka in real-time analytics applications.
4. Illustrate Hadoop streaming with an example.
5. Use PySpark for processing large datasets.
6. Apply regression analysis for sales prediction.
7. Develop a simple MapReduce word-count application.
8. Demonstrate real-time sentiment analysis using streaming data.
9. Apply Spark SQL for structured data processing.

10. Build an interactive dashboard using Tableau or Power BI.

L4: Analyzing

1. Compare traditional databases and NoSQL databases.
2. Differentiate batch processing and stream processing.
3. Compare Pig and Hive frameworks.
4. Analyze regression coefficients in predictive analytics.
5. Compare classification and clustering techniques.
6. Analyze the architecture and design of HDFS.
7. Compare Hadoop MapReduce and Apache Spark performance.
8. Analyze the phases involved in MapReduce execution.
9. Compare scaling up and scaling out approaches in Big Data systems.
10. Analyze privacy and security challenges in Big Data environments.

L5: Evaluating

1. Evaluate the effectiveness of NoSQL databases.
2. Assess the importance of real-time analytics platforms.
3. Evaluate Hadoop security mechanisms.
4. Assess the usefulness of visualization tools in analytics.
5. Evaluate machine learning techniques for Big Data applications.
6. Evaluate cloud-based Big Data platforms over conventional systems.
7. Assess Spark as an alternative to Hadoop MapReduce.
8. Evaluate the role of predictive analytics in business forecasting.
9. Assess the impact of edge analytics in smart city applications.
10. Evaluate green computing approaches in Big Data centres.

L6: Creating

1. Design a simple data pipeline using Apache Kafka.
2. Design a visualization dashboard for student performance analytics.
3. Design a Big Data architecture for healthcare analytics.
4. Create a real-time stock market prediction system using streaming analytics.
5. Develop a Hadoop-based log analysis application.
6. Propose a smart city analytics framework using edge analytics.

**Chairperson
Board of Studies (CSE)**

Course Objectives:

1. Introduce to some of the problems and solutions of NLP and their relation to linguistics and statistics.
2. To understand linguistic phenomena and learn to model them with formal grammars.
3. To understand and carry out proper experimental methodology for training and evaluating empirical NLP systems.
4. To learn how to manipulate probabilities, construct statistical models over string sand trees
5. To design, implement, and analyze NLP algorithms. Able to design different language modeling.

Course Outcomes:

At the end of the course, students will be able to:

Course Code	Course Outcomes (COs)	Mapping with POs and PSOs														Dok
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2		
R24CSPE 13.1	Show sensitivity to linguistic phenomena and an ability to model them with formal grammars	3	2	1	-	1	-	-	-	-	-	2	3	1	L1, L2	
R24CSPE 13.2	Understand and carry out proper experimental methodology for training and evaluating empirical NLP systems.	3	3	2	2	2	-	-	-	1	-	2	3	2	L2, L3	
R24CSPE 13.3	Able to manipulate probabilities, construct statistical models over strings and trees	3	3	2	2	2	-	-	-	-	-	2	3	2	L3, L4	
R24CSPE 13.4	Analyze semantic representations and meaning extraction techniques	3	3	2	2	2	-	-	-	1	-	2	3	2	L4, L5	
R24CSPE 13.5	Able to design, implement, and analyze NLP algorithms. Able to design different language modeling Techniques	3	3	3	2	2	1	-	1	1	-	3	3	3	L5, L6	

SYLLABUS**UNIT I:****15 Hours**

Natural Language processing (NLP): Introduction, Applications or Use cases of NLP, Components of NLP, Steps in NLP, Finding the Structure of Words: Words and Their Components, Lexemes, Morphemes, Morphology, Problems in morphological processing,

Typology, Morphological Typology

CO's–CO1

Self-Learning Topics: Tokenization techniques in NLP, NLTK library, Real-world NLP applications

UNIT II:**15 Hours**

Syntax Analysis: Parsing Natural Language, Tree banks: A Data-Driven Approach to Syntax, **Representation of Syntactic Structure:** Syntax Analysis using Dependency Graph, Syntax Analysis using Phrase Structure Trees,

Parsing Algorithms: Shift Reduce Parsing, Hyper Graphs and Chart Parsing (CYK Parsing), **Models for ambiguity Resolution in Parsing:** Probabilistic Context Free Grammar, Generative Models, Discriminative models for Parsing.

CO's–CO2

Self-Learning Topics: SpaCy parsing toolkit, Transition-based vs graph-based parsing

UNIT III:**13 Hours**

Language Modeling: Introduction, N-Gram Models, Language Model Evaluation, Parameter Estimation, Language Model Adaptation, Types of Language Models, Language-Specific Modelling Problems.

CO's–CO3

Self-Learning Topics: Smoothing techniques (Laplace, Good-Turing), Word embeddings

UNIT IV:**15 Hours**

Semantic Parsing: Introduction, Semantic Interpretation, System Paradigms, Word Sense Systems, Software.

COs–CO4

Self-Learning Topics: Named Entity Recognition (NER), Word Sense Disambiguation (WSD)

UNIT V:**14 Hours**

Predicate-Argument Structure: Meaning Representation Systems, Software.

Discourse Processing: Cohesion, Reference Resolution, Discourse Cohesion and Structure

Self-Learning Topics: Text summarization techniques, Chatbot design basics

CO's–CO5

Board of Studies : Computer Science and Engineering

Approved in BoS No : 03

Approved in ACM No: 03

Text Book:

1. Multilingual Natural Language Processing Applications: From Theory to Practice–Daniel M.Bikel and Imed Zitouni, Pearson Publication.
2. Speech and Natural Language Processing–Daniel Jurafsky & James H Martin, Pearson Publications.

Reference Books:

1. Natural Language Processing and Information Retrieval: Tanvier Siddiqui, U.S.Tiwary

Internal Assessment Pattern

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
L1	35	--
L2	40	--
L3	25	25

L4	--	35
L5	--	25
L6	--	15
Total (%)	100	100

Sample Short and Long Answers questions of Various Cognitive Levels

L1: Remember

1. Define Natural Language Processing (NLP).
2. List the components of NLP.
3. What is morphology?
4. Define lexeme and morpheme.
5. What is parsing?
6. Define context-free grammar (CFG).
7. What is an N-gram model?
8. Define language model.
9. What is semantic analysis?
10. Define discourse processing.
11. What is coreference resolution?

L2: Understand

1. Explain the steps involved in NLP.
2. Describe morphological processing with examples.
3. Explain dependency parsing.
4. Describe phrase structure trees.
5. Explain ambiguity in natural language.
6. Describe probabilistic context-free grammar (PCFG).
7. Explain N-gram language models.
8. Describe parameter estimation in NLP.
9. Explain semantic interpretation.
10. Describe discourse cohesion.
11. Explain reference resolution with example.

L3: Apply

1. Apply tokenization to a given sentence.
2. Perform stemming and lemmatization on sample words.
3. Construct a parse tree for a given sentence.
4. Apply CYK parsing for a simple grammar.
5. Use PCFG to resolve ambiguity in a sentence.
6. Compute probability of a sentence using bigram model.
7. Apply smoothing techniques for language models.
8. Perform part-of-speech tagging on a sentence.
9. Apply semantic role labeling to a sentence.
10. Perform named entity recognition on text.
11. Apply coreference resolution for a paragraph.

L4: Analysing

1. Compare rule-based and statistical NLP approaches.
2. Analyze challenges in morphological processing.

3. Compare dependency and constituency parsing.
4. Analyze ambiguity problems in parsing.
5. Compare generative and discriminative models.
6. Analyze performance of different parsing algorithms.
7. Compare unigram, bigram, and trigram models.
8. Analyze limitations of N-gram models.
9. Compare semantic and syntactic analysis.
10. Analyze issues in word sense disambiguation.
11. Compare discourse processing techniques.

L5: Evaluating

1. Evaluate different NLP approaches for real-world applications.
2. Assess effectiveness of morphological analysis techniques.
3. Evaluate parsing algorithms for accuracy and efficiency.
4. Judge suitability of PCFG in NLP tasks.
5. Evaluate language models based on perplexity.
6. Assess parameter estimation techniques.
7. Evaluate semantic analysis methods.
8. Judge effectiveness of word sense disambiguation methods.
9. Evaluate discourse processing models.
10. Assess different NLP tools (NLTK vs SpaCy).
11. Evaluate challenges in building NLP systems.

L6: Create

1. Design an NLP pipeline for text processing.
2. Develop a morphological analyzer for a language.
3. Construct a grammar for a simple language.
4. Design a parsing algorithm for sentence analysis.
5. Create a probabilistic language model for text generation.
6. Develop a POS tagging system.
7. Design a semantic analysis system.
8. Build a named entity recognition system.
9. Develop a chatbot using NLP techniques.
10. Design a sentiment analysis system.
11. Create a text summarization model.

Chairperson
Board of Studies (CSE)

R24CSPE16**DISTRIBUTED SYSTEMS****3 0 0 3**

(Computer Science and Engineering)

Course Objectives:

1. To understand the foundations of distributed systems.
2. To learn issues related to clock Synchronization, the need for global state and remote invocation in distributed systems.
3. To learn distributed mutual exclusion and deadlock detection algorithms.

Course Outcomes

At the end of the course, students will be able to:

Course Code	Course Outcomes (COs)	Mapping with POs and PSOs														Dok
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2		
R24CSPE 16 .1	Understand foundations and issues of distributed systems.	3	2	-	-	-	-	-	-	-	-	2	3	2	L1, L2	
R24CSPE 16 .2	Illustrate synchronization, global state, and remote invocation	3	3	2	-	2	-	-	-	-	-	2	3	2	L2, L3	
R24CSPE 16 .3	Develop mutual exclusion and deadlock detection algorithms.	3	3	3	2	2	-	-	-	-	-	2	3	2	L3, L4	
R24CSPE 16 .4	Apply P2P systems, distributed shared memory, and security	3	2	3	2	3	-	-	1	-	-	2	3	3	L4, L5	
R24CSPE 16 .5	Analyze transactions, consensus, and fault tolerance	3	3	2	3	3	-	-	-	-	-	2	3	3	L5, L6	

SYLLABUS**UNIT – I****15 Hours**

Introduction to Distributed Systems, Examples of Distributed Systems, Resource Sharing, Challenges in Distributed Systems, Relation to Computer System Components, Motivation, Message Passing vs Shared Memory Systems, Primitives for Distributed Communication, Synchronous vs Asynchronous Execution, Distributed Program Model, Model of Distributed Executions, Communication Network Models, Global State, Cuts, Logical Clocks Framework, Scalar Time, Vector Time, Physical Clock Synchronization, NTP

CO's-CO1

Self-Learning Topics: Relation to Parallel Systems, Process Communication Models, Past and Future Cones of Events

UNIT – II**13 Hours**

Message Ordering Paradigms, Asynchronous Execution with Synchronous Communication, Program Order in Distributed Systems, Group Communication, Causal Ordering, Total Ordering, Global State Concepts, Snapshot Recording Basics, FIFO Channel Snapshots, Remote Invocation Introduction, RMI Basics, RPC Concepts, Events and Notifications

CO's-CO2

Self-Learning Topics: Detailed RMI Implementation, Advanced Snapshot Algorithms

UNIT – III

13 Hours

Distributed Mutual Exclusion Basics, Lamport's Algorithm, Ricart-Agrawala Algorithm, Maekawa's Algorithm, Suzuki-Kasami Algorithm, Deadlock Detection Basics, Deadlock Models, Knapp's Classification, Single Resource Model, AND Model, OR Model

CO's-CO3

Self-Learning Topics: Performance Comparison of Mutual Exclusion Algorithms

UNIT – IV

12 Hours

Peer-to-Peer Computing Basics, Overlay Networks, Chord DHT, Content Addressable Networks, Distributed Shared Memory Concepts, Memory Consistency Models, Shared Memory Mutual Exclusion, Security Basics, Cryptographic Algorithms, Digital Signatures

CO's-CO4

Self-Learning Topics: Tapestry Architecture, Practical Cryptography Applications

UNIT – V

13 Hours

Distributed Transactions Basics, Flat Transactions, Nested Transactions, Atomic Commit Protocols, Concurrency Control, Checkpointing Basics, Failure Recovery Issues, Checkpoint-Based Recovery, Log-Based Recovery, Coordinated Checkpointing, Consensus Problem, Agreement in Failure-Free Systems

CO's-CO5

Self-Learning Topics: Advanced Consensus Algorithms, Fault Tolerance Enhancements

Board of Studies : Computer Science and Engineering

Approved in BoS No : 03

Approved in ACM No: 03

Text Books:

1. Distributed Systems: Principles and Paradigms – Andrew S. Tanenbaum & Maarten Van Steen
2. Distributed Systems: Concepts and Design – George Coulouris et al.
3. Distributed Computing: Principles and Applications – M. L. Liu
4. Distributed Systems – K. S. Easwarkumar

Reference Books:

1. Distributed Computing – Sunita Mahajan & Seema Shah
2. Parallel and Distributed Systems – Arun Kulkarni et al.
3. Distributed Systems – Ashish Khanna et al.
4. Building Distributed Systems – Ranjit Aneesh

Web References:

1. [NPTEL Distributed Systems Course \(IIT Patna\)](#)
2. [NPTEL Cloud Computing & Distributed Systems Course](#)
3. [Distributed Systems Tutorial \(GeeksforGeeks\)](#)
4. [MIT 6.824 Distributed Systems Course](#)

Internal Assessment Pattern:

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
L1	35	--
L2	40	--
L3	25	25
L4	--	35

L5	--	25
L6	--	15
Total (%)	100	100

Sample Sort and Long Answer Questions of Various Cognitive Levels

L1 (Remember)

1. What is a Distributed System?
2. What are examples of Distributed Systems?
3. What is resource sharing?
4. What are the challenges in Distributed Systems?
5. What is message passing and shared memory system?
6. What are communication primitives?
7. What is synchronous and asynchronous execution?
8. What is a logical clock?
9. What is vector time?
10. What is NTP (Network Time Protocol)

L2 (Understand)

1. How do Distributed Systems operate and what is their motivation?
2. How are computing environments and communication models described?
3. How does message passing differ from shared memory systems?
4. How does the distributed program model work?
5. How are global state, cuts, and logical clocks explained?
6. How do scalar time and vector time work?
7. How does physical clock synchronization use NTP work?
8. How do message ordering paradigm's function?
9. How do RPC and RMI work?
10. How does snapshot recording capture global state?

L3 (Apply)

1. How do you apply Lamport logical clock algorithm?
2. How do you calculate vector clocks?
3. How do you apply causal and total message ordering?
4. How do you solve snapshot recording problems?
5. How do you apply RPC/RMI in distributed applications?
6. How do you demonstrate distributed mutual exclusion?
7. How do you apply Lamport's mutual exclusion algorithm?
8. How do you solve Ricart-Agrawala algorithm problems?
9. How do you apply deadlock detection models?
10. How do you demonstrate group communication?

L4 (Analyze)

1. How do synchronous and asynchronous systems compare?
2. How are communication network models analyzed?
3. How do scalar clocks differ from vector clocks?
4. How is global state and consistent cuts analyzed?
5. How do message ordering techniques compare?

6. How do mutual exclusion algorithms compare?
7. How are deadlock models (AND, OR, Single resource) analyzed?
8. How are snapshot algorithms analyzed?
9. How do RPC and RMI compare?
10. How is distributed shared memory analyzed?

L5 (Evaluate)

1. How do you evaluate design challenges in distributed systems?
2. How do you assess clock synchronization techniques?
3. How do you evaluate message ordering protocols?
4. How do you justify the choice of mutual exclusion algorithms?
5. How do you evaluate deadlock detection techniques?
6. How do you assess distributed shared memory consistency models?
7. How do you evaluate peer-to-peer systems like Chord and CAN?
8. How do you justify cryptographic techniques in distributed systems?
9. How do you evaluate transaction management techniques?
10. How do you assess checkpointing and recovery mechanisms?

L6 (Create)

1. How would you design distributed system architecture?
2. How would you improve communication protocols?
3. How would you design a logical clock or synchronization mechanism?
4. How would you develop a distributed mutual exclusion solution?
5. How would you design a deadlock-free distributed system?
6. How would you create a distributed shared memory system?
7. How would you design a secure distributed system using cryptography?
8. How would you develop a distributed transaction system?
9. How would you propose a check pointing and recovery mechanism?
10. How would you design a consensus algorithm?

**Chairperson
Board of Studies (CSE)**

R24CSPE17

SOFT COMPUTING

3 0 0 3

(Professional Elective-IV)

Course Objectives:

1. Understand the fundamental concepts of soft computing techniques such as neural networks, fuzzy logic, genetic algorithms, and hybrid systems.
2. Analyze different neural network architectures and learning algorithms for solving complex problems.
3. Apply fuzzy logic principles to model uncertainty and develop intelligent decision-making systems.
4. Understand and implement genetic algorithms for optimization and search problems.
5. Design and evaluate hybrid soft computing systems for real-world applications.

Course Outcomes:

At the end of the course, students will be able to:

Course Code	Course Outcomes (COs)	Mapping with POs and PSOs														Dok
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2		
R24CSPE 17.1	Understand fundamentals of soft computing and distinguish it from hard computing.	3	2	1	1	1	-	-	-	-	-	-	2	1	L1 L2	
R24CSPE 17.2	Analyze and apply neural network models for problem solving.	3	3	2	2	2	-	-	-	-	-	-	3	2	L2 L3	
R24CSPE 17.3	Apply fuzzy logic principles to design intelligent systems.	3	3	3	2	2	-	-	-	-	-	-	3	2	L3 L4	
R24CSPE 17.4	Understand and implement genetic algorithms for optimization problems.	3	3	3	2	3	-	-	-	-	-	-	3	3	L4 L5	
R24CSPE 17.5	Analyze and design hybrid soft computing systems for real-world applications.	3	3	3	3	3	1	1	-	1	1	1	3	3	L5 L6	

SYLLABUS**UNIT I:****15 Hours**

Basic elements of Soft Computing: Difference between Hard Computing & Soft Computing - Introduction to Neural Networks - Fuzzy Logic - Genetic Algorithms - Hybrid Systems.

CO's-CO1

Self-Learning Topics: Need for soft computing in modern application, Components of soft computing.

UNIT II:**14 Hours**

Neural Networks: Neural Networks: What is Neural Network, Basic Models of Neural Networks - McCulloch-Pitts Neuron - Hebb Network, supervised learning network: perceptron networks - adaptive linear neuron, multiple adaptive linear neuron, BPN, RBF,TDNN- associative memory

network: auto-associative memory network, hetero-associative memory network, BAM, hopfield networks, iterative auto associative memory network & iterative associative memory network – unsupervised learning networks: Kohonen self organizing feature maps. **CO's–CO2**

Self-Learning Topics: Mathematical model of a neuron, working principle of McCulloch-Pitts neuron, Bidirectional Associative Memory (BAM)

UNIT III:

13 Hours

Fuzzy Logic System: Introduction to fuzzy logic, classical sets and fuzzy sets, fuzzy set operations, fuzzy relations, fuzzy composition, natural language and fuzzy interpretations, fuzzy inference system, fuzzy controllers. **CO's–CO3**

Self-Learning Topics: Difference between crisp logic and fuzzy logic Natural Language & Fuzzy Interpretation- Membership functions for linguistic terms

UNIT IV:

14 Hours

Genetic algorithm: Fundamental, basic concepts, working principle, encoding, fitness function, reproduction,

Genetic modelling: Inheritance operator, cross over, inversion & deletion, mutation operator, Bitwise operator, Generational Cycle, Convergence of GA, Applications & advances in GA, Differences & similarities between GA & other traditional methods. **COs–CO4**

Self-Learning Topics: Basic terminology: population, chromosome, gene, Definition and role of fitness function.

UNIT V:

13 Hours

Hybrid Systems: Sequential Hybrid Systems, Auxiliary Hybrid Systems, Embedded Hybrid Systems, Neuro-Fuzzy Hybrid Systems, Neuro-Genetic Hybrid Systems, Fuzzy-Genetic Hybrid Systems.

GA based Back propagation Networks: GA based Weight Determination, K - factor determination in Columns. **CO's–CO5**

Self-Learning Topics: Real-world applications of hybrid systems, Use of GA to optimize weights

Board of Studies : Computer Science and Engineering

Approved in BoS No : 03

Approved in ACM No : 03

Text Books:

1. Soft Computing – Advances and Applications - Jan 2015 by B.K. Tripathy and J. Anuradha – Cengage Learning
2. S. N. Sivanandam, S N Deepa, Principles of Soft Computing, 2nd Edition, Wiley India, 2007
3. V. Kecman, Learning and Soft Computing, Pearson Education, India, 2006.
4. S, Rajasekaran & G.A. Vijayalakshmi Pai, Neural Networks, Fuzzy Logic & Genetic Algorithms, Synthesis & applications, PHI Publication, 1st Edition, 2009.

Reference Books:

1. S. Haykins, Neural networks: a comprehensive foundation Pearson Education, India, 2002
2. David E. Goldberg, “Genetic Algorithms-In Search, optimization and Machine learning”, Pearson Education
3. J.S.R.Jang, C.T.Sun and E.Mizutani, “Neuro-Fuzzy and Soft Computing”, Pearson Education 2004.
4. Timothy J. Ross, “Fuzzy Logic with Engineering Applications”, McGraw- Hill International

Editions, 1995

5. G.J. Klir & B. Yuan, “Fuzzy Sets & Fuzzy Logic”, PHI, 1995.
6. X. Yu and M. Gen, Introduction to Evolutionary Algorithms, Springer Verlag, 2010.

Web References:

1. <https://sci2s.ugr.es/links>
2. <https://pravin-hub-rgb.github.io/BCA/resources/sem5/sc/index.html>
3. <https://vtupulse.com/soft-computing/soft-computing-video-tutorial>.

Internal Assessment Pattern

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
L1	35	--
L2	40	--
L3	25	25
L4	--	35
L5	--	25
L6	--	15
Total (%)	100	100

Sample Short and Long Answers questions of Various Cognitive Levels

L1: Remember

1. Define Soft Computing
2. List components: Neural Networks, Fuzzy Logic, Genetic Algorithms
3. State difference between hard computing and soft computing
4. Define Artificial Neural Network
5. List models: McCulloch-Pitts, Hebb, Perceptron, ADALINE, BPN, RBF, TDNN
6. Identify supervised, unsupervised, associative networks
7. Define Fuzzy Logic.
8. List fuzzy set operations and relations
9. Define Genetic Algorithm
10. List terms: population, chromosome, gene, fitness function
11. Define hybrid systems
12. List types: Neuro-Fuzzy, Neuro-Genetic, Fuzzy-Genetic

L2: Understand

1. Explain why soft computing is needed in modern applications
2. Describe how soft computing handles uncertainty and approximation
3. Explain working of McCulloch-Pitts neuron
4. Describe learning rules (Hebbian, supervised learning)
5. Explain difference between crisp logic and fuzzy logic
6. Describe fuzzy inference system
7. Explain working principle of GA

8. Describe operators: crossover, mutation, selection
9. Explain working of hybrid models
10. Describe GA-based backpropagation networks

L3: Apply

1. Use soft computing techniques in simple real-life problems (e.g., decision-making systems)
2. Solve problems using perceptron or ADALINE
3. Implement basic neural network models
4. Construct membership functions
5. Solve problems using fuzzy operations
6. Solve optimization problems using GA
7. Implement encoding and fitness evaluation
8. Apply hybrid techniques in problem-solving
9. Demonstrate evidence collection process.

L4: Analysing

1. Compare hard computing vs soft computing approaches
2. Break down components of soft computing systems
3. Compare supervised vs unsupervised learning
4. Differentiate associative memory networks (BAM, Hopfield)
5. Analyze fuzzy relations and compositions
6. Interpret natural language using fuzzy logic
7. Compare GA with traditional optimization methods
8. Analyze convergence behavior
9. Compare different hybrid system architectures

L5: Evaluating

1. Justify advantages of soft computing over traditional methods
2. Assess performance of different neural network models
3. Evaluate effectiveness of fuzzy controllers
4. Evaluate efficiency of GA in different applications
5. Evaluate performance of hybrid systems vs single techniques

L6: Create

1. Design a simple hybrid soft computing model combining two techniques
2. Design a neural network model for pattern recognition
3. Design a fuzzy logic controller for a real-world system
4. Design a GA-based solution for a real-world optimization problem
5. Design a hybrid intelligent system combining ANN + GA or Fuzzy + GA.

**Chairperson
Board of Studies (CSE)**

R24CSPE18

FOUNDATION OF DATA SCIENCE

3 0 0 3

(Professional Elective - IV)

Course Objectives:

1. To understand the fundamentals of data science and its applications.
2. To learn data preprocessing, visualization, and statistical analysis.
3. To study probability and statistical methods for data analysis.
4. To explore data mining and machine learning basics.
5. To apply tools and techniques for real-world data analysis.

Course Outcomes:

At the end of the course, students will be able to:

Course Code	Course Outcomes (COs)	Mapping with POs and PSOs														Dok
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2		
R24CSPE 18.1	Understand data science concepts and data types	3	-	-	-	2	-	-	-	-	-	2	3	2	L1, L2	
R24CSPE 18.2	Apply data preprocessing and visualization techniques	3	3	-	2	3	-	-	2	2	-	2	3	2	L2, L3	
R24CSPE 18.3	Analyze data using statistical methods	3	3	-	2	3	-	-	-	-	-	2	3	3	L3, L4	
R24CSPE 18.4	Apply basic machine learning techniques	3	3	2	3	3	-	-	-	-	-	2	3	2	L4, L5	
R24CSPE 18.5	Evaluate data-driven solutions for real-world problems	3	2	3	2	3	2	3	-	-	2	3	-	-	L5, L6	

SYLLABUS**UNIT I:****12 Hours****Introduction to Data Science:** Definition, importance, applications, data science lifecycle.**Types of Data:** Structured, semi-structured, unstructured data.**Data Collection & Sources:** Databases, APIs, web data.**Data Preprocessing:** Data cleaning, handling missing values, normalization.**CO's–CO1****Self-Learning Topics:** Data science tools (Python, R).**UNIT II:****15 Hours****Data Visualization:** Importance, visualization techniques, charts (bar, line, histogram, and scatter).**Exploratory Data Analysis (EDA):** Summary statistics, correlation, covariance.**Tools:** Python libraries (Matplotlib, Seaborn).**CO's–CO2****Self-Learning Topics:** Dashboard tools (Tableau, Power BI).**UNIT III:****13 Hours****Probability Theory:** Basic probability, conditional probability, Bayes theorem**Statistics:** Mean median, mode, variance, standard deviation.**Distributions:** Normal distribution, binomial distribution.

Hypothesis Testing: t-test, chi-square test.

CO's–CO3

Self-Learning Topics: Central Limit Theorem

UNIT IV:

15 Hours

Machine Learning Basics: Supervised vs Unsupervised learning.

Algorithms: Linear regression, logistic regression, k-means clustering.

Model Evaluation: Accuracy, precision, recall.

COs–CO4

Self-Learning Topics: Decision trees.

UNIT V:

13 Hours

Applications of Data Science: Healthcare, finance, social media.

Big Data Concepts: Introduction to Hadoop and Spark.

Ethics in Data Science: Privacy, bias, security.

Future Trends: AI integration, automation.

CO's–CO5

Self-Learning Topics: Data science in industry

Board of Studies : Computer Science and Engineering

Approved in BoS No : 03

Approved in ACM No : 03

Expert Talk (To be Delivered by SMEs from Industries)

CO's/ PO's / PSO's

1. Data Science in Industry (Netflix, Amazon use cases) CO1–CO5, PO1, PO2, PO3, PO11, PSO1, PSO2
2. Real-time analytics and dashboards CO1–CO5, PO1, PO2, PO3, PO11, PSO1, PSO2
3. Machine Learning in production systems CO1–CO5, PO1, PO2, PO3, PO11, PSO1, PSO2
4. Ethical AI and data governance CO1–CO5, PO1, PO2, PO3, PO11, PSO1, PSO2

Text Books:

1. Joel Grus – *Data Science from Scratch*
2. Cathy O'Neil – *Doing Data Science*

Reference Books:

1. Hastie, Tibshirani – *Elements of Statistical Learning*
2. Python Data Science Handbook – Jake Vander Plas

Web References:

1. NPTEL Data Science Courses
2. Kaggle Learning Platform
3. Coursera Data Science Specialization

Internal Assessment Pattern

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
L1	35	--
L2	40	--
L3	25	25
L4	--	35
L5	--	25
L6	--	15
Total (%)	100	100

Sample Short and Long Answers questions of Various Cognitive Levels

L1: Remember

1. Define Data Science.
2. What are different types of data?
3. Define structured and unstructured data.
4. What is data pre-processing?
5. Define data visualization.
6. What is mean in statistics?
7. Define probability.
8. What is machine learning?
9. Define supervised learning.
10. What is Big Data?

L2: Understand

1. Explain the data science lifecycle.
2. Describe different data sources used in data science.
3. Explain the importance of data cleaning.
4. Differentiate between structured and unstructured data.
5. Explain different types of data visualization techniques.
6. Describe mean, median, and mode.
7. Explain the concept of probability with example.
8. Differentiate supervised and unsupervised learning.
9. Explain the role of EDA in data analysis.
10. Describe ethical issues in data science.

L3: Apply

1. Apply data cleaning techniques on a sample dataset.
2. Create a bar chart using given data.
3. Calculate mean and standard deviation for a dataset.
4. Apply normalization techniques on data.
5. Perform correlation analysis on a dataset.
6. Implement simple linear regression.
7. Apply k-means clustering for grouping data.
8. Create a histogram using sample data.
9. Apply Bayes theorem in a simple problem.
10. Use Python tools to visualize data.

L4: Analysing

1. Analyze the data science lifecycle with an example.
2. Compare different types of data visualization techniques.
3. Analyze the impact of missing data in datasets.
4. Compare mean vs median in skewed data.
5. Analyze the performance of regression models.
6. Compare supervised and unsupervised learning methods.
7. Analyze correlation between variables.
8. Examine the effect of outliers in datasets.
9. Compare different clustering techniques.
10. Analyze ethical challenges in data science.

L5: Evaluating

1. Evaluate the effectiveness of data preprocessing techniques.
1. Assess different visualization methods for large datasets.
2. Evaluate statistical methods used in data analysis.
3. Assess the performance of machine learning models.
4. Evaluate the impact of data quality on results.
5. Compare different regression models.
6. Evaluate the scalability of big data tools.
7. Assess ethical implications of data usage.
8. Evaluate clustering results for a dataset.
9. Justify the use of data science in decision-making

L6: Create

2. Design a complete data science workflow for a problem.
3. Create a data visualization dashboard.
4. Develop a machine learning model for prediction.
5. Design a data pre-processing pipeline.
6. Create a statistical analysis report.
7. Develop a clustering model for customer segmentation.
8. Design a real-time data analysis system.
9. Create big data processing architecture.
10. Develop a solution for handling missing data.
11. Design a data-driven solution for a real-world problem.

**Chairperson
Board of Studies (CSE)**

Course Objectives:

1. To understand the basic concepts and working of Information Retrieval systems.
2. To learn indexing methods and data structures used for storing and searching information.
3. To understand automatic indexing and clustering of documents.
4. To learn different search techniques and how results are ranked and visualized.
5. To study text search algorithms and evaluate system performance using measures like precision and recall.

Course Outcomes:

At the end of the course, students will be able to:

Course Code	Course Outcomes (COs)	Mapping with POs and PSOs														Dok
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2		
R24CSP E19.1	Explain the basic concepts and components of Information Retrieval systems.	2	2	–	–	1	–	–	–	1	–	1	2	1	L1, L2	
R24CSP E19.2	Understand and apply text pre-processing and indexing techniques.	3	2	1	–	3	–	–	–	–	–	1	3	2	L2, L3	
R24CSP E19.3	Apply automatic indexing and clustering methods to organize data.	3	3	2	1	3	–	–	–	–	–	1	3	2	L3, L4	
R24CSP E19.4	Use different search techniques and similarity measures to retrieve relevant information.	3	3	2	1	3	1	–	–	–	–	1	3	2	L4	
R24CSP E19.5	Evaluate the performance of Information Retrieval systems using standard metrics like precision and recall.	2	3	1	3	2	1	–	–	1	–	2	2	2	L5, L6	

SYLLABUS**UNIT I:****15 Hours**

Introduction: Definition, Objectives, Functional Overview, Relationship to DBMS, Digital libraries and Data Warehouses. Information Retrieval System Capabilities: Search, Browse, Miscellaneous

CO's–CO1

Self-Learning Topics: Working of Information Retrieval systems (Google search model)

UNIT II:**13 Hours**

Cataloging and Indexing: Objectives, Indexing Process, Automatic Indexing, Information Extraction. **Data Structures:** Introduction, Stemming Algorithms, Inverted file structures, N-gram data structure, PAT data structure, Signature file structure, Hypertext data structure.

CO's–CO2

Self-Learning Topics: Basics of stemming algorithms in text processing

UNIT III:

15 Hours

Automatic Indexing: Classes of automatic indexing, Statistical indexing, Natural language, Concept indexing, Hypertext linkages Document and Term Clustering: Introduction, Thesaurus generation, Item clustering, Hierarchy of clusters.

CO's–CO3

Self-Learning Topics: Document clustering and its applications (e.g., recommendation systems)

UNIT IV:

14 Hours

User Search Techniques: Search statements and binding, Similarity measures and ranking, Relevance feedback, Selective dissemination of information search, weighted searches of Boolean systems, Searching the Internet and hypertext. Information Visualization: Introduction, Cognition and perception, Information visualization technologies.

COs–CO4

Self-Learning Topics: Concept of information visualization in search systems

UNIT V:

13 Hours

Text Search Algorithms: Introduction, Software text search algorithms, Hardware text search systems. Information System Evaluation: Introduction, Measures used in system evaluation, Measurement example – TREC results.

CO's–CO5

Self-Learning Topics: Precision and Recall in system evaluation

Board of Studies : Computer Science and Engineering

Approved in BoS No : 03

Approved in ACM No: 03

Expert Talk (To be Delivered by SMEs from Industries)

CO's/ PO's / PSO's

1. **Information Retrieval Systems in Real-World Applications** (Search engines like Google, digital libraries, enterprise search systems, DBMS vs IR systems, cloud-based IR platforms) CO1–CO5, PO1, PO2, PO3, PO11, PSO1, PSO2
2. **Indexing and Data Structures in Search Engines** (Inverted index in Google search, stemming in NLP systems, N-gram models, signature files, hypertext linking in web systems) CO1–CO5, PO1, PO2, PO3, PO11, PSO1, PSO2
3. **Machine Learning and Clustering in Information Retrieval** (Document clustering, recommendation systems, NLP-based indexing, semantic search, AI-based information extraction) CO1–CO5, PO1, PO2, PO3, PO11, PSO1, PSO2

Text Books:

1. Information Storage and Retrieval Systems: Theory and Implementation by Gerald J. Kowalski, Mark T. Maybury, Second Edition, Kluwer Academic Publishers.

Reference Books:

1. Frakes, W.B., Ricardo Baeza-Yates: Information Retrieval Data Structures and Algorithms, Prentice Hall, 1992
2. Modern Information Retrieval By Yates Pearson Education
3. Information Storage & Retrieval By Robert Korfhage – John Wiley & Sons

Web References:

1. <https://nlp.stanford.edu/IR-book/>
2. <https://web.stanford.edu/class/cs276/>
3. <https://trec.nist.gov/>

Internal Assessment Pattern

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
L1	35	--
L2	40	--
L3	25	25
L4	--	35
L5	--	25
L6	--	15
Total (%)	100	100

Sample Short and Long Answers questions of Various Cognitive Levels

L1: Remember

1. Define Information Retrieval system.
2. List the objectives of Information Retrieval.
3. List any two IR system capabilities.
4. Define indexing in Information Retrieval.
5. List different data structures used in IR systems.
6. Define clustering in Information Retrieval.
7. List types of automatic indexing techniques.
8. Define similarity measure in IR.
9. List user search techniques.
10. Define information visualization.
11. Define precision and recall.
12. List types of text search algorithms.
13. What is system evaluation?

L2: Understand

1. Explain the functional overview of an Information Retrieval system.
2. Explain Search and Browse capabilities in IR system
3. Describe the relationship between IR systems and DBMS..
4. Explain the process of automatic indexing
5. Explain N-gram data structure.
6. Explain information extraction.
7. Explain statistical indexing.
8. Describe natural language indexing.
9. Describe hierarchy of clusters.
10. Explain weighted Boolean search.
11. Describe SDI (Selective Dissemination of Information).
12. Explain information visualization technologies.
13. Explain software-based search algorithms.
14. Explain precision and recall.

L3: Apply

1. Identify IR system components in a real-world search engine.
2. Illustrate the difference between search and browse with examples

3. Apply IR concepts to a library management system
4. Construct an inverted index for given documents.
5. Build a simple N-gram model.
6. Demonstrate signature file usage.
7. Apply clustering to group documents.
8. Apply NLP techniques in indexing.
9. Apply similarity measures in ranking results.
10. Rank search results using a method.
11. Calculate precision and recall for given data.
12. Use evaluation metrics in IR system.
13. Demonstrate TREC evaluation.

L4: Analysing

1. Differentiate between Information Retrieval systems and DBMS with suitable examples
2. Compare Digital Libraries and Data Warehouses in terms of structure and usage.
3. Examine the challenges of handling unstructured data in IR systems compared to DBMS
4. Differentiate between PAT structure and signature file structure.
5. Examine the advantages of stemming algorithms in IR.
6. Compare clustering and classification techniques.
7. Differentiate between statistical and concept indexing
8. Examine the role of hypertext links in IR systems.
9. Analyze ranking methods in IR systems.
10. Examine the impact of relevance feedback on search results.
11. Compare evaluation methods used in IR systems.
12. Analyze performance of IR systems using metrics.
13. Analyze different text search algorithms.

L5: Evaluating

1. Evaluate the importance of Information Retrieval systems in modern web applications.
2. Justify why IR systems are more suitable than DBMS for text-based data.
3. Evaluate the effectiveness of inverted indexing in large datasets.
4. Critically analyze different indexing approaches in IR systems.
5. Evaluate the performance of automatic indexing techniques.
6. Evaluate clustering techniques used in IR systems.
7. Critically analyze concept indexing approaches.
8. Evaluate different indexing techniques for text organization.
9. Justify the importance of relevance feedback.
10. Critically analyze SDI systems in information retrieval.
11. Evaluate search strategies used in modern search engines.
12. Evaluate IR system performance using standard metrics.
13. Assess trade-off between precision and recall

L6: Create

1. Design a basic architecture of an Information Retrieval system.
2. Develop a model for a university digital library system.
3. Create a simple web-based search system for document retrieval.
4. Design an efficient inverted indexing system for a search engine.
5. Create an N-gram based text retrieval system

6. Propose an optimized data structure for fast retrieval.
7. Design a document clustering system for news articles
8. Design a hierarchical document clustering model.
9. Propose an intelligent clustering framework for IR.
10. Design an SDI-based alert system.
11. Propose an intelligent search and visualization system.
12. Develop a text search algorithm for document retrieval
13. Propose an improved evaluation methodology for IR systems.

**Chairperson
Board of Studies (CSE)**

Course Objectives:

1. To understand the fundamentals of Python programming, including syntax, data types, and operators.
2. To apply control structures such as conditional statements and loops for problem-solving.
3. To demonstrate the use of strings, lists, tuples, and dictionaries for data manipulation.
4. To develop modular programs using functions and basic object-oriented programming concepts.
5. To understand file handling, input/output operations, and basic debugging techniques.

Course Outcomes:

At the end of the course, students will be able to:

Course Code	Course Outcomes (COs)	Mapping with POs and PSOs												Dok	
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1		PSO2
R24CSOE1 1.1	Apply Python syntax, data types, and operators to develop basic programs.	3	1	-	1	2	-	-	-	-	-	1	2	2	L1, L2
R24CSOE1 1.2	Design solutions using conditional statements and looping constructs.	3	2	1	2	3	-	-	1	-	-	1	2	2	L2, L3
R24CSOE1 1.3	Manipulate and manage data using strings, lists, tuples, and dictionaries.	2	3	-	2	3	-	-	-	-	-	1	2	3	L3, L4
R24CSOE1 1.4	Develop modular and reusable code using functions and basic object-oriented concepts.	3	2	1	3	3	-	-	-	-	-	1	3	2	L4, L5
R24CSOE1 1.5	Perform file handling operations and implement debugging techniques for error-free programs.	3	2	2	3	3	1	1	-	-	1	2	2	2	L5, L6

SYLLABUS**UNIT I:****15 Hours**

Introduction to Python Programming: Features/Characteristic of Python - Basic Syntax - Basic Data Types - Simple Input-Output - Precedence of Operators - Type Conversion - Conditional Statements - Looping: for, while, Nested Loops; Terminating Loops - Skipping Specific Conditions - Scope and Extent of Variables - Testing and Debugging Principles.

CO's-CO1

Self-Learning Topics: Command line execution of Python programs

UNIT II:**15 Hours**

Strings & Lists: Concept - Escape Characters - String Special Operations - String Formatting Operator - Single Quotes - Double Quotes - Triple Quotes - Raw String - Unicode Strings - Built-in String Methods - List Type Built-in Methods - Special Features of Lists. **CO's-CO2**

Self-Learning Topics: Advanced string slicing and indexing techniques.

UNIT III:

14 Hours

Tuples, Dictionaries & Functions: Tuples - Tuple Operators and Built-in Functions - Special Features of Tuples - Dictionaries, Operators, Built-in Functions, Built-in Methods, Dictionary Keys - Functions - Calling Functions - Creating Functions - Formal Arguments - Positional Arguments - Default Arguments - Variable-Length Arguments. **CO's-CO3**

Self-Learning Topics: Nested dictionaries and real-world applications

UNIT IV:

10 Hours

Object Oriented Programming: Introduction to Classes, Objects and Methods; Standard Libraries **COs-CO4**

Self-Learning Topics: Encapsulation and abstraction concepts.

UNIT V:

14 Hours

Files and Input/Output: File Objects - File Built-in Function - File Built-in Methods - File Built-in Attributes - Standard Files - Command-Line Arguments - File System - File Execution - Persistent Storage Modules. **CO's-CO5**

Self-Learning Topics: Exception handling in files operations

Board of Studies : Computer Science and Engineering

Approved in BoS No : 03

Approved in ACM No: 03

Text Books:

1. Hetland, Magnus Lie, "Beginning Python from Novice to Professional," Apress, 2nd Edition, 2017.
2. Lutz, M., "Learning Python: Powerful Object-Oriented Programming," O'Reilly Media, Inc. 5th Edition, 2013.

Reference Books:

1. FabeoNelli, "Python Data Analytics," Apress, 2nd Edition, 2018.
2. Paul Barry, "Head first Python: A Brain-Friendly Guide," O'Reilly Media, Inc., 2nd, 2016.

Web References:

1. <https://www.python.org/doc/>
2. <https://docs.python.org/3/tutorial/>
3. <https://www.w3schools.com/python/>
4. <https://www.geeksforgeeks.org/python/>
5. <https://www.programiz.com/python-programming/>

Internal Assessment Pattern

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
L1	35	--
L2	40	--
L3	25	25
L4	--	35

L5	--	25
L6	--	15
Total (%)	100	100

Sample Short and Long Answers questions of Various Cognitive Levels

L1: Remember

1. What is Python?
2. List any four features of Python.
3. Name basic data types in Python.
4. What is a variable?
5. List arithmetic operators in Python.
6. What is a loop?
7. Name two types of loops in Python.
8. What is a string?
9. Define list in Python.
10. What is a tuple?
11. What is a dictionary?
12. What is a function?
13. What is a file in Python?

L2: Understand

1. Explain Python syntax rules.
2. Describe the use of indentation in Python.
3. Differentiate between = and == operators.
4. Explain type conversion with example.
5. Describe conditional statements in Python.
6. Explain for loop with syntax.
7. Explain while loop with syntax.
8. What is string formatting? Explain.
9. Explain escape characters with example.
10. Differentiate between list and string.
11. Explain tuple characteristics.
12. Describe dictionary structure.
13. Explain scope of variables.

L3: Apply

1. Write a program to add two numbers.
2. Write a program to check even or odd number.
3. Write a program using if-else statement.
4. Write a program using for loop.
5. Write a program using while loop.
6. Write a program to print multiplication table.
7. Write a program to find factorial of a number.
8. Write a program to reverse a string.
9. Write a program to create a list and display elements.
10. Write a program to access tuple elements.
11. Write a program to create a dictionary.

12. Write a program using a function.

L4: Analysing

1. Compare for and while loops.
2. Differentiate list and tuple.
3. Compare dictionary and list.
4. Analyze nested loops usage.
5. Identify errors in given code snippet.
6. Compare local and global variables.
7. Analyze function vs built-in function.
8. Compare different string operations.
9. Analyze data type conversion issues.
10. Differentiate append and insert in list.
11. Compare file read and write operations.
12. Analyze use of break and continue.

L5: Evaluating

1. Justify use of functions in programming.
2. Evaluate importance of debugging.
3. Which loop is better: for or while? Justify.
4. Evaluate efficiency of lists vs tuples.
5. Assess importance of file handling.
6. Judge correctness of a given algorithm.
7. Evaluate use of dictionaries in real applications.
8. Compare different ways of string formatting.
9. Critically analyze nested loop performance.
10. Evaluate role of indentation in Python.
11. Judge best method for input handling.

L6: Create

1. Create a file handling system for records.
2. Design a login system using dictionaries.
3. Develop a program using OOP concepts.
4. Create a program for banking operations.
5. Design a program for library management.
6. Build a program using nested loops.
7. Create a program to process list data.
8. Develop a program to handle string operations.
9. Design a program combining functions and loops.

**Chairperson
Board of Studies (CSE)**

R24CSOE12**OBJECT ORIENTED ANALYSIS AND DESIGN****3 0 0 3**

(Open Elective-III)

Course Objectives:

1. Understand the structure and nature of complex systems.
2. Learn basic structural concepts like classes, relationships, and diagrams.
3. To develop the ability to model system structure using class and object diagrams, including advanced Structural concepts.
4. Learn system behavior using use case, interaction, and activity diagrams.
5. To explore advanced behavioral and architectural modeling techniques, including state machines, Component and Deployment diagrams.

Course Outcomes:

At the end of the course, students will be able to:

Course Code	Course Outcomes (COs)	Mapping with POs and PSOs														Domain
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2		
R24CSO E12.1	Understand methods to design and manage complexity in system architecture.	3	2	3	1	1	0	0	0	1	1	0	2	1	L2 L3	
R24CSO E12.2	Understand UML concepts and the importance of modeling in software development.	3	2	2	1	2	0	0	0	1	1	0	2	2	L2 L3	
R24CSO E12.3	Understand and apply class and object diagram concepts for system modeling.	3	3	2	2	2	0	0	0	1	1	0	3	2	L3 L4	
R24CSO E12.4	Understand behavioral modeling using use case, activity, and interaction diagrams.	3	3	3	2	2	0	0	0	2	2	1	3	2	L3 L4	
R24CSO E12.5	Understand advanced behavioral concepts like state machines and processes.	3	3	3	2	2	1	0	0	2	2	1	3	3	L4 L5	

SYLLABUS**UNIT I:****15 Hours**

The Structure of Complex Systems, The Inherent Complexity of Software, Attributes of Complex System, Organized and Disorganized Complexity, Bringing Order to Chaos, Designing Complex Systems.

CO's-CO1

Self-Learning Topics: Banking Transaction Systems

UNIT II: **13 Hours**

Introduction to UML: Importance of modeling, principles of modeling, object-oriented modeling, conceptual model of the UML, Architecture, and Software Development Life Cycle. Basic Structural Modeling: Classes, Relationships, common Mechanisms, and diagrams. **CO's–CO2**

Self-Learning Topics: Software Design Documentation, Traffic management system modelling.

UNIT III: **15 Hours**

Class & Object Diagrams: Terms, concepts, modeling techniques for Class & Object Diagrams. Advanced Structural Modeling: Advanced classes, advanced relationships, Interfaces, Types and Roles, Packages. **CO's–CO3**

Self-Learning Topics: Library Management Systems, Crypt Analysis.

UNIT IV: **15 Hours**

Basic Behavioral Modeling-I: Interactions, Interaction diagrams Use cases, Use case Diagrams, Activity Diagrams. **COs–CO4**

Self-Learning Topics: Online booking systems, Vacation Tracking System.

UNIT V: **13 Hours**

Advanced Behavioral Modeling: Events and signals, state machines, processes and Threads, time and space, state chart diagrams. Architectural Modeling: Component, Deployment, Component diagrams and Deployment diagrams. **CO's–CO5**

Self-Learning Topics: Web Applications (Weather Forecasting)

Board of Studies : Computer Science and Engineering

Approved in BoS No: 03

Approved in ACM No: 03

Text Books:

1. Grady BOOCH, Robert A. Maksimchuk, Michael W. ENGLE, Bobbi J. Young, Jim Conallen, Kellia Houston, “Object- Oriented Analysis and Design with Applications”, 3rd edition, 2013, PEARSON.
2. Grady Booch, James Rumbaugh, Ivar Jacobson: The Unified Modeling Language User Guide, Pearson Education.

Reference Books:

1. Meilir Page-Jones: Fundamentals of Object-Oriented Design in UML, Pearson Education.
2. Pascal Roques: Modeling Software Systems Using UML2, WILEY- Dreamtech India Pvt. Ltd.
3. Atul Kahate: Object Oriented Analysis & Design, The McGraw-Hill Companies.
4. Applying UML and Patterns: An introduction to Object – Oriented Analysis and Design and Unified Process, Craig Larman, Pearson Education.

Web References:

1. <https://nptel.ac.in/courses/106105153>
2. https://www.tutorialspoint.com/object_oriented_analysis_design/index.htm
3. <https://www.geeksforgeeks.org/software-engineering/object-oriented-analysis-and-design/>

Internal Assessment Pattern

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
L1	35	--
L2	40	--
L3	25	25
L4	--	35
L5	--	25
L6	--	15
Total (%)	100	100

Sample Short and Long Answers questions of Various Cognitive Levels

L1: Remember

1. Define a complex system and describe its basic structure with suitable examples.
2. Explain the concept of system hierarchy in complex systems and list its main characteristics.
3. Explain the importance and principles of modeling in software engineering.
4. Describe object-oriented modeling and its features.
5. Describe the concepts and modeling techniques for class diagrams.
6. Explain different types of relationships in UML with examples.
7. Describe use cases and use case diagrams with examples.
8. Explain activity diagrams and their components.
9. Describe state machines and state chart diagrams in detail.
10. Explain processes and threads with examples.
11. Discuss about defining the TTMS architecture.

L2: Understand

1. Describe in detail the methods used to bring order to chaotic systems.
2. Explain the process of designing complex systems with relevant examples.
3. Describe the Software Development Life Cycle (SDLC) and its phases.
4. Explain basic structural modeling concepts including classes and relationships.
5. Describe advanced relationships, interfaces, types, and roles in UML.
6. Explain the concept of packages and their importance in large systems.
7. Explain interaction modeling with different types of diagrams.
8. Describe in detail use case modeling and its significance.
9. Describe component and deployment diagrams in detail.
10. Explain architectural modeling and its importance in system design.
11. What is an object diagram? Explain identifying objects in object diagrams.
12. Describe terms and notations of use cases with suitable examples.

L3: Apply

1. Illustrate organized and disorganized complexity using a suitable system diagram.

2. Apply system design principles to represent a real-world complex system with a block diagram.
3. 3)Draw a UML class diagram for a simple student management system.
4. Model a software system using basic UML structural diagrams (classes and relationships).
5. Create a conceptual model diagram for an online shopping system.
6. Create an object diagram for a banking system scenario.
7. Draw UML diagrams illustrating association, aggregation, and composition.
8. Model a system using packages, interfaces, and advanced relationships.
9. Create an activity diagram for an ATM transaction process.
10. Draw a sequence diagram for a login system.
11. Describe client and implementation views of Vacation Tracking System.
12. Explain about Key Abstractions and Mechanisms of TTMS.

L4: Analyze

1. Compare and analyze organized complexity and disorganized complexity in real-world systems.
2. Analyze how different attributes of complex systems affect system design decisions.
3. Analyze object-oriented modeling and compare it with traditional modeling approaches.
4. Discuss about the elaboration phase of Cryptanalysis.
5. Examine various UML relationships and analyze when each should be used.
6. Analyze advanced classes and relationships in complex system modeling.
7. Examine use case diagrams and analyze how they capture system requirements.
8. Analyze activity diagrams and identify how they represent workflows effectively.
9. What are requirements of Cryptanalysis? Explain about analysis of knowledge sources.
10. Break down component and deployment diagrams and analyze their differences.
11. Analyze architectural modeling and evaluate its importance in large-scale systems.
12. Discuss about scenarios of Weather Monitoring System.
13. Discuss about requirements and post transition of Weather Monitoring System.

L5: Evaluate

1. Evaluate the effectiveness of different approaches used to manage the complexity of software systems.
2. Justify how system attributes influence the success or failure of complex system design.
3. Evaluate the use of UML as a standard modeling language in software development.
4. Justify the importance of object-oriented modeling compared to traditional modeling techniques.
5. Critically evaluate different types of relationships in UML.
6. Assess the importance of interfaces and packages in large-scale systems.
7. Evaluate the effectiveness of class diagrams in representing real-world system structures.
8. Critically assess the importance of interfaces and packages in large-scale system design.
9. Evaluate the role of use case diagrams in capturing system requirements effectively.
10. Justify the effectiveness of activity diagrams in modeling system workflows.
11. Explain about allocating nonfunctional requirements and specifying interfaces in satellite-based navigation.
12. Describe dependencies and affirmations of Cryptanalysis.

13. Describe sensor mechanism and display mechanism of Weather Monitoring System
14. How events and signals are useful in UML? Explain.

L6: Create

1. Design a complex system for a banking application and illustrate its structure.
2. Create a model showing how a software system can be designed to handle complexity effectively.
3. Propose a system design approach to manage a large-scale real-world application (e.g., railway reservation system).
4. Design a UML model for an online shopping system including structural diagrams.
5. Create a complete object-oriented model for a library management system.
6. Develop a UML architecture for a traffic management system.
7. Design a class diagram and object diagram for a hospital management system.
8. Create a UML model using advanced relationships, interfaces, and packages for a university system.
9. Develop a complete structural model for a banking system using UML diagrams.
10. Design a use case diagram and activity diagram for an online food ordering system.
11. Create a complete behavioral model for a ticket booking system using UML diagrams.
12. Develop a UML model for a chat application showing interactions and workflows.
13. Design a state machine diagram for an ATM system showing all possible states.
14. Create component and deployment diagrams for a web-based application.
15. Develop an architectural model for a distributed e-commerce system.

**Chairperson
Board of Studies (CSE)**

Course Objectives:

1. To understand the fundamental concepts of digital image processing, including image acquisition, sampling, quantization, and pixel relationships.
2. To Analyze and apply various image transformation techniques such as Fourier, DCT, Walsh, Hadamard, and other orthogonal transforms.
3. To Apply spatial and frequency domain techniques for image enhancement and improvement of image quality.
4. To Understand and implement image restoration techniques and morphological operations for image correction and structure analysis.
5. To apply segmentation techniques to partition images into meaningful regions using edge, region, and threshold-based methods.

Course Outcomes:

At the end of the course, students will be able to:

Course Code	Course Outcomes (COs)	Mapping with POs and PSOs												Dok	
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1		PSO2
R24CS OE13.1	Understand the fundamental concepts of digital image processing, including image acquisition, sampling, and quantization.	3	2	1	-	-	-	-	-	-	-	-	3	1	L1, L2
R24CS OE13.2	Apply and analyze various mathematical transforms such as Fourier, DCT, Walsh, and Hadamard for image processing in transform domains.	3	3	2	-	2	-	-	-	-	-	-	3	2	L2, L3
R24CS OE13.3	Design and implement image enhancement techniques in spatial and frequency domains to improve image quality.	3	2	3	2	3	-	-	-	-	-	-	3	3	L3, L4
R24CS OE13.4	Analyze image degradation and apply restoration techniques along with morphological operations for image correction and structural analysis.	3	3	2	3	2	-	-	-	-	-	-	3	3	L4, L5
R24CS OE13.5	Apply image segmentation and compression techniques including edge detection, region-based methods, Huffman coding, and JPEG standards for efficient image representation.	3	3	2	2	3	-	-	-	-	-	-	3	3	L5, L6

SYLLABUS

UNIT I: 14 Hours

Image Processing Fundamentals: Introduction: Basics of Image Processing, Examples of Fields that Use Digital Image Processing, Image Sensing and Acquisition, Image Sampling and Quantization, Some Basic Relationships between Pixels. **CO's–CO1**

Self-Learning Topics Mathematical Tools Used in Digital Image Processing

UNIT II: 13 Hours

Image Transforms: Need for Transform, Image Transforms, Fourier Transform, 2D Discrete Fourier Transform, Properties of 2D-DFT, Walsh Transform, Hadamard Transform, Discrete Cosine Transform, Haar Transform, Slant Transform, Discrete Cosine Transform. **CO's –CO2**

Self-Learning Topics KL Transform, Singular Value Decomposition

UNIT III: 14 Hours

Image Enhancement: Enhancement in Spatial Domain, Enhancement through Point Processing, Types of Point Processing, Gray level Transformations, Histogram processing, Spatial Domain High-Pass Filtering, Spatial Domain High-Pass Filtering, Filtering in Frequency Domain. **CO's–CO3**

Self-Learning Topics Local or Neighborhood criterion, Low Pass (Smoothing) and High Pass (Sharpening) Filters in Frequency Domain

UNIT IV: 15 Hours

Image Restoration & Morphology: Image Restoration: Degradation Model, Algebraic Approach to Restoration, Inverse Filtering, Least Mean Square Filters.

Image Morphology: Basic Morphological algorithms Boundary Extraction, Region Filling, Extraction of Connected Components, Convex Hull, Skeletons. **CO's –CO4**

Self-Learning Topics Constrained Least Squares Restoration, Interactive Restoration

UNIT V: 13 Hours

Image Compression & Recognition: Image Compression: Redundancies and their Removal Methods, Fidelity Criteria, Image Compression Models, Huffman and Arithmetic Coding, Error Free Compression, Lossy Compression, Lossy and Lossless Predictive Coding, JPEG 2000 Standards. **CO's –CO5**

Self-Learning Topics Transform Based Compression

Board of Studies : Computer Science and Engineering

Approved in BoS No : 03

Approved in ACM No: 03

Text Book:

1. Digital Image Processing - Rafael C. Gonzalez, Richard E. Woods, 3rd Edition, Pearson.
2. Digital Image Processing- S Jayaraman, S Esakkirajan, T Veerakumar- MC GRAW HILL EDUCATIO

Reference Books:

1. Digital Image Processing and Analysis-Human and Computer Vision Application with using CVIP Tools - ScotteUmbaugh, 2nd Ed, CRC Press, 2011
2. Digital Image Processing using MATLAB – Rafael C. Gonzalez, Richard E Woods and Steven L. Eddings, 2nd Edition, MC GRAW HILL EDUCATION, 2010.
3. Digital Image Processing and Computer Vision – Somka, Hlavac, Boyle- Cengage Learning (Indian edition) 2008.
4. Introductory Computer Vision Imaging Techniques and Solutions- Adrian low, 2008,2 nd

Edition

Internal Assessment Pattern

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
L1	35	--
L2	40	--
L3	25	25
L4	--	35
L5	--	25
L6	--	15
Total (%)	100	100

Sample Short and Long Answers questions of Various Cognitive Levels**L1: Remember**

1. Define digital image processing.
2. State the difference between sampling and quantization.
3. List types of image transforms.
4. State properties of 2D DFT.
5. State histogram equalization.
6. Define image enhancement.
7. List morphological operations.
8. Define image compression.
9. State Huffman coding.
10. Define image restoration.

L2: Understand

1. Explain the basic components of a digital image processing system.
2. Describe image sensing and acquisition with a neat diagram.
3. Explain the need for image transforms.
4. Explain the concept of frequency domain representation.
5. Explain spatial domain enhancement techniques.
6. Describe gray level transformations.
7. Explain least mean square filter.
8. Describe erosion and dilation.
9. Explain lossy and lossless compression.
10. Explain predictive coding.

L3: Apply

1. Convert a given analog image into a sampled and quantized form.
2. Compute pixel adjacency (4-neighbourhood and 8-neighbourhood) for a given matrix.
3. Apply DCT on a small image block.
4. Compute 2D DFT of a given image matrix.
5. Apply histogram equalization on a given image.
6. Perform smoothing and sharpening using filters.

7. Apply morphological operations on binary images.
8. Restore a degraded image using inverse filtering.
9. Compress a simple image using run-length encoding.
10. Apply Huffman coding to a data set.

L4: Analysing

1. Compare sampling and quantization with suitable examples.
2. Analyze how image resolution affects image quality.
3. Compare Fourier and DCT transforms.
4. Analyze energy compaction property of transforms.
5. Analyze the effect of high-pass filtering on images.
6. Compare spatial and frequency domain enhancement.
7. Analyze causes of image degradation.
8. Compare restoration and enhancement.
9. Analyze different types of image redundancies.
10. Compare lossy and lossless compression techniques.

L5: Evaluating

1. Justify the importance of sampling rate in digital image processing
2. Evaluate the effect of quantization levels on image compression.
3. Evaluate which transform is best for image compression and why.
4. Justify the use of DCT in JPEG standard.
5. Evaluate which enhancement technique is suitable for noisy images.
6. Justify the use of histogram processing.
7. Evaluate effectiveness of LMS filter in noise removal.
8. Justify use of morphology in image processing.
9. Evaluate efficiency of arithmetic coding over Huffman coding.
10. Justify need for image compression in real-time systems.

. L6: Create

1. Design a simple model for image acquisition system.
2. Design a transform-based image compression system.
3. Design an algorithm for image enhancement in both spatial and frequency domain.
4. Design a system for image restoration using filtering techniques.
5. Design an image compression model for multimedia applications.

**Chairperson
Board of Studies (CSE)**

R24CSOE14**WEB TECHNOLOGIES****3 0 0 3**

(Open Elective-III)

Course Objectives:

1. To understand different Internet Technologies
2. To learn java-specific web services architecture
3. To Develop web applications using frameworks

Course Outcomes:

At the end of the course, students will be able to:

Course Code	Course Outcomes (COs)	Mapping with POs and PSOs														Dok
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2		
R24CSOE 14.1	Construct a basic website using HTML and Cascading Style Sheets	3	2	3	-	2	2	-	1	2	-	1	3	2	L1, L2	
R24CSOE 14.2	Build dynamic web page with validation using Java Script objects and by applying different event handling mechanisms	3	3	3	1	3	1	-	2	2	-	2	3	3	L2, L3	
R24CSOE 14.3	Develop server-side programs using Servlets and JSP.	3	3	3	2	3	2	-	2	2	2	2	3	3	L3, L4	
R24CSOE 14.4	Construct simple web pages in PHP and to represent data in XML format.	3	2	3	1	3	2	-	2	2	1	2	3	3	L4, L5	
R24CSOE 14.5	Develop interactive web applications.	3	3	3	2	3	2	1	3	3	2	3	3	3	L5, L6	

SYLLABUS**UNIT I: Website Basics, HTML5, CSS3, Web 2.0****13 Hours**

Web Essentials: Clients, Servers and Communication, The Internet, World Wide Web, HTTP Request Message, HTTP Response Message, Web Clients, Web Servers, HTML5, Forms, Images, Tables, Lists, Audio, Video controls, CSS3, External style sheets, CSS selectors, Colors, Text styling, Layout basics, Responsive design, Bootstrap

CO's-CO1

Self-learning topics: Responsive web design, Flexbox and Grid layouts, HTTP vs HTTPS basics, UI design principles

UNIT II: Client-Side Programming**15 Hours**

JavaScript, Introduction to JavaScript, DOM manipulation, Form validation, Events handling, JSON, API calls using fetch, Local Storage and Session Storage in web applications, JavaScript error handling and debugging techniques

CO's-CO2

Self-learning topics: ES6 features, Asynchronous JavaScript (Promises, async/await), Working with REST APIs, Browser developer tools

UNIT III: Server-Side Programming**12 Hours**

Servlets, Java Servlet basics, Servlet life cycle, Request handling (GET, POST), Session management, Cookies, JDBC database connectivity using JDBC. **CO's–CO3**

Self-learning topics: REST API basics, MVC architecture, Authentication basics, SQL Injection prevention

UNIT IV: PHP and XML

13 Hours

Introduction to PHP, Variables, Forms handling, CRUD operations with MySQL, XML basics, XML vs JSON **CO's–CO4**

Self-learning topics: PHP with MySQL projects, File handling, REST API integration, XPath basics

UNIT V: Web Frameworks & Tools

12 Hours

Introduction to React, Basics of Node.js, Introduction to AngularJS, Basics of Firebase, UI/UX fundamentals, Web application structure, Component lifecycle in React, State management basics

Self-learning topics: React vs Angular comparison, Node.js backend basics, Firebase authentication, Modern frontend development **CO's–CO5**

Board of Studies : Computer Science and Engineering

Approved in BoS No : 03

Approved in ACM No: 03

Expert Talk (To be Delivered by SMEs from Industries)

CO's/ PO's / PSO's

1. Modern web development uses HTML5 for semantic structure and multimedia support, CSS3 for responsive design and advanced styling, and Bootstrap for mobile-first UI development. HTTP request-response model forms the base of client-server communication in web applications. Web 2.0 enables interactive and user-driven websites like social platforms. CO1–CO5, PO1, PO2, PO3, PO11, PSO1, PSO2
2. JavaScript is used for building interactive web applications through DOM manipulation and event handling. Form validation ensures data correctness on the client side. JSON is widely used for data exchange between frontend and backend systems. CO1–CO5, PO1, PO2, PO3, PO11, PSO1, PSO2
3. Java Servlets handle server-side processing of client requests using GET and POST methods. Session management and cookies are used to maintain user state in web applications. JDBC provides database connectivity for enterprise-level applications. CO1–CO5, PO1, PO2, PO3, PO11, PSO1, PSO2

Text Books:

1. Web Technologies – Uttam K. Roy (Oxford University Press India)
2. Internet & World Wide Web: How to Program – Deitel & Deitel (Pearson India)
3. Java: The Complete Reference – Herbert Schildt (McGraw Hill India)
4. Web Technologies (Theory and Practice) – S. Padma Priya (Dreamtech Press India)
5. Beginning PHP, MySQL, JavaScript and HTML5 for Web Development – Wrox (Wiley India)

Reference Books:

1. Web Technologies – Uttam K. Roy (Oxford University Press India),
2. Internet & World Wide Web: How to Program – Deitel & Deitel (Pearson India),
3. Java: The Complete Reference – Herbert Schildt (McGraw Hill Education India),

Web References:

1. <https://developer.mozilla.org> (MDN Web Docs – HTML5, CSS3, JavaScript, DOM, APIs)

2. <https://www.w3schools.com> (HTML, CSS, JavaScript, PHP, XML beginner tutorials)
3. <https://www.javatpoint.com> (Servlets, JDBC, PHP, Web Technologies simplified notes)
4. <https://docs.oracle.com> (Java Servlet, JDBC, official Java backend documentation)

Internal Assessment Pattern

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
L1	35	--
L2	40	--
L3	25	25
L4	--	35
L5	--	25
L6	--	15
Total (%)	100	100

Sample Short and Long Answers questions of Various Cognitive Levels

L1: Remember

1. What is HTML5?
2. What is CSS3?
3. What is JavaScript?
4. Define DOM.
5. What is a Servlet?
6. Define JDBC.
7. What is PHP?
8. What is XML?
9. What is React?
10. What is Node.js?

L2: Understand

1. Explain client-server architecture of web.
2. Describe HTTP request and response model.
3. Explain DOM structure in JavaScript.
4. Describe form validation process in JavaScript.
5. Explain Servlet life cycle.
6. Describe GET and POST methods.
7. Explain XML structure.
8. Describe PHP form handling.
9. Explain React component concept.
10. Explain Firebase services.

L3: Apply

1. Create HTML page with forms, tables, images.
2. Apply CSS using external stylesheet.
3. Write JavaScript form validation program.

4. Fetch API data using JavaScript.
5. Write Servlet using GET/POST methods.
6. Connect Java program with database using JDBC.
7. Write PHP program for form handling.
8. Perform CRUD operations using PHP & MySQL.
9. Create React component.
10. Build Node.js simple server.

L4: Analyze

1. Compare HTML5 and HTML4.
2. Differentiate inline, internal, external CSS.
3. Compare synchronous and asynchronous JavaScript.
4. Analyze DOM manipulation techniques.
5. Compare cookies and sessions.
6. Analyze Servlet architecture.
7. Compare XML and JSON.
8. Analyze PHP vs other server-side scripting languages.
9. Compare React and AngularJS.
10. Analyze frontend vs backend roles.

L5: Evaluate

1. Evaluate Bootstrap in modern web design.
2. Assess responsive web design importance.
3. Evaluate asynchronous JavaScript usage.
4. Evaluate API-based web applications.
5. Evaluate session management importance.
6. Evaluate JDBC in enterprise applications.
7. Evaluate PHP in modern development.
8. Evaluate XML usage vs JSON.
9. Evaluate React in frontend development.
10. Evaluate full-stack development frameworks.

L6: Create

1. Design responsive website using HTML5, CSS3, Bootstrap.
2. Create interactive web page using JavaScript + API.
3. Design web application using Servlets + JDBC.
4. Design dynamic website using PHP + MySQL.
5. Create full-stack application using React + Node.js + Firebase.

Chairperson
Board of Studies (CSE)

R24CSOE15 FUNDAMENTAL OF INFORMATION SECURITY 3 0 0 3
(Computer Science and Engineering)

Course Objectives:

1. To understand the fundamentals of information security concepts and principles.
2. To analyze threats, vulnerabilities, and cyber-attacks in computer systems.
3. To understand cryptography, authentication, and access control mechanisms.
4. To illustrate network security, web security, and secure communication techniques.
5. To understand cyber laws, ethical hacking concepts, and security management practices.

Course Outcomes:

At the end of the course, students will be able to:

Course Code	Course Outcomes (COs)	Mapping with POs and PSOs												Dok	
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1		PSO2
R24CSOE 15.1	Describe fundamentals of information security principles, threats, and vulnerabilities	3	-	-	-	2	-	-	-	-	-	2	3	2	L1 L2
R24CSOE 15.2	Analyze cryptographic techniques and authentication mechanisms	3	3	-	2	3	-	-	2	-	-	2	3	2	L2 L3
R24CSOE 15.3	Analyze process synchronization mechanisms and dead lock handling techniques.	3	3	2	2	3	-	-	-	2	-	2	3	3	L3 L4
R24CSOE 15.4	Analyze system security, web security, and malware protection techniques	3	2	2	2	3	-	-	-	-	-	2	3	2	L4 L5
R24CSOE 15.5	Evaluate cyber laws, ethical hacking practices, and information security management	3	2	3	2	3	2	2	-	-	2	3	-	-	L5 L6

SY3LLABUS0

UNIT I:

15 Hours

Introduction to Information Security: Overview of Information Security, Need for Security, Security Goals – Confidentiality, Integrity, Availability, Threats and Vulnerabilities, Types of Attacks, Security Services and Mechanisms. Security Models and Standards: OSI Security Architecture, Security Policies, Risk Management, Security Standards and Frameworks, Cyber Security Overview.

CO's–CO1

Self-Learning Topics: Evolution of Cyber Security and Recent Security Incidents.

UNIT II:

15 Hours

Cryptography: Classical Encryption Techniques, Symmetric Key Cryptography, Asymmetric Key

Cryptography, Public Key Infrastructure, Hash Functions, Digital Signatures. Authentication Mechanisms: User Authentication, Password Security, Multifactor Authentication, Biometric Authentication, Key Management Techniques, **CO's–CO2**

Self-Learning Topics: Modern Encryption Standards and Applications.

UNIT III: **15 Hours**

Network Security: Network Security Concepts, Firewalls, Intrusion Detection Systems, Intrusion Prevention Systems, VPNs, Secure Socket Layer (SSL), Transport Layer Security (TLS). Wireless and Communication Security: Wireless Security Protocols, Email Security, Secure Communication Techniques, Network Monitoring and Packet Analysis. **CO's–CO3**

Self-Learning Topics: Cloud Security and IoT Security Challenges

UNIT IV: **15 Hours**

System Security: Operating System Security, Malware, Viruses, Worms, Trojans, Ransomware, Antivirus Mechanisms, System Hardening Techniques. Web Security: Web Application Security, SQL Injection, Cross Site Scripting (XSS), Session Hijacking, Secure Coding Practices, Security Testing. **COs–CO4**

Self-Learning Topics: Mobile Security and Secure Software Development Practices.

UNIT V: **10 Hours**

Cyber Laws and Ethics: Cyber Crimes, Cyber Laws, Intellectual Property Rights, Ethical Hacking Concepts, Digital Forensics Basics. Security Management: Information Security Management System (ISMS), Disaster Recovery, Business Continuity Planning, Incident Response, Security Auditing and Compliance. **CO's–CO5**

Self-Learning Topics: Cyber Security Governance and Security Awareness Programs

Board of Studies : Computer Science and Engineering

Approved in BoS No : 03

Approved in ACM No: 03

Expert Talk (To be Delivered by SMEs from Industries) **CO's/ PO's / PSO's**

1. Information Security in Cloud Computing (Cloud Threats, Security Models, Identity and Access Management in Cloud Platforms) CO1–CO5, PO1, PO2, PO3, PO11, PSO1, PSO2
2. Ethical Hacking and Penetration Testing (Security Assessment Tools, Vulnerability Analysis, Real-Time Attack Demonstrations) CO1–CO5, PO1, PO2, PO3, PO11, PSO1, PSO2
3. Cyber Security in Industry Applications (Banking Security, Healthcare Data Protection, Enterprise Security Management) CO1–CO5, PO1, PO2, PO3, PO11, PSO1, PSO2

Text Books:

1. Cryptography and Network Security – William Stallings, 7th Edition, Pearson.
2. Information Security Principles and Practices – Mark Stamp, 3rd Edition, Wiley.
3. Cyber Security Essentials – James Graham, Richard Howard, Ryan Olson.

Reference Books:

1. Network Security Essentials – William Stallings.
2. Information Security: Principles and Practice – Deven Shah.

3. Cyber Security and Cyber Laws – Alfred Basta.
4. Principles of Information Security – Michael E. Whitman and Herbert J. Mattord.

Web References:

1. <https://nptel.ac.in/>
2. <https://www.coursera.org/>
3. https://www.tutorialspoint.com/computer_security/
4. <https://owasp.org/>

Internal Assessment Pattern

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
L1	35	--
L2	40	--
L3	25	25
L4	--	35
L5	--	25
L6	--	15
Total (%)	100	100

Sample Short and Long Answers questions of Various Cognitive Levels

L1: Remember

1. Define Information Security.
2. What is Confidentiality?
3. Define Vulnerability.
4. What is Cryptography?
5. What is a Firewall?
6. Define Malware.
7. What is Authentication?
8. Define Digital Signature.
9. What is SQL Injection?
10. Define Cyber Crime?

L2: Understand

1. Explain the goals of information security.
2. Describe the types of cyber-attacks.
3. Explain symmetric and asymmetric encryption.
4. Describe the role of firewalls in network security.
5. Explain how malware affects computer systems.
6. Describe secure communication protocols.
7. Explain the concept of ethical hacking.
8. Describe disaster recovery planning.

L3: Apply

1. Apply encryption techniques to secure data transmission.
2. Demonstrate how a firewall filters network traffic.
3. Apply authentication methods in a secure login system.
4. Demonstrate the process of vulnerability assessment.
5. Apply secure coding practices to prevent web attacks.
6. Implement access control mechanisms for system protection.

L4: Analysing

1. Analyze different types of cyber-attacks and their impact.
2. Compare symmetric and asymmetric cryptography.
3. Analyze firewall and IDS functionalities.
4. Compare different malware protection techniques.
5. Analyze web vulnerabilities and mitigation techniques.
6. Examine the effectiveness of incident response strategies.

L5: Evaluating

1. Evaluate the effectiveness of encryption algorithms.
2. Assess the importance of multifactor authentication.
3. Evaluate different network security mechanisms.
4. Assess the role of ethical hacking in organizations.
5. Evaluate cyber security policies in enterprise environments.
6. Assess the effectiveness of disaster recovery plans.

L6: Create

1. Design a secure communication system using cryptographic techniques.
2. Develop a basic intrusion detection model for network security.
3. Create a security policy for a software organization.
4. Design secure web application architecture.
5. Develop a cyber security awareness program for educational institutions.

**Chairperson
Board of Studies (CSE)**

R24CSSC05**Data Analytics and Visualization**
(CSE-Skill Oriented Course)**0 0 2 1****Course Objectives:**

1. To provide fundamental knowledge of NumPy and Pandas for creating, manipulating, and analyzing structured data.
2. To develop skills in data preprocessing and wrangling, including data cleaning, handling missing values, and transforming data for analysis.
3. To enable students to visualize and analyze data using various plotting techniques and data aggregation methods for meaningful insights.

Course Outcomes:

At the end of the course, students will be able to:

Course Code	Course Outcomes (COs)	Mapping with POs and PSOs												Dok	
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1		PSO2
R24CSSC 05.1	Understand the fundamental concepts of NumPy, Pandas, and data wrangling techniques for handling structured data.	2	1	-	-	-	-	-	-	-	-	-	-	-	L1
R24CSSC 05.2	Apply appropriate methods to clean, preprocess, and transform data using data wrangling techniques.	2	2	1	-	-	-	-	-	-	-	-	-	-	L2
R24CSSC 05.3	Use tools like NumPy, Pandas, Matplotlib, and Seaborn to perform data manipulation and visualization.	2	2	2	-	-	-	-	-	-	-	-	-	-	L3
R24CSSC 05.4	Analyze datasets using advanced visualization techniques and grouping operations to identify patterns and relationships.	2	3	2	-	-	-	-	-	-	-	-	-	-	L4
R24CSSC 05.5	Evaluate and design effective data analysis and visualization solutions for real-world problems.	3	3	3	-	-	-	-	-	-	-	-	-	-	L5

Developing the following programs:**Week 1:****CO's-CO1**

1. Write a Python program to create 1D and 2D NumPy arrays with a specified data type, and display the arrays along with their data types.
2. Write a Python program to create a NumPy array, display its shape, size, and dimensions, and access elements using indexing.

CO's-CO1

Week 2:

3. Write a Python program to create a 2D NumPy array, extract a subarray using slicing, and find its transpose.
4. Write a Python program to create a NumPy array, perform basic arithmetic operations, and extract specific elements using indexing and slicing. **CO's-CO1**

Week 3:

5. Write a Python program to create a NumPy array and reshape it into different dimensions, then display the original array, reshaped array, and their attributes (shape and size).
6. Write a Python program to create a Pandas Series and demonstrate indexing operations by accessing specific elements.

Week 4

CO's-CO1

7. Write a Python program to create a Data Frame, add a new column, and access specific rows and columns using indexing.
8. Write a Python program to read data from a CSV file into a Data Frame and then write it into an Excel file.

Week 5

CO's-CO1

9. Write a Python program to create a Data Frame with missing values, handle them appropriately, and perform aggregation operations like sum and mean.
10. Write a Python program to create two Data Frames, merge them using a common column, perform group by operation, and apply aggregation functions. **CO's-CO2**

Week 6

11. Write a Python program to load a dataset and perform basic data cleaning by handling missing values and correcting incorrect data types.
12. Write a Python program to clean and standardize inconsistent data formats in a dataset (such as date formats, text case, or column names). **CO's-CO2**

Week 7

13. Write a Python program to identify outliers in a dataset and handle them using appropriate methods.
14. Write a Python program to detect and remove duplicate records from a dataset. **CO's-CO2**

Week 8

15. Write a Python program to normalize and standardize numerical data in a dataset and display summary statistics after processing.
16. Write a Python program to create a line plot using Matplotlib with properly labeled axes, title, and legend

Week 9

CO's-CO3

17. Write a Python program to create scatter and bar plots in multiple subplots within a single figure using Matplotlib.
18. Write a Python program to create a histogram and box plot for a given dataset using Matplotlib **CO's-CO3**

Week 10

19. Write a Python program to create a pie chart and customize it with labels, colors.
20. Write a Python program to create multiple Seaborn plots (such as boxplot or scatterplot) and apply different color palettes and customizations. **CO's-CO3**

Week 11

21. Write a Python program to create a crosstab of categorical data and visualize the relationship between two numerical variables using a scatter plot.
22. Write a Python program to create a histogram and box plot to analyze the distribution of a dataset.

CO's-CO4

Week 12

23. Write a Python program to create a heatmap and a tree map to represent structured data visually.
24. Write a Python program to visualize data using a bump chart, waterfall chart, and waffle chart.

CO's-CO5

Week 13

25. Write a Python program to create a DataFrame and perform data aggregation operations such as sum, mean, and count on numerical columns.
26. Write a Python program to group data using the groupby() function and apply aggregation functions like sum and average on grouped data.

CO's-CO5

Week 14

27. Write a Python program to demonstrate the split-apply-combine approach by grouping a dataset and applying different aggregation functions to each group.
28. Write a Python program to create a pivot table from a dataset and perform aggregation operations on it.

Board of Studies: Computer Science and Engineering

Approved in BOS No:03

Approved in ACM No: 03

Text Books:

1. Introduction to Data mining by Vipin Kumar, Pang-Ning Tan, Michael Steinback, Anuj Karpatne;2018
2. Ashutosh Nandeshwar, Tableau Data Visualization Cookbook, 1e, Packt Publishing.
3. Introduction to statistics by PkGiri and Banerjee, Academic publishers
4. Python for Data Analysis by Wes McKinney, 2nd Edition, O'REILLY

Reference Books:

1. Fabio Nelli, Python Data Analytics, First edition.
2. Data Wrangling with Python: Tips and Tools to Make Your Life Easier, Jacqueline Kazil and Katharine Jarmul, O'Reilly
3. Interactive Data Visualization: Foundations, Techniques, and Applications, Ward, Grinstein Keim, Natick A. K. Peters Ltd.

Sample Short and Long Answers questions of Various Cognitive Levels

L1: Remember

1. Identify a NumPy array.
2. Recognize the concept of indexing in NumPy.
3. Define a Pandas Series.
4. Describe a DataFrame.
5. Explain the concept of data cleaning.
6. List the causes of duplicate data in a dataset.
7. Explain what a line plot is.
8. Describe Seaborn and its purpose.
9. Explain the purpose of a histogram.

10. Define data aggregation.

L2: Understand

1. Explain how NumPy arrays differ from Python lists.
2. Illustrate how indexing and slicing are used in NumPy arrays.
3. Explain the structure and components of a Pandas DataFrame.
4. Summarize the steps involved in reading and writing CSV and Excel files using Pandas.
5. Explain the need for data cleaning in data analysis.
6. Illustrate how missing data and duplicate data affect dataset quality.
7. Explain how different plots (line, bar, scatter) are used to represent data.
8. Describe how Seaborn improves data visualization compared to Matplotlib.
9. Explain how heatmaps and histograms help in understanding data patterns.
10. Explain how groupby operations help in analyzing datasets.

L3: Apply

1. Develop a Python program to create a 2D NumPy array, perform slicing, and reshape it into a different dimension.
2. Construct a Python program to create a DataFrame, add a new column, and perform aggregation operations like sum and mean.
3. Demonstrate a Python program to load a dataset and perform data cleaning by handling missing values and incorrect data types.
4. Build a Python program to create a line plot and bar plot with proper labels and legend.
5. Develop a Python program to group data using groupby() and create a pivot table.
6. Apply indexing and slicing techniques to extract required data from arrays and DataFrames.
7. Carry out suitable techniques to handle duplicates and normalize data in a dataset.
8. Modify visualization plots using different color palettes and styles.
9. Choose appropriate visualization techniques (heatmap, scatter plot, box plot) to represent data effectively.
10. Choose suitable visualization techniques to analyze relationships in data. Also, develop a Python program to create a heatmap and scatter plot for a dataset.

L4: Analysing

1. Analyze array transformations by developing a Python program to create a NumPy array, perform slicing, and compare original and transposed arrays.
2. Examine data operations by constructing a Python program to create a DataFrame and analyze aggregation results using groupby().
3. Investigate data quality issues by developing a Python program to identify and handle missing values and outliers in a dataset.
4. Analyze data representation by building a Python program to compare different plots (line, bar, scatter) for the same dataset.
5. Differentiate grouping techniques by creating a Python program to analyze results using groupby() and pivot tables.
6. Differentiate between NumPy arrays and Pandas DataFrames in terms of structure and data handling.
7. Analyze the impact of missing values, duplicates, and outliers on data quality.
8. Compare different plotting techniques and analyze their suitability for various types of data.
9. Examine advanced visualization techniques (heatmap, treemap, bump chart) and analyze their effectiveness.

10. Analyze relationships in a dataset using appropriate visualization techniques. Also, develop a Python program to compare a heatmap and box plot for the same dataset and interpret the results.

L5: Evaluating

1. Design a Python program to create and manipulate NumPy arrays for efficient data processing and justify the chosen operations.
2. Develop a Python program to construct a DataFrame and design a solution to handle missing data and perform aggregation effectively.
3. Formulate a Python program to clean a dataset by handling missing values, duplicates, and outliers, and justify the techniques used.
4. Design a Python program to create multiple plots using Matplotlib and Seaborn and justify the choice of visualization techniques.
5. Develop a Python program to design a data summarization approach using groupby(), pivot tables, and crosstab, and evaluate the results.
6. Evaluate the effectiveness of NumPy and Pandas in handling large-scale data processing tasks.
7. Justify the selection of appropriate data cleaning techniques for different types of data issues.
8. Evaluate different visualization techniques and justify the most suitable method for representing complex datasets.
9. Design a visualization strategy using advanced techniques (heatmaps, choropleth maps, treemaps) for real-world data analysis.
10. Evaluate different visualization methods to represent multi-dimensional data effectively. Also, develop a Python program to design a choropleth map and justify its usefulness for geospatial analysis.

**Chairperson
Board of Studies (CSE)**

Course Objectives:

1. To introduce the fundamental principles of quantum computing, including qubits, superposition, entanglement, and quantum measurement, and to build a strong foundation in quantum mechanics concepts relevant to computation.
2. To understand the design and operation of quantum gates and circuits, and to enable students to represent and manipulate quantum information using various quantum gate operations.
3. To study key quantum algorithms such as Deutsch, Deutsch–Jozsa, and Grover’s algorithm, and to analyze their computational advantages over classical algorithms.
4. To understand the challenges of quantum noise and decoherence, and to explore quantum error correction techniques along with applications in quantum cryptography and secure communication.
5. To explore different physical models of quantum computing, analyze implementation challenges, and evaluate real-world applications and future directions of quantum technologies.

Course Outcomes

At the end of the course, students will be able to:

Course Code	Course Outcomes (COs)	Mapping with POs and PSOs														Dok
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2		
R24CSPE21 .1	Understand basic concepts of quantum computing and quantum mechanics	3	-	-	-	2	-	-	-	-	-	2	3	2	L1, L2	
R24CSPE21 .2	Apply quantum gates and circuits for computation	3	3	-	2	3	-	-	2	2	-	2	3	2	L2, L3	
R24CSPE21 .3	Analyze quantum algorithms and their efficiency	3	3	-	2	3	-	-	-	-	-	2	3	3	L3, L4	
R24CSPE21 .4	Understand quantum error correction and cryptography	3	-	-	2	3	-	-	-	-	-	2	3	2	L4, L5	
R24CSPE21 .5	Evaluate quantum computing models and future trends	3	2	3	2	3	2	3	-	-	2	3	-	-	L5, L6	

SYLLABUS**UNIT I:****15 Hours**

Quantum Computing Fundamentals: Introduction to Quantum Computing, Classical vs Quantum Computing, Qubits, Bloch Sphere Representation, Quantum States and Superposition, Quantum Measurement, Density Matrices and Quantum Entanglement.

CO’s–CO1

Self-Learning Topics: Applications of Quantum Computing.

UNIT II: 14 Hours

Quantum Gates and Circuits: Quantum Gates: Pauli Gates (X, Y, Z), Hadamard Gate, Phase Shift and Rotation Gates, Controlled Gates (CNOT, Toffoli), Quantum Circuits Representation, Quantum Circuit Design Principles and Quantum Parallelism. **CO's-CO2**

Self-Learning Topics: IBM Qiskit basics.

UNIT III: 13 Hours

Quantum Algorithms: Deutsch Algorithm, Deutsch-Jozsa Algorithm, Grover's Search Algorithm, Shor's Algorithm, Quantum Fourier Transform and Complexity of Quantum Algorithms. **CO's-CO3**

Self-Learning Topics: Comparison with classical algorithms

UNIT IV: 15 Hours

Quantum Error Correction & Security: Quantum Noise and Decoherence, Quantum Error Correction Codes, Fault Tolerance, Quantum Cryptography, Quantum Key Distribution (QKD) and Security Challenges in Quantum Systems. **COs-CO4**

Self-Learning Topics: Post-quantum cryptography.

UNIT V: 12 Hours

Quantum Computing Models & Applications: Quantum Computing Models: NMR Model, Spintronics Model, Linear Optical Model. Physical Realization of Quantum Computers, Scalability Issues, Industrial Applications and Future of Quantum Computing. **CO's-CO5**

Self-Learning Topics: Quantum supremacy

Board of Studies : Computer Science and Engineering

Approved in BoS No: 03

Approved in ACM No: 03

Expert Talk (To be Delivered by SMEs from Industries) CO's/ PO's / PSO's

1. Quantum Computing in Cloud Platforms (IBM Q, Azure Quantum) CO1-CO5, PO1, PO2, PO3, PO11, PSO1, PSO2
2. Quantum Machine Learning CO1-CO5, PO1, PO2, PO3, PO11, PSO1, PSO2
3. Quantum Cryptography in Cyber security CO1-CO5, PO1, PO2, PO3, PO11, PSO1, PSO2
4. Real-time Applications of Quantum Computing CO1-CO5, PO1, PO2, PO3, PO11, PSO1, PSO2

Text Books:

1. Eric R. Johnston, Nic Harrigan – *Programming Quantum Computers*
2. Christine Corbett Moran – *Mastering Quantum Computing with IBM QX*

Reference Books:

1. Michael Nielsen & Isaac Chuang – *Quantum Computation and Quantum Information*
2. Phillip Kaye – *Introduction to Quantum Computing*

Web References:

1. NPTEL Quantum Computing Courses
2. IBM Quantum Experience
3. Qiskit Documentation

Internal Assessment Pattern

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
L1	35	--
L2	40	--
L3	25	25
L4	--	35
L5	--	25
L6	--	15
Total (%)	100	100

Sample Short and Long Answers questions of Various Cognitive Levels

L1: Remember

1. Define a qubit?
2. What is superposition in quantum computing?
3. Define quantum entanglement.
4. What is a quantum gate?
5. List different types of quantum gates.
6. What is a Bloch sphere?
7. Define quantum measurement.
8. What is a density matrix?
9. What is decoherence?
10. Define quantum circuit.

L2: Understand

1. Explain the concept of superposition with an example.
2. Describe quantum entanglement and its importance.
3. Explain how quantum measurement affects a quantum state.
4. Differentiate between classical bits and qubits.
5. Explain the working of a Hadamard gate.
6. Describe the role of quantum gates in computation.
7. Explain the concept of quantum parallelism.
8. Describe how quantum circuits are represented.
9. Explain decoherence and its impact on quantum systems.
10. Summarize the need for quantum computing over classical computing

L3: Apply

1. Apply a Hadamard gate to the state $|0\rangle$ and find the output.
2. Apply Pauli-X gate on $|1\rangle$ and determine the result.
3. Construct a simple quantum circuit using H and CNOT gates.
4. Demonstrate superposition using a quantum circuit.
5. Apply measurement on a qubit in superposition state.
6. Solve a basic problem using Deutsch algorithm.
7. Implement a two-qubit entanglement circuit.
8. Apply quantum gates to transform $|0\rangle$ into $|1\rangle$.
9. Demonstrate the effect of a phase shift gate.

10. Construct a basic circuit for quantum teleportation (conceptual)

L4: Analysing

1. Compare classical computing and quantum computing.
2. Analyze the behavior of a qubit under measurement.
3. Compare different quantum gates and their functions.
4. Analyze the role of entanglement in quantum algorithms.
5. Compare Deutsch and Deutsch–Jozsa algorithms.
6. Analyze Grover’s algorithm performance.
7. Examine the impact of decoherence on computation.
8. Compare different quantum circuit models.
9. Analyze advantages and limitations of quantum computing.
10. Compare classical search vs quantum search algorithms

L5: Evaluating

1. Evaluate the efficiency of quantum algorithms over classical algorithms.
2. Assess the security of quantum cryptography.
3. Evaluate the effectiveness of quantum error correction techniques.
4. Judge the feasibility of large-scale quantum computers.
5. Evaluate different quantum computing models.
6. Assess the impact of noise in quantum systems.
7. Evaluate scalability challenges in quantum computing.
8. Compare different quantum hardware technologies.
9. Evaluate the role of quantum computing in cyber security.
10. Justify whether quantum supremacy is achievable in near future.

L6: Create

1. Design a quantum circuit to create superposition.
2. Develop a quantum circuit to generate entanglement.
3. Create a simple quantum algorithm for a basic problem.
4. Design a circuit for quantum teleportation.
5. Develop a model for quantum error correction.
6. Create a hybrid classical-quantum algorithm.
7. Design a circuit using multiple quantum gates.
8. Develop a conceptual framework for scalable quantum systems.
9. Create a solution using Grover’s algorithm for search problems.
10. Design a new application of quantum computing in real-world scenarios

**Chairperson
Board of Studies (CSE)**

R24CSPE22

PRINCIPLES OF DEEP LEARNING
(Professional Elective-V)

3 0 0 3

Course Objectives:

1. Understand Boolean retrieval systems, inverted index construction, and text preprocessing techniques such as tokenization, normalization, stemming, and phrase querying.
2. Understand dictionary structures and efficient indexing techniques along with tolerant retrieval methods such as wildcard queries, spelling correction, and dynamic index construction.
3. Understand statistical properties of text data and apply compression techniques along with term weighting methods such as TF-IDF for effective information retrieval.
4. Understand the vector space model for document scoring and ranking using similarity measures, query representation, and document weighting schemes.
5. Understand evaluation methods of information retrieval systems and architecture of web search engines including crawling, indexing, ranking, and web analysis techniques.

Course Outcomes

At the end of the course, students will be able to:

Course Code	Course Outcomes (COs)	Mapping with POs and PSOs												Dok	
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1		PSO2
R24CSPE 22.1	Understand deep learning, regularization techniques, and optimization methods, enabling them to design and train effective deep learning models.	3	–	2	–	2	–	–	–	–	1	–	3	–	L1, L2
R24CSPE 22.2	Understand the convolutional neural networks	3	3	2	2	3	--	--	--	--	--	--	3	3	L2, L3
R24CSPE 22.3	Describe various sequence modeling techniques	3	3	2	2	3	--	--	--	--	--	--	3	3	L3, L4
R24CSPE 22.4	Design various real time deep learning applications	3	3	3	3	3	1			1	2	1	3	3	L4, L5
R24CSPE 22.5	Understand various types of Auto encoders and different techniques of Deep Generative Models.	3	3	3	2	3	--	--	--	--	1	--	3	3	L5, L6

SYLLABUS**UNIT I:****15 Hours**

Introduction to Deep Learning: Introduction - Curse of Dimensionality, Historical trends in Deep Learning, Deep feed - Networks, Gradient Learning, Hidden Units, Architecture design, back propagation, activation functions.

Regularization and Optimization: Regularization techniques- L1, L2, dataset augmentation, early stopping, dropout, surrogate loss functions, batch and mini batch algorithms. Batch normalization, Comparison of shallow learning and deep learning, Importance and Applications of Deep Learning.

CO's–CO1

Self-Learning Topics: Comparison of Machine Learning and Deep Learning, TensorFlow and PyTorch Introduction

UNIT II:

13 Hours

Convolutional Networks: The Convolution Operation, Pooling, Convolution, Basic Convolution Functions, Structured Outputs, Data Types, Efficient Convolution Algorithms, Random or Unsupervised Features, The Neuro-scientific Basis for Convolutional Networks, Dataset Augmentation, Image Captioning.

CO's–CO2

Self-Learning Topics: Popular CNN Architectures (AlexNet, VGG, ResNet), Real-Time Computer Vision Applications

UNIT III:

13 Hours

Sequence Modeling- Recurrent and Recursive Nets: Unfolding Computational Graphs, Recurrent Neural Networks, Bidirectional RNNs, Encoder-Decoder Sequence-to-Sequence Architectures, Deep Recurrent Networks, Recursive Neural Networks, Leaky Units, LSTM.

CO's–CO3

Self-Learning Topics: Speech Recognition using Deep Learning, Natural Language Processing Applications

UNIT IV:

15 Hours

Practical Methodology: Performance Metrics, Default Baseline Models, Determining Whether to Gather More Data, Selecting Hyper-parameters, Debugging Strategies, Case study: Multi-Digit Number Recognition.

Applications: Large-Scale Deep Learning, Computer Vision, Speech Recognition, Natural Language Processing, Other Applications.

COs–CO4

Self-Learning Topics: Deep Learning in Healthcare, Edge AI and Real-Time AI Systems

UNIT V:

13 Hours

Auto encoders: Under complete Auto encoders, Regularized Auto encoders, Representational Power, Layer Size and Depth Stochastic Encoders and Decoders, De-noising Auto encoders, Applications of Auto encoders.

Deep Generative Models:

Directed Generative Nets-Variable Auto encoders, Generative adversarial networks, Representation Learning.

CO's–CO5

Self-Learning Topics: Reinforcement Learning Basics, Ethical Issues in Generative AI

Board of Studies : Computer Science and Engineering

Approved in BoS No : 03

Approved in ACM No: 03

Expert Talk (To be Delivered by SMEs from Industries)

CO's/ PO's / PSO's

1. Modern Information Retrieval Systems in Industry

(Industrial applications of Boolean retrieval, inverted indexing, TF-IDF ranking, vector space model, spelling correction, web crawling, indexing, and large-scale search engines like Google and enterprise search systems)

Text Books:

1. Ian Good fellow, Yoshua Bengio, Aaron Courville, “Deep Learning”, MIT Press, 2016.

Reference Books:

1. Josh Patterson and Adam Gibson, “Deep learning: A practitioner's approach”, O'Reilly Media, First Edition, 2017.

2. Fundamentals of Deep Learning, Designing next-generation machine intelligence algorithms, Nikhil Buduma, O'Reilly, Shroff Publishers, 2019.

Web References:

1. NPTEL (SWAYAM): <https://nptel.ac.in>

Internal Assessment Patter

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
L1	35	--
L2	40	--
L3	25	25
L4	--	35
L5	--	25
L6	--	15
Total (%)	100	100

Sample Short and Long Answers questions of Various Cognitive Levels**L1: Remember**

1. Define inverted index. (Unit I)
2. What is tokenization? (Unit I)
3. Define wildcard query. (Unit II)
4. What is TF-IDF? (Unit III)
5. Define vector space model. (Unit IV)

L2: Understand

1. Explain Boolean retrieval model in information retrieval. (Unit I)
2. Describe stop word removal and normalization. (Unit I)
3. Explain k-gram indexes for wildcard queries. (Unit II)
4. Describe edit distance in spelling correction. (Unit II)
5. Explain document scoring in vector space model. (Unit IV)

L3: Apply

1. Apply inverted index to process a Boolean query. (Unit I)
2. Use stemming and lemmatization in text preprocessing. (Unit I)
3. Apply k-gram indexing for handling wildcard queries. (Unit II)
4. Compute TF-IDF values for a document collection. (Unit III)
5. Apply vector space model to rank search results. (Unit IV)

L4: Analyzing

1. Analyze Boolean retrieval vs ranked retrieval systems. (Unit I)
2. Compare stemming and lemmatization techniques. (Unit I)
3. Analyze different spelling correction methods. (Unit II)
4. Compare variable byte codes and gamma codes for compression. (Unit III)
5. Analyze query-document similarity in vector space model. (Unit IV)

L5: Evaluate

1. Evaluate the effectiveness of inverted index in search systems. (Unit I)

2. Assess the importance of spelling correction in search engines. (Unit II)
3. Evaluate TF-IDF weighting in information retrieval systems. (Unit III)
4. Assess advantages of vector space model over Boolean model. (Unit IV)
5. Evaluate web search engine evaluation metrics. (Unit V)

L6: Create

1. Design an inverted index system for document retrieval. (Unit I)
2. Develop a spelling correction system using edit distance. (Unit II)
3. Design a compression technique for posting lists. (Unit III)
4. Create a document ranking system using vector space model. (Unit IV)
5. Design a basic web search engine architecture. (Unit V)

**Chairperson
Board of Studies (CSE)**

R24CSPE23**UX/UI Design****3 0 0 3**

(Professional Elective-V)

Course Objectives:

The objectives of this course for the student are to:

1. To introduce the fundamental concepts of UX/UI design, design thinking, and user-centered design principles.
2. To enable students to understand and apply user research techniques and visual design principles for identifying user needs.
3. To develop skills in structuring information architecture and creating wireframes for effective user experience.
4. To provide knowledge of interaction design principles and hands-on experience with prototyping tools.
5. To equip students with skills to perform usability testing, design responsive interfaces, and build professional design portfolios.

Course Outcomes

At the end of the course, students will be able to:

Course Code	Course Outcomes (COs)	Mapping with POs and PSOs														Domain
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2		
R24CSPE 23.1	Recall and explain UX/UI design concepts, design thinking stages, and user-centered design principles.	2	1	1	-	1	-	-	1	1	1	-	1	1	L1, L2	
R24CSPE 23.2	Apply user research methods and visual design principles to identify user requirements and design elements.	2	3	2	2	1	2	1	2	2	1	-	2	1	L2, L3	
R24CSPE 23.3	Analyze and design information architecture, user flows, and wireframes for applications and websites	2	2	3	2	2	1	1	1	3	2	1	3	2	L3, L4	
R24CSPE 23.4	Develop interactive prototypes using design tools and apply interaction design principles.	2	2	3	2	3	1	1	1	3	2	1	3	3	L4, L5	
R24CSPE 23.5	Evaluate usability of interfaces and create responsive UI designs and professional design portfolios.	2	3	3	3	2	2	2	2	3	2	1	3	3	L5, L6	

SYLLABUS**UNIT I:****14 Hours****Introduction to UX/UI Design Tools**

Definition of UX and UI Design, The design process, User-centered design principles, Importance of UX/UI design in the digital world. **CO's–CO1**

Self-Learning Topics: Case studies of good and bad UX, Understand design fundamentals

UNIT II:

15 Hours

User Research in Practice

Conducting user interviews and surveys for a chosen design project. Analyzing user data to identify key insights and pain points. Visual Design: Design principles (balance, hierarchy, contrast, etc.), Typography and color theory, Visual identity and branding. **CO's–CO2**

Self-Learning Topics: User research methods, Understand user needs & visual aesthetics

UNIT III:

15 Hours

Information Architecture: Website and application structure, Content organization and navigation systems, Card sorting and sitemaps, and developing user personas based on user research data. Creating user scenarios to illustrate user needs and interactions. Creating low-fidelity wireframes to illustrate the basic layout and functionality. Applying design principles for usability and user flow. Visual Design Principles, implementing design principles (e.g., typography, color theory) to create visually appealing interfaces. **CO's–CO3**

Self-Learning Topics: Usability principles & user flow, Structure user-friendly systems

UNIT IV:

15 Hours

Interaction Design and Site Mapping: User interface elements and their functionality, Interaction design principles (affordance, feedback, etc.), Wireframing and prototyping techniques developing sitemaps and user flows for a chosen application/website. Evaluating the effectiveness of information architecture for intuitive navigation. Design Tools and Resources: Introduction to industry-standard design software (e.g., Figma, Sketch, and Adobe XD). **COs–CO4**

Self-Learning Topics: User flows & sitemap creation, Develop interactive designs

UNIT V:

13 Hours

Prototyping and User Testing: Building interactive prototypes using design software functionalities. Conducting usability testing with users and iterating on designs based on feedback. Developing a basic style guide for visual consistency. Design for Different Platforms (Apply, Create) Exploring design considerations for responsive web and mobile interfaces. Adapting UI design for touch-based interactions. Design Presentation and Portfolio (Communicate, Create) Learning effective techniques for presenting design ideas and prototypes. Building a design portfolio showcasing the completed projects from the course. **COs–CO5**

Self-Learning Topics: Portfolio development & presentation, Build complete UX solutions

Board of Studies : Computer Science and Engineering

Approved in BoS No : 03

Approved in ACM No:03

Expert Talk (To be Delivered by SMEs from Industries)

CO's/ PO's / PSO's

1. UX in Product-Based Companies (like Google, Amazon)
2. Design Systems and Scalable UI Development
3. Accessibility in UX Design (Inclusive Design)

Text Books:

1. Donald Norman, The Design of Everyday Things, Basic Books.
2. Steve Krug, Don't Make Me Think, Pearson Education.
3. Jesse James Garrett, The Elements of User Experience, New Riders.

Reference Books:

1. Alan Cooper, About Face: The Essentials of Interaction Design, Wiley.
2. Jeff Gothelf, Lean UX, O'Reilly.
3. William Lidwell, Universal Principles of Design, Rockport.
4. Bill Buxton, Sketching User Experiences, Morgan Kaufmann.
5. Susan Weinschenk, 100 Things Every Designer Needs to Know About People, New Riders.

Web References:

1. www.interaction-design.org
2. www.nngroup.com
3. www.figma.com
4. www.coursera.org
5. www.dribbble.com

Internal Assessment Pattern

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
L1	35	--
L2	40	--
L3	25	25
L4	--	35
L5	--	25
L6	--	15
Total (%)	100	100

Sample Short and Long Answers questions of Various Cognitive Levels**L1: Remember**

1. Define UX design.
2. Define UI design.
3. List design thinking stages.
4. What is a wireframe?
5. What is a prototype?
6. Define usability.
7. What is typography?
8. What is color theory?
9. Define user persona.
10. What is a sitemap?

L2: Understand

1. Explain UX vs UI.
2. Describe user-centered design.
3. Explain importance of UX/UI in applications.
4. Describe visual hierarchy.
5. Explain usability testing.
6. Discuss role of user research.

7. Explain navigation design.
8. Describe interaction design principles.
9. Explain responsive design.
10. Discuss importance of prototypes.

L3: Apply

1. Apply user research methods to a problem.
2. Create a user persona.
3. Design a simple wireframe.
4. Apply color theory in UI design.
5. Create a sitemap for a website.
6. Develop a low-fidelity prototype.
7. Apply usability principles to improve UI.
8. Use Figma to design a screen.
9. Apply typography principles.
10. Create a user flow diagram.

L4: Analysing

1. Analyze a website's UX design.
2. Compare good vs bad UI design.
3. Analyze navigation structure of an app.
4. Examine usability issues in an interface.
5. Compare different wireframing approaches.
6. Analyze user feedback data.
7. Examine interaction design flaws.
8. Analyze visual hierarchy in a UI.
9. Compare design tools.
10. Analyze responsive design challenges.

L5: Evaluating

1. Evaluate usability of a mobile app.
2. Justify design choices in UI.
3. Evaluate effectiveness of a prototype.
4. Critique an existing website design.
5. Evaluate user research methods.
6. Assess UI consistency.
7. Evaluate accessibility in design.
8. Justify color scheme selection.
9. Evaluate navigation efficiency.
10. Assess design improvements.

L6: Create

1. Design a complete UX solution for an app.
2. Create a high-fidelity prototype.
3. Develop a responsive UI design.
4. Design a user journey map.
5. Create a UI for a real-world problem.
6. Develop a design portfolio.
7. Design a mobile application interface.

8. Create a usability testing plan.
9. Develop a branding style guide.
10. Design an innovative UX solution aligned with user needs.

**Chairperson
Board of Studies (CSE)**

Course Objectives:

1. To provide students with foundational knowledge of Computer Vision, including image formation, image representation, pre-processing techniques, colour image processing, and the overall computer vision pipeline for solving real-world visual analysis problems.
2. To provide students with knowledge of feature detection, description, and matching techniques used in Computer Vision, including invariant feature extraction, feature tracking, robust matching using RANSAC, and applications such as image stitching and panorama creation.
3. To provide students with an understanding of image segmentation fundamentals and techniques, including thresholding, region-based, clustering-based, and edge-based methods, along with an introduction to semantic and deep learning-based segmentation models and evaluation metrics used in Computer Vision applications.
4. To provide students with knowledge of image alignment and registration techniques, including holography estimation, feature-based alignment, robust matching using RANSAC, camera pose estimation, and the fundamentals of multi-view geometry, Structure from Motion (SfM), and SLAM for 3D computer vision applications.
5. To provide students with foundational knowledge of deep learning techniques for computer vision, including image classification, convolutional neural networks, transfer learning, object detection, face recognition, scene understanding, and Vision Transformers, along with their applications in autonomous driving, healthcare, and surveillance systems.

Course Outcomes:

At the end of the course, students will be able to:

Course Code	Course Outcomes (COs)	Mapping with POs and PSOs												Dok	
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1		PSO2
R24CSPE 24.1	Provide an introduction to computer vision including fundamentals of image formation.	3	2	1	-	2	-	-	-	-	-	1	2	1	L1, L2
R24CSPE 24.2	Enumerate the concepts of Feature detection and Matching	3	3	1	-	3	-	-	-	-	-	1	3	2	L2, L3
R24CSPE 24.3	Discuss about Image Segmentation Techniques	2	3	2	1	2	-	-	1	-	-	1	3	2	L3, L4
R24CSPE 24.4	Discuss applications of Feature based alignment like pose estimation.	2	2	3	2	3	1	-	1	-	-	1	3	3	L4, L5
R24CSPE 24.5	Discuss different recognition techniques.	2	3	2	2	3	1	1	1	2	1	3	3	3	L5, L6

SYLLABUS

UNIT I: Fundamentals of Computer Vision**15 Hours**

Introduction to Computer Vision and applications, Image formation basics and pinhole camera model, Image representation (pixels, resolution, bit depth), Sampling and quantization, Image preprocessing: filtering, noise removal, histogram equalization, Colour image processing (RGB, HSV basics), Computer Vision pipeline (end-to-end workflow).

CO's-CO1

Self-Learning Topics: Basics of linear algebra for vision, working with NumPy for image processing, Image file formats (JPEG, PNG, and TIFF) and compression basics.

UNIT II: Feature Extraction and Matching**14 Hours**

Feature detection (corners, blobs), Feature descriptors (SIFT, SURF, HOG – overview), Feature matching techniques and similarity measures, Feature invariance (scale, rotation, and illumination), Feature tracking basics, RANSAC for robust matching, Application: Image stitching (panorama creation)

CO's-CO2

Self-Learning Topics: Feature matching using Brute Force and FLANN matcher, Harris corner detector, Scale-space theory and Difference of Gaussian.

UNIT III: Image Segmentation**13 Hours**

Segmentation fundamentals, Thresholding techniques (Otsu method), Region-based segmentation (region growing), Clustering-based segmentation (K-means), Edge-based segmentation basics, Introduction to semantic segmentation, Deep learning-based segmentation (U-Net overview), Evaluation metrics (IoU, Dice coefficient)

CO's-CO3

Self-Learning Topics: Graph-based segmentation methods, medical image segmentation basics.

UNIT IV: Image Alignment and 3D Vision**12 Hours**

Feature-based alignment basics, Nomography estimation, RANSAC for alignment, Image registration techniques, Camera pose estimation, Introduction to multi-view geometry, Structure from Motion (SfM – basics), SLAM (Simultaneous Localization and Mapping – intro)

CO's-CO4

Self-Learning Topics: Epipolar geometry, Stereo vision and disparity map estimation, Augmented Reality basics using vision.

UNIT V: Object Recognition and Deep Learning:**13 Hours**

Image classification basics, Convolutional Neural Networks (CNNs), Transfer learning for vision tasks, Object detection (YOLO, SSD – overview), Face recognition using deep learning, Instance and category recognition, Scene understanding, Introduction to Vision Transformers (ViT), Applications: Autonomous driving, healthcare, surveillance

CO's-CO5

Self-Learning Topics: Object detection implementation using YOLO, Instance segmentation vs object detection, Edge AI.

Board of Studies : Computer Science and Engineering

Approved in BoS No : 03

Approved in ACM No:03

Text Books:

1. Richard Zaleski, "Computer Vision: Algorithms and Applications", Springer, 2010.
2. Rafael C. Gonzalez "Digital Image Processing", Pearson Education; Fourth edition (2018)

Reference Books:

1. Forsyth /Ponce, "Computer Vision: A Modern Approach", Pearson Education India; 2nd edition (2015)
2. S. Nagabhushana, "Computer Vision and Image processing", New Age International Pvt Ltd; First edition (2005)

Web References:

1. [Computer Vision Tutorial – Geeks for Geeks](#)

Internal Assessment Pattern

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
L1	35	--
L2	40	--
L3	25	25
L4	--	35
L5	--	25
L6	--	15
Total (%)	100	100

Sample Short and Long Answers questions of Various Cognitive Levels**L1: Remember**

1. Define Computer Vision and list its applications.
2. What is image segmentation?
3. Define feature detection and give two examples.
4. What is homography in image alignment?
5. What is CNN?
6. Explain image representation (pixels, resolution, bit depth).
7. Describe types of feature descriptors (SIFT, SURF, HOG).
8. Explain thresholding and Otsu method.
9. Define RANSAC and its purpose.
10. Explain tomography estimation.
11. Describe CNN architecture basics.
12. Explain object detection methods (YOLO, SSD).
13. Define Structure from Motion (SfM).

L2: Understand

1. Explain pinhole camera model.
2. Explain feature invariance.
3. Differentiate segmentation and classification.
4. Explain camera pose estimation.
5. Explain transfer learning.
6. Explain sampling and quantization with examples.
7. Describe feature matching techniques.
8. Explain K-means clustering in segmentation.
9. Explain image registration techniques.
10. Explain CNN working with layers.
11. Explain histogram equalization.
12. Explain semantic vs instance segmentation.
13. Explain multi-view geometry basics.

L3: Apply

1. Apply filtering to remove noise in an image.

2. Apply Harris corner detection steps.
3. Apply thresholding on given image data.
4. Apply RANSAC for matching.
5. Apply CNN for classification problem.
6. Demonstrate image preprocessing steps.
7. Apply feature matching using brute force method.
8. Solve segmentation using K-means clustering.
9. Apply homography for image stitching.
10. Implement object detection pipeline.
11. Apply histogram equalization for contrast enhancement.
12. Apply region growing segmentation.
13. Apply transfer learning for image classification.

L4: Analysing

1. Compare RGB and HSV color models.
2. Compare SIFT and SURF.
3. Compare segmentation techniques.
4. Compare SLAM and SfM.
5. Compare YOLO and SSD.
6. Analyze computer vision pipeline.
7. Compare feature detectors and descriptors.
8. Analyze clustering vs region-based segmentation.
9. Analyze epipolar geometry.
10. Analyze CNN vs Vision Transformer.
11. Compare image filtering techniques.
12. Analyze RANSAC performance.
13. Compare semantic and instance segmentation.

L5: Evaluating

1. Justify importance of preprocessing.
2. Evaluate feature matching accuracy.
3. Evaluate segmentation metrics (IoU, Dice).
4. Evaluate camera calibration importance.
5. Evaluate deep learning in vision tasks.
6. Evaluate different image formats (JPEG, PNG).
7. Justify need for invariant features.
8. Evaluate segmentation methods performance.
9. Evaluate homography limitations.
10. Evaluate CNN performance in classification.
11. Evaluate filtering techniques for noise removal.
12. Evaluate RANSAC vs traditional methods.
13. Evaluate object detection models in real-world applications.

L6: Create

1. Design simple CV pipeline.
2. Design feature extraction workflow.
3. Design segmentation approach for medical images.
4. Design AR application using vision.
5. Design CNN model for classification.

6. Develop complete image processing system.
7. Design feature-based panorama stitching system.
8. Create segmentation model using U-Net.
9. Develop SLAM-based navigation system.
10. Design objects detection system using YOLO.
11. Create end-to-end CV pipeline for autonomous driving.
12. Design stereo vision system for depth estimation.
13. Develop real-time face recognition system.

**Chairperson
Board of Studies (CSE)**

Course Objectives:

1. To impart basic concepts of computer architecture and organization.
2. To explain key skills of constructing cost-effective computer systems.
3. To familiarize students with basic CPU organization.
4. To help students understand various memory devices.
5. To facilitate students in learning I/O communication and multiprocessors.

Course Outcomes:

At the end of the course, students will be able to:

Course Code	Course Outcomes (COs)	Mapping with POs and PSOs														Dok
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2		
R24CSOE 16.1	Understand computer structure and arithmetic operations.	3	-	2	1	2	-	-	-	-	1	-	3	1	L1, L2	
R24CSOE 16.2	Describe instruction formats and CPU organization.	3	3	2	2	2	-	-	2		1	-	3	2	L2, L3	
R24CSOE 16.3	Apply register transfer and micro-operations.	3	3	3	2	3	-	-	-	-	1	-	3	2	L3, L4	
R24CSOE 16.4	Analyze memory hierarchy and memory management techniques.	3	-	3	3	3	-	-	-	-	1	-	3	2	L4, L5	
R24CSOE 16.5	Explain I/O organization and multiprocessor systems.	3	2	2	2	3	2	3	-	1	1	1	2	2	L5, L6	

SYLLABUS**UNIT I:****15 Hours**

Structure of computers & computer arithmetic: Computer types, Functional units, Basic operational concepts, Von Neumann Architecture, Bus Structures, Software, Performance, Multiprocessors and Multicomputer, Data representation, Fixed and Floating point, Error detection and correction codes. Addition and Subtraction, Multiplication and Division algorithms, Floating-point Arithmetic Operations, Decimal arithmetic operations.

CO's-CO1

Self-Learning Topics: Floating point representation, Error correction techniques, Number systems

UNIT II:**13 Hours**

Basic Computer Organization and Design: Instruction codes, Computer Registers, Computer Instructions and Instruction cycle, Timing and Control, Memory-Reference Instructions, Input-Output and interrupt, Stack organization, Instruction Formats, Addressing Modes, Data Transfer and Manipulation, CISC and RISC architecture.

CO's-CO2

Self-Learning Topics: Modern processors, Instruction pipelining basics, Addressing mode examples

UNIT III:

14 Hours

Register Transfer and Micro-Operations: Register Transfer Language, Register Transfer, Bus and Memory Transfers, Arithmetic Micro-Operations, Logic Micro-Operations, Shift Micro-Operations, Arithmetic logic shift unit. Micro-programmed Control, Control Memory, Address Sequencing, Design of Control Unit.

CO's–CO3

Self-Learning Topics: RTL examples, Control unit design, Microprogram sequencing

UNIT IV:

13 Hours

Memory System: Memory Hierarchy, Semiconductor Memories, RAM, ROM, Types of ROM, Cache Memory, Virtual memory, Paging, Secondary Storage, RAID.

CO's–CO4

Self-Learning Topics: Cache mapping techniques, Virtual memory implementation, RAID levels

UNIT V:

13 Hours

Input Output and Multiprocessors: I/O interface, Programmed IO, Memory Mapped IO, Interrupt Driven IO, DMA. Characteristics of multiprocessors, Interconnection structures, Inter Processor Arbitration, Communication and Synchronization, Cache Coherence.

CO's–CO5

Self-Learning Topics: DMA controllers, Cache coherence protocols, Parallel processing concepts

Board of Studies : Computer Science and Engineering

Approved in BoS No : 03

Approved in ACM No: 03

Expert Talks:

1. Modern Processor Architectures and CPU Design CO's/PO's/PSO's Mapped: CO2, CO3, CO5, PO1, PO2, PO3, PO5
2. Memory Systems and High-Performance Computing CO's/PO's/PSO's Mapped: CO4, CO5, PO1, PO2, PO3, PO5
3. I/O Systems and Embedded Computing Applications CO's/PO's/PSO's Mapped: CO1, CO5, PO1, PO3, PO5

Text Books:

1. M. Morris Mano, Computer System Architecture, 3rd Edition, Pearson/PHI.
2. Carl Hamacher, Zvonko Vranesic, Safwat Zaky, Computer Organization, McGraw Hill.

Reference Books:

1. William Stallings, Computer Organization and Architecture, Prentice Hall.
2. Andrew S. Tanenbaum, Structured Computer Organization, Pearson Education.
3. John P. Hayes, Computer Architecture and Organization, Tata McGraw Hill.

Internal Assessment Pattern

Cognitive Level	Internal Assessment–1 (%)	Internal Assessment–2 (%)
L1 (Remember)	35	-
L2 (Understand)	40	-
L3 (Apply)	25	25
L4 (Analyze)	-	35

L5 (Evaluate)	-	25
L6 (Create)	-	15

Sample Questions of Various Cognitive Levels

L1: Remember

1. Define Computer Architecture.
2. What is Von Neumann Architecture?
3. Define cache memory.
4. What is a register?
5. Define instruction cycle.
6. What is micro-operation?
7. Define RAM and ROM.
8. What is DMA?
9. Define interrupt.
10. What is multiprocessor system?

L2: Understand

1. Explain the functional units of a computer.
2. Describe floating-point representation.
3. Explain instruction formats.
4. Describe addressing modes.
5. Explain stack organization.
6. Describe register transfer language.
7. Explain memory hierarchy.
8. Describe cache memory organization.
9. Explain programmed I/O.
10. Describe inter processor communication.

L3: Apply

1. Perform binary addition using 2's complement method.
2. Apply Booth's multiplication algorithm for given numbers.
3. Demonstrate instruction cycle operation.
4. Apply RTL statements for data transfer.
5. Perform arithmetic micro-operations.
6. Construct memory address mapping.
7. Apply paging technique for virtual memory.
8. Demonstrate interrupt handling process.
9. Implement DMA transfer operation.
10. Apply cache mapping techniques.

L4: Analyze

1. Compare fixed-point and floating-point representations.
2. Analyze CISC and RISC architectures.
3. Compare memory-reference and I/O instructions.
4. Analyze shift micro-operations.

5. Compare hardwired and microprogrammed control units.
6. Analyze cache memory performance.
7. Compare different RAID levels.
8. Analyze virtual memory techniques.
9. Compare programmed I/O and interrupt-driven I/O.
10. Analyze multiprocessor interconnection structures.

L5: Evaluate

1. Evaluate the performance of cache memory.
2. Assess the advantages of RISC architecture.
3. Evaluate instruction pipelining techniques.
4. Assess memory hierarchy efficiency.
5. Evaluate DMA over programmed I/O.
6. Judge the effectiveness of virtual memory.
7. Evaluate different cache coherence protocols.
8. Assess various processor communication methods.
9. Evaluate secondary storage performance.
10. Assess multiprocessor synchronization methods.

L6: Create

1. Design a simple arithmetic logic unit (ALU).
2. Create a register organization for a CPU.
3. Design instruction formats for a processor.
4. Develop a micro programmed control unit.
5. Create memory hierarchy for a computer system.
6. Design cache memory organization.
7. Develop interrupt handling mechanism.
8. Create a DMA controller model.
9. Design a multiprocessor interconnection network.
10. Develop a basic computer architecture model.

Chairperson
Board of Studies (CSE)

R24CSPOE17 COMPUTER FORENSICS AND LAW PROTECTION 3 0 0
3

(Open Elective Course-IV)

Course Objectives:

- 1.To correctly define and identify appropriate applications of computer forensics and cyber security concepts.
- 2.To understand and analyze computer forensic evidence collection and investigation methodologies.
- 3.To identify essential and up-to-date concepts, algorithms, tools, protocols, and procedures in computer forensics.
- 4.To design procedures at suspected crime scenes ensuring digital evidence integrity and preservation.
- 5.To understand cyber laws, digital crimes, and legal frameworks associated with digital investigations.

Course Outcomes:

At the end of the course, students will be able to:

Course Code	Course Outcomes (COs)	Mapping with POs and PSOs														Dok
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2		
R24CSOE 17.1	Understand concepts of cyber security, cyber threats, and computer forensics fundamentals	3	-	-	-	2	-	-	-	-	-	2	3	2	L1, L2	
R24CSOC 17.2	Analyze cybercrimes, evidence collection methods, and digital forensic investigation procedures	3	3	-	2	3	-	-	2	-	-	2	3	2	L2, L3	
R24CSOE 17.3	Apply digital forensic tools and methodologies for e-mail and social media investigations	3	3	2	2	3	-	-	-	2	-	2	3	3	L3, L4	
R24CSOE 17.4	Analyze information warfare, cybercrime issues, identity theft, and legal implications.	3	2	2	2	3	-	-	-	-	-	2	3	2	L4, L5	
R24CSOE 17.5	Evaluate cyber laws, forensic case handling, digital signatures, and e-commerce security mechanisms.	3	2	3	2	3	2	2	-	-	2	3	-	-	L5, L6	

SYLLABUS

UNIT I

15 Hours

Introduction: Computer Forensics Fundamentals, Types of Computer Forensics Technology, Types of Computer Forensics Systems, Vendor and Computer Forensics Services. Cyber Security Overview: Cyber Threats, Vulnerabilities, Cyber Attacks, Security Challenges, Digital Crime Concepts.

CO's-CO1

Self-Learning Topics: Recent Cyber Security Incidents and Emerging Threats.

UNIT II

15 Hours

Computer Forensics Evidence and Capture: Data Recovery, Evidence Collection and Data Seizure, Duplication and Preservation of Digital Evidence, Computer Image Verification and Authentication.

Current Digital Forensics Tools: Evaluating Digital Forensics Tool Needs, Digital Forensics Software Tools, Command-Line Forensics Tools, Digital Forensics Hardware Tools, Validating and Testing Forensics Software.

Self-Learning Topics: Modern Digital Investigation Platforms and Open-Source Forensic Tools.

UNIT III

15 Hours

Computer Forensic Analysis: Discovery of Electronic Evidence, Identification of Data, Reconstructing Past Events, fighting against Macro Threats, Information Warfare Arsenal, Tactics of Military, Terrorists, and Private companies.

E-mail and Social Media Investigations: Role of E-mail in Investigations, Client and Server Concepts in E-mail, Investigating E-mail Crimes and Violations, E-mail Servers, Specialized E-mail Forensics Tools, Social Media Investigation Techniques.

CO's-CO3

Self-Learning Topics: Cloud Forensics and Mobile Device Investigations.

UNIT IV

15 Hours

Information Warfare: Surveillance Tools, Hackers and Theft of Components, Contemporary Computer Crimes, Identity Theft and Identity Fraud, Organized Crime and Terrorism.

Legal Issues: Government Efforts, Prosecution Procedures, First Amendment and Computer Crimes, Fourth Amendment and Legal Issues in Digital Investigations.

COs-CO4

Self-Learning Topics: International Cyber Laws and Digital Privacy Regulations.

UNIT V

10 Hours

Computer Forensic Cases: Developing Forensic Capabilities, Searching and Seizing Computer Related Evidence, Processing Evidence and Report Preparation, Future Challenges in Digital Forensics.

Basics of Law and Technology: Indian IT Laws, Scope and Jurisprudence, Digital Signatures, E-Commerce Security, Mobile Communication Security, Smart Cards, Expert Systems, Possible Crime Scenarios and Legal Coverage.

CO's-CO5

Self-Learning Topics: Block chain Forensics and Crypto currency Crime Investigation.

Board of Studies : Computer Science and Engineering

Approved in BoS No : 03

Approved in ACM No: 03

Expert Talk (To be Delivered by SMEs from Industries)

CO's/ PO's / PSO's

- 1.Digital Forensics in Cyber Crime Investigation and Incident Response CO1-CO5, PO1, PO2, PO3, PO11, PSO1, PSO2
- 2.Ethical Hacking, Cyber Threat Intelligence, and Security Operations Centers (SOC) CO1-CO5, PO1, PO2, PO3, PO11, PSO1, PSO2
- 3.Cyber Laws, Digital Evidence Handling, and Legal Compliance in Organizations CO1-CO5, PO1, PO2, PO3, PO11, PSO1, PSO2

Text Books:

1. John R. Vacca, *Computer Forensics: Computer Crime Scene Investigation*, Cengage Learning, 2nd Edition, 2005.
2. Nelson, Phillips, Stuart, *Guide to Computer Forensics and Investigations*, 6th Edition, Cengage.
3. Marjie T. Britz, *Computer Forensics and Cyber Crime: An Introduction*, Pearson Education, 2nd Edition, 2008.
4. Doctrine of IT Act of India, Government of India Publication, 2000

Reference Books:

1. Marie-Helen Maras, *Computer Forensics: Cybercriminals, Laws, and Evidence*, Jones & Bartlett Learning, 2nd Edition, 2014.
2. Eamon P. Doherty, *Digital Forensics for Handheld Devices*, CRC Press.
3. Chad Steel, *Windows Forensics*, Wiley, 1st Edition, 2006.
4. Majid Yar, *Cybercrime and Society*, SAGE Publications Ltd, 2nd Edition, 2013.
5. Robert M. Slade, *Software Forensics: Collecting Evidence from the Scene of a Digital Crime*, Tata McGraw Hill, 1st Edition, 2004.

Web References:

1. <https://nptel.ac.in/>
2. <https://www.coursera.org/>
3. <https://www.cybrary.it/>
4. <https://owasp.org/>

Internal Assessment Pattern

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
L1	35	--
L2	40	--
L3	25	25
L4	--	35
L5	--	25
L6	--	15
Total (%)	100	100

Sample Short and Long Answers questions of Various Cognitive Levels

L1: Remember

1. Define Computer Forensics.
2. What is Digital Evidence?
3. Define Cyber Crime.
4. What is Data Recovery?
5. Define Identity Theft.
6. What is E-mail Forensics?
7. Define Information Warfare.
8. What is Digital Signature?
9. Define Smart Card.

10. What is Cyber Security? .

L2: Understand

1. Explain the fundamentals of computer forensics.
2. Describe the process of evidence collection and preservation.
3. Explain the role of forensic tools in investigations.
4. Describe cybercrime reporting procedures.
5. Explain e-mail investigation techniques.
6. Describe identity fraud and its impact.
7. Explain the role of Indian IT laws in cybercrime prevention.
8. Describe legal challenges in digital investigations

L3: Apply

1. Demonstrate the process of digital evidence collection.
2. Apply forensic tools for analyzing suspicious files.
3. Implement procedures for preserving digital evidence.
4. Apply e-mail forensic methods in cybercrime investigations.
5. Demonstrate the use of digital signatures in secure communication.
6. Apply cyber security measures to protect e-commerce systems.

L4: Analysing

1. Analyze different types of computer forensic systems.
2. Compare hardware and software forensic tools.
3. Analyze the role of surveillance tools in information warfare.
4. Compare cybercrime investigation methodologies.
5. Analyze legal issues associated with digital evidence.
6. Examine cyber threats related to social media communications.

L5: Evaluating

1. Evaluate the effectiveness of digital forensic tools.
2. Assess the impact of identity theft on organizations.
3. Evaluate cyber laws related to digital crimes.
4. Assess the role of government agencies in cybercrime prevention.
5. Evaluate the security challenges in e-commerce systems.
6. Assess preventive measures against cyber frauds.

L6: Create

1. Design a digital forensic investigation framework for cybercrime cases.
2. Develop a secure evidence preservation model for forensic analysis.
3. Create a cyber security awareness program for organizations.
4. Design a secure e-commerce transaction system with digital signatures.
5. Develop a forensic incident response plan for enterprise systems.

**Chairperson
Board of Studies (CSE)**