Shirpur Education Society's

R. C. Patel Institute of Technology, Shirpur

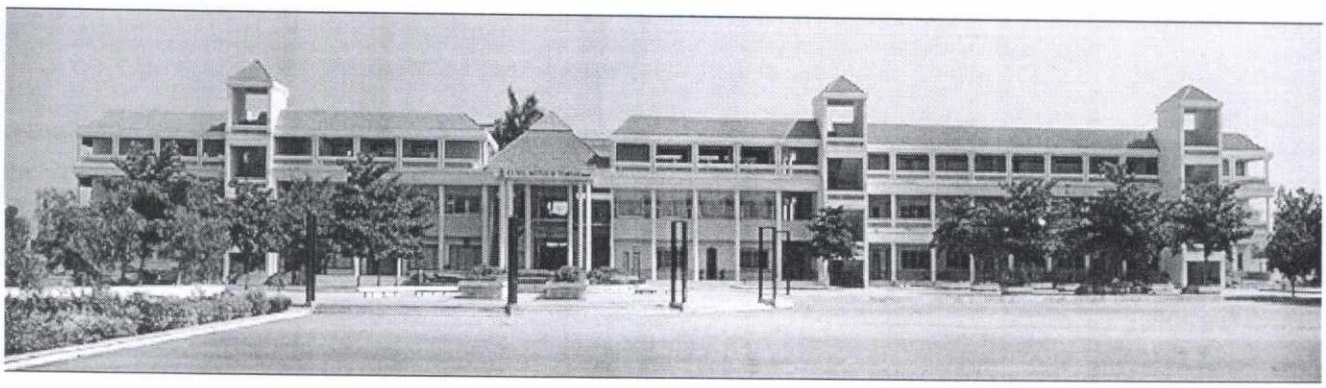
(An Autonomous Institute)

Course Structure and Syllabus

Third Year B. Tech

Artificial Intelligence and Machine Learning

With effect from Year 2025-26



Shahada Road, Near Nimzari Naka, Shirpur, Maharashtra 425405
Ph: 02563 259 802, Web: www.rcpit.ac.in

Third Year B. Tech Artificial Intelligence and Machine Learning Semester-V (w.e.f. 2025-26)

Sr	Course Category	Course Code	Course Title	Teaching Scheme			Evaluation Scheme					Total	Credit	
							Continuous Assessment (CA)							
							TA	Term Test 1 (TT1)	Term Test 2 (TT2)	Average of (TT1 & TT2)	ESE			
				L	T	P	[A]			[B]	[C]	[A+B+C]		
1	PC	RCP23ACPC501	Machine Learning	3			25	15	15	15	60	100	3	4
	PC	RCP23ALPC501	Machine Learning Laboratory			2	25				25	50	1	
2	PC	RCP23ACPC502	Natural Language Processing	3			25	15	15	15	60	100	3	4
	PC	RCP23ALPC502	Natural Language Processing Laboratory			2	25				25	50	1	
3	PC	RCP23ACPC503	Image Processing and Computer Vision	3			25	15	15	15	60	100	3	4
	PC	RCP23ALPC503	Image Processing and Computer Vision Laboratory			2	25				25	50	1	
4	MD	RCP23ACMD504	Computer Network	3			25	15	15	15	60	100	3	3
5	MD	RCP23ALMD505	Programming Laboratory-III (Full Stack Development using NextJs)			2	25				25	50	1	1
6@	PE	RCP23ACPE511	Human Machine Interaction	3			25	15	15	15	60	100	3	4
		RCP23ALPE511	Human Machine Interaction Laboratory			2	25				25	50	1	
		RCP23ACPE512	Advanced Data Structures and Algorithms	3			25	15	15	15	60	100	3	
		RCP23ALPE512	Advanced Data Structures and Algorithms Laboratory			2	25				25	50	1	
		RCP23ACPE513	Recommendation Systems	3			25	15	15	15	60	100	3	
		RCP23ALPE513	Recommendation Systems Laboratory			2	25				25	50	1	
7	HS	RCP23ITHSX06	Environmental Science Tutorial		1		25					25	1	1
8	SC	RCP23IPSC501	Semester Project-III			2	25				25	50	1	1
9	SC	RCP23ALSC506	Employability Skill Development Program-I			2	25				25	50	1	1
			Total	15	1	14	325			75	475	875	23	

Prepared by:
Dr. N. G. Shinde

Prof. Dr. D. R. Patil
BOS Chairman

Prof. Dr. P. J. Deore
Dean Academics/Dy. Director

Checked by:
Dr. S. P. Patil

Prof. S. P. Shukla
C.O.E.

Prof. Dr. J. B. Patil
Director

R. C. Patel Institute of Technology, Shirpur
SES Shirpur Education Society
Autonomous

Semester - V

Program: Artificial Intelligence and Machine Learning	T. Y. B.Tech	Semester: V
Machine Learning (RCP23ACPC501)		
Machine Learning Laboratory (RCP23ALPC501)		

Prerequisite: Knowledge of basic probability and statistics, Data Mining and Analytics concepts

Course Objective(s):

1. To understand key machine learning concepts: hypothesis formation, biasvariance trade-off, and model evaluation metrics
2. To master regression, classification, and clustering techniques
3. To apply machine learning algorithms to real-world datasets effectively

Course Outcomes:

On completion of the course, the learner will be able to:

CO	Course Outcomes	Blooms Level	Blooms Description
CO1	Explain the types of machine learning and issues in machine learning.	L2	Understand
CO2	Analyze model performance using evaluation metrics.	L4	Analyze
CO3	Implement and tune regression and classification algorithms.	L3	Apply
CO4	Apply knowledge of Bayesian learning principles.	L3	Apply
CO5	Apply association rule mining and clustering techniques.	L3	Apply
CO6	Explain the fundamentals of Neural Networks.	L2	Understand

Machine Learning (RCP23ACPC501)
Course Contents

Unit-I — 05 Hrs.

Introduction to Machine Learning:

Types of Machine Learning, Issues in Machine Learning, Application of Machine Learning, Steps involved in developing a Machine Learning Application, Hypothesis and Inductive Bias, Bias-Variance Trade-off, Performance measures, Data Validation.

Evaluation & Selection: Metrics for Evaluating Classifier Performance, Holdout Method and Random Subsampling, Cross Validation, Bootstrap, Model Selection Using Statistical Tests of Significance, Comparing Classifiers Based on Cost–Benefit and ROC Curves.

Unit-II — 08 Hrs.

Regression:

Linear Regression, Least Minimum Slope (LMS) algorithm, Gradient Descent, Lasso and Ridge Regression. Polynomial Regression. Logistic Regression, Maximum Likelihood Function.

Classification: Introduction to decision tree, Learning Decision tree using ID3 and Gini index; CART, Overfitting. Ensemble methods: Bagging (Random Forest) and Boosting (XG Boost).

Unit-III — 08 Hrs.

Bayesian Learning:

Bayesian Learning, Naïve Bayes, Bayesian Network: Representation in Bayesian, Belief Network, Inference in Bayesian Network, Applications of Bayesian Network. Classification Model.

Unit-IV — 05 Hrs.

Introduction to Support Vector Machine:

Support Vectors, Kernels: Linear, Polynomial and Radial Basis Function (RBF) Kernel Association Rule Mining: Market Basket Analysis, Apriori algorithm and measures of association.

Unit-V — 08 Hrs.

Clustering:

Cluster Analysis and Requirements of Cluster Analysis Partitioning Methods: k-Means, k-Medoids Hierarchical Methods: Agglomerative, Divisive. Dimensionality Reduction: Dimensionality Reduction Techniques: Principal Component Analysis.

Unit-VI — 05 Hrs.

Introduction to Neural Networks and Deep Learning:

Deep Learning applications, Association of biological neuron with artificial network, activation functions, weights, bias, threshold, learning rate, momentum factor.

McCulloch Pitts Neuron:

Theory and architecture; linear separability; Hebb Network: Theory and algorithm, ANN architectures. Hyperparameter tuning and batch normalization, Machine Learning vs Deep Learning.

Machine Learning Laboratory (RCP23ALPC501)

List of Laboratory Experiments

Suggested Experiments: (Any 08)

1. Perform Linear Regression.
    - Perform data cleaning
    - EDA
    - Data transformation
    - Model Training
    - Performance evaluation
2. Perform Logistic Regression.
    - Perform data cleaning
    - EDA
    - Data transformation
    - Model Training
    - Performance evaluation
3. Perform Decision Tree using GINI.
    - Data cleaning
    - EDA
    - Data transformation
    - Model Training, Visualize Decision Tree
    - Performance evaluation.
4. Perform CART decision tree algorithm
    - Data cleaning
    - EDA
    - Data transformation

- Model Training, Visualize Decision Tree
- Performance evaluation

5. Perform Ensemble methods
    - Data cleaning
    - EDA
    - Data transformation
    - Model Training
    - Performance evaluation
6. Perform Bayesian Classification
    - Data cleaning
    - EDA
    - Data transformation
    - Model Training
    - Performance evaluation
7. Compare performance of classification algorithms.
    - Model Training
    - Performance evaluation
    - Comparison of performance of different classification algorithms
8. Perform Support Vector Machine.
    - Data cleaning
    - EDA
    - Data transformation
    - Dimensionality reduction
9. Perform K-means/ K-Medoids clustering.
    - Data cleaning
    - EDA
    - Data transformation
    - Clustering
10. Study a machine learning patent.
11. Mini project based on any machine learning application.

Any other experiment based on syllabus may be included which would help the learner to understand topic/concept.

Oral examination will be based on the entire syllabus, including the practicals performed during laboratory sessions.

Text Books:

1. Mitchell, T. M., "Machine Learning", 1st Edition, McGraw Hill, 2017.
2. Ethem Alpaydın, "Introduction to Machine Learning", 4th Edition, MIT Press, 2020.
3. Peter Harrington, "Machine Learning In Action", 1st Edition, DreamTech Press, 2012.
4. Bruce, P., Bruce, A., & Gedeck, P., "Practical statistics for data scientists: 50+ essential concepts using R and Python", 2nd Edition, O'Reilly Media, 2020.

Reference Books:

1. "Data Mining for Business Analytics (An Indian Adaptation): Concepts, Techniques and Applications in Python", Cambridge University Press, ISBN NO. 978-1108727747, 2019.
2. Andreas C. Müller and Sarah Guido , "Introduction to Machine Learning with Python: A Guide for Data Scientists", O'reilly, 2016.
3. Stephen Marsland, "Machine Learning an Algorithmic Perspective", CRC Press, 2015.
4. Han Kamber, "Data Mining Concepts & Techniques", Morgann Kaufmann Publishers, 2012.
5. Kevin P. Murphy, "Machine Learning A Probabilistic Perspective", 2012.

Web Links:

1. https://onlinecourses.nptel.ac.in/noc21_cs06/preview
2. https://www.datacamp.com/tutorial/tableau-tutorial-for-beginners
3. https://www.kaggle.com/code/ekami66/detailed-exploratory-data-analysis-with-python

Program: Artificial Intelligence and Machine Learning	T. Y. B.Tech	Semester: V
Natural Language Processing (RCP23ACPC502)		
Natural Language Processing Laboratory (RCP23ALPC502)		

Prerequisite: Python Programming

Course Objective(s):

1. To introduce the fundamental concepts and techniques of Natural Language Processing for analyzing words based on Morphology and CORPUS.
2. To examine the NLP models and interpret algorithms for classification of NLP sentences by using both the traditional, symbolic and the more recent statistical approach.
3. To get acquainted with the algorithmic description of the main language levels that includes morphology, syntax, semantics, and pragmatics for information retrieval and machine translation applications.

Course Outcomes:

On completion of the course, the learner will be able to:

CO	Course Outcomes	Blooms Level	Blooms Description
CO1	Understand the Principles and Process of Natural Languages and real-world applications.	L2	Understand
CO2	Demonstrate understanding of state-of-the-art algorithms and techniques for text-based processing of natural language with respect to morphology.	L3	Apply
CO3	Perform POS tagging for a given natural language and select a suitable language modelling technique based on the structure of the language.	L6	Create
CO4	Check the syntactic and semantic correctness of sentences using grammars and labelling.	L5	Evaluate

Natural Language Processing (RCP23ACPC502) Course Contents

Unit-I — 04 Hrs.

Introduction to Natural Language Processing:

Origin & History of NLP, Stages in NLP, Ambiguities and its types in English and Indian Regional Languages; Applications of NLP- Machine Translation, Information Retrieval, Question Answering System, Sentiment Analysis, Text Categorization , Text Summarization, Named Entity Recognition.

Unit-II — 04 Hrs.

Computational tools for text analysis:

Basic Terms: Tokenization, Tokenization in the NLTK, Tokenizing text, Stemming, Lexicon free FST Porter Stemmer algorithm, Lemmatization, Natural Language Toolkit (NLTK): Corpora and other data resources. Uses of corpora: Lexicography, Grammar and syntax, Stylistics, Training and evaluation. Basic corpus analysis: Frequency distribution building and analyzing a corpus.

Unit-III — 09 Hrs.

Word Level Analysis (statistical language model):

Inflectional Morphology, Derivational Morphology, Regular expression with types, Finite State Automata, NFA and DFA, Morphological Models: finite state morphology, Morphological parsing with FST (Finite State Transducer), Grams and its variation: Bigram, Trigram, Simple (Unsmoothed) N-grams; N-gram Sensitivity to the Training Corpus, Evaluating N-grams: Perplexity, smoothing: Laplace Smoothing, Good-Turing Discounting .

Unit-IV — 09 Hrs.

Syntax analysis:

Part-Of-Speech tagging (POS); Tag set for English (Upenn Treebank), Difficulties/Challenges in POS tagging , Rule-based, Stochastic and Transformation-based tagging, Generative Model: Hidden Markov Model (HMM Viterbi) for POS tagging; Issues in HMM POS tagging, Discriminative Model: Maximum Entropy model, Conditional random Field (CRF), Syntax tree vs Parse tree, Parsers: Top down and Bottom up parsers, CYK.

Unit-V — 08 Hrs.

Semantic Analysis:

Lexical Semantics: Corpus study: Study of Various language dictionaries like WordNet, Babe[illegible] Attachment for fragment of English. Relations among lexemes & their senses – Homonymy, Polysem[illegible]

Synonymy, Hyponymy, Semantic Ambiguity, Word Sense Disambiguation (WSD), Knowledge based approach (Lesk's Algorithm), Supervised (Naïve Bayes, Decision List).

Unit-VI 05 Hrs.

Pragmatic & Discourse Processing:

Discourse: Reference Resolution, Reference Phenomena, Syntactic & Semantic constraint on coherence, Anaphora Resolution using Hobbs and Centering Algorithm, Discourse segmentation, Coreference resolution .

Natural Language Processing Laboratory (RCP23ALPC502)

List of Laboratory Experiments

Suggested Experiments: (Any 08)

1. Preprocessing steps in NLP: Tokenization, stop word Removal, Lemmatization, stemming using NLTK and SPACY .
2. Implement Named Entity Recognition for any given text.
3. Perform morphological analysis and word generation for any given text.
4. Implement Chunking for the given input text.
5. Build a POS tagger using HMM.
6. Similarity Detection in NLP.
7. Implement N-Gram model for the given text input.
8. Generate word cloud using Python.
9. Any application of NLP: Spell Check, Autocorrect, plagiarism detection, sentiment analysis, sarcasm detection or text analytics in any domain.

Any other experiment based on syllabus may be included which would help the learner to understand topic/concept.

Oral examination will be based on the entire syllabus, including the practicals performed during laboratory sessions.

Text Books:

1. Raymond S. T. Lee, "Natural Language Processing: A Textbook with Python Implementation", 1st Edition, 2023.
2. Lewis Tunstall, Leandro von Werra, Thomas Wolf, "Natural Language Processing with Transformers", O'Reilly , 2022.

3. Thushan Ganegedara, Andrei Lopatenko, "Natural Language Processing with TensorFlow: The definitive NLP book to implement the most sought-after machine learning models and tasks", 2nd Edition, 2022.

4. Daniel Jurafsky, James H. and Martin, Speech and Language Processing, "An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition", Pearson, 2014

Reference Books:

1. Masato Hagiwara, "Real-World Natural Language Processing: Practical applications with deep learning", Mnaning, 2021.

2. Ashish Bansal, "Advanced Natural Language Processing with TensorFlow 2: Build effective real-world NLP applications using NER, RNNs, seq2seq models, Transformers, and more", Packt Publishing, 2021.

Web Links:

1. https://medium.com/data-science-in-your-pocket/pos-tagging-using-hidden-markov-models-hmm-viterbi-algorithm-in-nlp-mathematics-explained-d43ca89347c4
2. https://towardsdatascience.com/text-generation-using-n-gram-model-8d12d9802aa0
3. https://towardsdatascience.com/how-to-create-beautiful-word-clouds-in-python-cfcf85141214
4. https://medium.com/analytics-vidhya/best-nlp-algorithms-to-get-document-similarity-a5559244b23b
5. https://towardsdatascience.com/how-to-chunk-text-data-a-comparative-analysis-3858c4a0997a
6. https://towardsdatascience.com/how-to-chunk-text-data-a-comparative-analysis-3858c4a0997a
7. https://medium.com/@raghvendra.zarkar18/natural-language-processing-65f82c8dd7e0

Online Courses and Tutorials:

1. NPTEL Course : Natural Language Processing - Course (nptel.ac.in).
2. Coursera: Natural Language Processing Specialization [4 courses] (DeepLearning.AI) — Coursera
3. Udemy: NLP - Natural Language Processing with Python — Udemy

Program: Artificial Intelligence and Machine Learning	T. Y. B.Tech	Semester: V
Image Processing and Computer Vision (RCP23ACPC503)		
Image Processing and Computer Vision Laboratory (RCP23ALPC503)		

Prerequisite: Basic Mathematics, Python Programming

Course Objective(s):

1. To provide a fundamental understanding of image processing concepts, including image representation, enhancement techniques, and transformation methods
2. To equip students with the knowledge of image segmentation, morphological processing, feature detection, and motion analysis for practical applications in computer vision.
3. To introduce various image compression techniques and their significance in reducing storage and transmission costs while maintaining image quality

Course Outcomes:

On completion of the course, the learner will be able to:

CO	Course Outcomes	Blooms Level	Blooms Description
CO1	Apply image enhancement in spatial domain	L3	Apply
CO2	Analyse image in frequency domain through different transforms and enhancement in frequency domain	L4	Analyze
CO3	Apply different image segmentation, feature detection techniques on images	L3	Apply
CO4	Examine and apply different morphological operations on an image	L3	Apply
CO5	Apply geometric transformations on an image and Evaluate optical flow algorithms	L3	Apply
CO6	Implement and evaluate image compression techniques	L3	Apply

Image Processing and Computer Vision (RCP23ACPC503) Course Contents

Unit-I — 08 Hrs.

Image Processing Fundamentals and Image Enhancement:

types, Image formats, Sampling & Quantization.

Spatial Domain Techniques:

Point Processing; Digital Negative, Contrast Stretching, Thresholding, Bit Plane Slicing, Power Law Transformation, Dynamic Range Compression. Histogram Modelling; Histogram Stretching & Histogram Equalization. Neighbourhood Processing; Noise, Smoothing (Low Pass Averaging Filter, Low Pass Median Filter), Sharpening Filters (High Pass Filtering & High Boost Filtering). Mean filters, Order statistic filters

Unit-II — 08 Hrs.

Image Segmentation:

Connectivity of Pixels, Detection of discontinuities (Point, Line, Edge), Detection of Edges (Computing Gradients, 1^{st} order Derivative Filters, 2^{nd} order Derivative Filters, Laplacian of Gaussian). Region-based segmentation-Region Growing, Region Splitting, Region Merging, Region Split & Merge.

Unit-III — 07 Hrs.

Image Transforms & Morphological Processing:

Fourier Transform, 1D DFT, Frequency domain techniques - 2D-DFT, Low pass Filter (Ideal, Butterworth, Gaussian), High pass Filter (Ideal, Butterworth, Gaussian). Hadamard Transform, Walsh Transform.

Morphological Operations: Dilation, Erosion, Opening, Closing

Unit-IV — 05 Hrs.

Geometric Transformations: Translation, Rotation, Scaling, Shearing

Feature Detection & Description:

Interest or Corner Point Detectors- Harris and Hessian. Histogram of Oriented Gradients, Scale Invariant Feature Transform (SIFT), Speeded up Robust Features (SURF), Scale-Space Analysis- Image Pyramids

Unit-V — 06 Hrs.

Object Segmentation & Detection:

Canny Edge Detection, Difference of Gaussian (DOG), Graph Technique, Distance Metrics, G

Thresholding, Otsu's Thresholding, Watershed Algorithm

Motion Analysis & Optical Flow: Introduction to Motion Detection, Lucas-Kanade, Horn-Schunck

Unit-VI

05 Hrs.

Image Compression: Run-Length Encoding (RLE), Huffman Coding, LZW (Lempel-Ziv-Welch), Improved Grey Scale coding (IGS)

Image Processing and Computer Vision Laboratory (RCP23ALPC503)

List of Laboratory Experiments

Suggested Experiments: (Any 08)

1. Point Processing Techniques (Digital Negative, Thresholding, Intensity Transformation,Contrast Stretching).
2. Sharpening & Smoothing filters
3. Histogram Modelling
4. Frequency Domain Filtering.
5. Edge detection.
6. Morphological Operations
7. Object Detection
8. Image assessment with NumPy and OpenCV
9. Image Transformation in OpenCV
10. Feature Detection using OpenCV- Corner
11. Image Arithmetic Operations
12. Image Compression
13. Motion analysis and Action detection
14. Project Based Learning
15. Research Article Review

Any other experiment based on syllabus may be included which would help the learner to understand topic/concept.

Practical examination will be based on the entire syllabus, including the practicals performed during laboratory sessions.

Text Books:

1. Rafael C. Gonzalez, "Digital Image Processing", 4th Edition, 2018
2. Christopher M. Bishop, "Pattern Recognition and Machine Learning", 1st Edition, 2006.
3. DRichard Szeliski, "Computer Vision: Algorithms and Applications", 2nd Edition, 2022.

Reference Books:

1. Emanuele Trucco and Alessandro Verri, "Introductory Techniques for 3-D Computer Vision", 1st Edition, 1998.
2. William K. Pratt, "Digital Image Processing: PIKS Scientific Inside", 4th Edition, 2007
3. David Forsyth and Jean Ponce, "Computer Vision: A Modern Approach", 2nd Edition, 2011.
4. E. R. Davies, "Computer and Machine Vision: Theory, Algorithms, Practicalities", 4th Edition, 2012.
5. S. Jayaraman, S. Esakkirajan, and T. Veerakumar, "Digital Image Processing", 2nd Edition, 2019.
6. Adrian Kaehler and Gary Bradski, "Learning OpenCV: Computer Vision with the OpenCV Library", 1st Edition, 2008.

Web Links:

1. https://opencv.org/
2. https://staff.fnwi.uva.nl/r.vandenboomgaard/IPCV20162017/index.html
3. https://www.geeksforgeeks.org/computer-vision/
4. https://egyankosh.ac.in/handle/123456789/90205

Online Courses:

1. **Computer Vision and Image Processing – Fundamentals and Applications:**
   - https://onlinecourses.nptel.ac.in/noc22_ee48/preview/
   - https://www.coursera.org/learn/introduction-computer-vision-watson-opencv
2. Computer Vision:
   - https://onlinecourses.nptel.ac.in/noc19_cs58/preview

- https://www.coursera.org/specializations/computer-vision

3. **Digital Image Processing:**

   - https://onlinecourses.nptel.ac.in/noc20_ee75/preview

   - https://www.coursera.org/learn/introduction-image-processing

Program: Artificial Intelligence and Machine Learning	T.Y. B.Tech	Semester: V
Computer Network (RCP23ACMD504)		

Course Objective(s):

1. To Master the principles of computer networking across all layers, from physical to application, including modern technologies like 5G.
2. Develop proficiency in network design, protocols, and problem-solving techniques for various networking scenarios.

Course Outcomes:

On completion of the course, the learner will be able to:

CO	Course Outcomes	Blooms Level	Blooms Description
CO1	Understanding the Demonstrate the concepts of data communication at physical layer and compare ISO - OSI model and TCP/IP model.	L2	Understand
CO2	Understand the fundamental concepts of the Data Link Layer and analyze different MAC protocols.	L4	Analyze
CO3	Design IP addressing schemes and implement routing solutions.	L3	Explain
CO4	Analyze various transport layer, application layer protocols.	L4	Apply

Computer Network (RCP23ACMD504)
Course Contents

Unit-I Introduction to Networking — 06 Hrs.

Introduction to computer network, network application, network software and hardware components, Network topology, design issues for the layers. Reference models: Layer details of OSI, TCP/IP models.

Unit-II Physical Layer — 05 Hrs.

Introduction to Digital Communication System: Guided Transmission Media: Twisted pair, Coaxial, Fiber optics. Unguided media: Bluetooth. Data Encoding techniques.

Unit-III Data Link Layer — 08 Hrs.

DLL Design Issues (Services, Framing, Error Control, Flow Control), Error Detection and Correction (Hamming Code, Parity, CRC, Checksum), Elementary Data Link protocols: Stop and Wait, Sliding Window (Go Back N, Selective Repeat), HDLC Medium Access Control Sublayer: Channel Allocation problem, Multiple Access Protocol (Aloha, Carrier Sense Multiple Access (CSMA/CA, CSMA/CD).

Unit-IV Network Layer — 10 Hrs.

Network Layer: Network Layer design issues, Communication Primitives: Unicast, Multicast, Broadcast. IPv4 Addressing (Classfull and Classless), IPv4 Protocol, Network Address Translation (NAT), IPv6.

Routing algorithms: Link state routing, Distance Vector Routing.

Routing Protocols: ARP, RARP, ICMP, IGMP, RIP, OSPF.

Congestion control algorithms: Open loop congestion control, Closed loop congestion control. Token & Leaky bucket algorithms.

Unit-V Transport Layer — 06 Hrs.

The Transport Service, Port Addressing, Transport service primitives, Berkeley Sockets, Connection management (Handshake, Teardown), UDP, TCP, TCP Congestion Control: Slow Start.

Unit-VI Application Layer — 04 Hrs.

DNS: Name Space, Resource Record and Types of Name Server. HTTP, SMTP, Telnet, FTP, DHCP

Textbooks:

1. A.S. Tanenbaum, "Computer Networks", 6th Edition, Pearson Education, 2020.
2. B.A. Forouzan, "Data Communications and Networking with TCP/IP Protocol Suite",

Edition, TMH, 2022.

3. James F. Kurose, Keith W. Ross, "Computer Networking", A Top-Down Approach Featuring the Internet, 6^{th} Edition, Pearson, 2017.

4. David Hanes, Jerome Henry, Rob Barton, Gonzalo Salgueiro and Patrick Grossetete, "IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things", 1/e, 2018.

Reference Books:

1. Behrouz A. Forouzan, Firouz Mosharraf, "Computer Networks: A Top-Down Approach", Mc Graw Hill, 2023.

2. Dhanashree K. Toradmalle, "Computer Networks and Network Design", Wiley, 2020.

Web Links:

1. **Web Resources:**
   (a) https://www.netacad.com/courses/networking/networking-essentials
   (b) https://www.coursera.org/learn/computer-networking
   (c) https://www.edx.org/course/introduction-to-networking
2. **Online Courses: NPTEL / Swayam:**
   (a) https://nptel.ac.in/courses/106/105/106105081
   (b) https://nptel.ac.in/courses/106105183
   (c) https://onlinecourses.swayam2.ac.in/cec21_cs04/preview

Program: Artificial Intelligence and Machine Learning	T. Y. B.Tech	Semester: V
Programming Laboratory-III (Full Stack Development using NextJs) (RCP23ALMD505)		

Prerequisite: Basic knowledge of JavaScript, React.js, and fundamental database concepts

Course Objective(s):

1. Understand the fundamentals of Next.js and its advantages over traditional React applications.
2. Learn how to build full-stack applications using Next.js and integrate various databases.
3. Implement server-side rendering (SSR), static site generation (SSG), and API routes in Next.js.
4. Explore authentication, authorization, and state management techniques.
5. Deploy Next.js applications on cloud platforms like Vercel, Netlify, or AWS.
6. Optimize web applications for performance and SEO.
7. Work with relational (MySQL, PostgreSQL) and NoSQL (MongoDB, Firebase) databases for dynamic data handling.

Course Outcomes:

On completion of the course, the learner will be able to:

CO	Course Outcomes	Blooms Level	Blooms Description
CO1	Demonstrate an understanding of Next.js architecture and its role in modern web development.	L3	Apply
CO2	Develop full-stack applications integrating databases and API routes.	L6	Create
CO3	Implement authentication and authorization techniques for secure web applications.	L3	Apply
CO4	Apply relational (MySQL, PostgreSQL) and NoSQL (MongoDB, Firebase) databases for dynamic data handling	L3	Apply

Programming Laboratory-III (Full Stack Development using NextJs)(RCP23ALMD505) Course Contents

Unit-I — 03 Hrs.

Introduction to Next.js:

Install Next.js and configure a basic project. Explore the file structure of a Next.js project. Develop a multi-page application using the Next.js routing system. Implement dynamic and nested routes.

Unit-II — 05 Hrs.

Styling and UI Design:

Use CSS Modules for component-level styling. Integrate Tailwind CSS for responsive design. Create a global layout component for consistent UI across pages.

Unit-III — 05 Hrs.

Data Fetching and API Routes:

Implement getStaticProps, getServerSideProps, and getStaticPaths. Compare SSR, SSG, and CSR. Create API routes in the pages/api/ directory. Implement GET, POST, PUT, and DELETE requests.

Unit-IV — 05 Hrs.

Database Integration:

Set up a MySQL or PostgreSQL database using Prisma ORM. Perform CRUD operations. Connect to MongoDB using Mongoose. Perform CRUD operations on a products collection.

Unit-V — 05 Hrs.

Authentication and Authorization:

Implement Google and GitHub authentication using NextAuth.js. Manage user sessions securely. Implement different user roles (Admin, User, Guest). Restrict access to certain pages based on roles.

Unit-VI — 03 Hrs.

Performance Optimization and SEO and Deployment and Cloud Services:

Implement metadata using next/head. Optimize images using next/image. Deploy a Next.js project on Vercel and Netlify. Set up environment variables for security. Use Supabase or PlanetScale for database integration. Compare performance with traditional databases.

Suggested List of Experiments:

1. Develop a Multi-Page Application with Dynamic and Nested Routes in Next.js

2. Develop a Responsive Web Design with Tailwind CSS in Next.js

3. Build a Reusable Layout Component for Consistent UI

4. Implement API Routes in Next.js: Building CRUD Operations with GET, POST, PUT, and DELETE Requests

5. Set up a database, integrate Prisma ORM, and performing CRUD operations using Prisma.

6. Connect to MongoDB with Mongoose and Perform CRUD Operations on a Products Collection in Next.js

7. Implement Google and GitHub Authentication with NextAuth.js

8. Implement user roles (Admin, User, Guest), restricting access to pages based on roles, and managing secure user sessions.

9. Optimize the Performance and SEO in Next.js using next/head and next/image for Metadata and Image Optimization

10. Deploy a Next.js Application on Vercel and Netlify: Environment Variables and Database Integration with Supabase/PlanetScale

11. Mini Project

Minimum eight experiments and mini project from the above suggested list or any other experiment based on syllabus will be included, which would help the learner to apply the concept learnt. **Practical examination will be based on the entire syllabus, including the practicals performed during laboratory sessions.**

Text Books:

1. Gianni Ciolli, Boriss Mejías, Jimmy Angelakos, Vibhor Kumar, Simon Riggs, "PostgreSQL 16 Administration Cookbook", 2023.

2. Luca Ferrari, Enrico Pirozzi, "Learn PostgreSQL: Use, Manage, and Build Secure and Scalable Databases with PostgreSQL 16", Packt Publishing Ltd, 2023.

3. Roberto Rescigno, "Tailwind CSS: A Guide to Using the Popular Utility-first CSS framework", 2023.

4. Fritz Schneider and Thomas Powell, "JavaScript: The Complete Reference", 2012.

Reference Books:

1. Michele Riva, "Real-World Next.js: Build scalable, high-performance, and modern web applications using Next.js, the React framework for production", 2022.

2. Kartik Bhat, "Ultimate Tailwind CSS Handbook: Build sleek and modern websites with immersive UIs using Tailwind CSS", 2023.

3. Noel Rappin, "Modern CSS with Tailwind: Flexible Styling Without the Fuss", 2021.

Web Links:

1. https://nextjs.org/docs

2. https://www.coursera.org/learn/introduction-to-mongodb

3. https://www.coursera.org/learn/learn-tailwind-css?

4. https://www.coursera.org/learn/introduction-to-next-js

Program: Artificial Intelligence and Machine Learning	T. Y. B.Tech	Semester: V
Human Machine Interaction (RCP23ACPE511)		
Human Machine Interaction Laboratory (RCP23ALPE511)		

Course Objective(s):

1. This course provides an opportunity to learn and apply the design principles of Human Machine Interaction.
2. Learners will learn the basic human psychology of everyday actions and will be able to design an UI prototype of an application.
3. This course covers the discussion on various interaction design concepts.
4. The laboratory experiments are designed to practice the concepts and to adopt the systematic approach for interface design using various UX tools.

Course Outcomes:

On completion of the course, the learner will be able to:

CO	Course Outcomes	Blooms Level	Blooms Description
CO1	Understand Foundational Concepts of Human-Machine Interaction.	L2	Understand
CO2	Illustrate User-Centered Design Principles and Processes	L4	Analyze
CO3	Analyze and Design Effective User Interfaces.	L4	Analyze
CO3	Discover the various UX prototyping tools and its applications.	L6	Create

Human Machine Interaction (RCP23ACPE511) Course Contents

Unit-I — 06 Hrs.

Introduction:

The Human: History of User Interface Designing, I/O channels, Hardware, Software and Operating environments, The Psychopathology of everyday Things, Psychology of everyday actions, Reasoning and problem solving. The computer: Devices, Memory, processing and networks. Interaction: Models, frameworks, Ergonomics, styles, elements, interactivity, Paradigms.

Unit-II — 08 Hrs.

Design & Software Process:

Mistakes performed while designing a computer system, Human interaction with computers, importance of human characteristics human consideration, Human interaction speeds. Interactive Design basics, process, scenarios, navigation, Iteration and prototyping. HMI in software process: software life cycle, usability engineering, Prototyping in practice, design rationale. Design rules: principles, standards, guidelines, rules. Recognize the goals, Goal directed design process. Evaluation Techniques: Universal Design.

Unit-III — 07 Hrs.

Graphical User Interface and Web Interface:

The Graphical User Interface: Popularity of graphics, the concept of direct manipulation, characteristics of GUI, Web user Interface: Interface popularity, characteristics. The merging of graphical Business systems and the Web. Principles of user interface design.

Unit-IV — 07 Hrs.

Design Guidelines & Interaction Styles:

Perception, Gestalt principles, visual structure, reading is unnatural, color, vision, memory, six behavioral patterns, recognition and recall, learning, factors affecting learning, and time. Interaction Styles: Menus, Windows, Device-based and Screen- based Controls. Communication: Text messages, Feedback, and Guidance, Icons, Multimedia, and colors.

Unit-V — 06 Hrs.

Communication:

Text Messages, Feedback and Guidance, Icons, Multimedia, Color and Human Vision, Colors for Textual Graphics, Stastical Graphics Screen, and Web pages.

Case Study-Voice enabled Android Application for Vehicular Complaint System, Applying Human Computer Interaction to individual security using mobile application, Machine learning Applied to Human Learning, Educative Games for Autistic Children.

Unit-VI

05 Hrs.

UX tools:

Figma, Just In Mind, and any open-source tool for prototype designing like Penpot, MockFlow, Pencil Project. Mobile Ecosystem: Platforms, Application frameworks: Types of Mobile Applications: Widgets, Applications.

Human Machine Interaction Laboratory (RCP23ALPE511)

List of Laboratory Experiments

Suggested Experiments: (Any 08)

1. A literature survey on Human Machine Interaction (based on IEEE/Scopus-Indexed Publication)
2. To Study of open-source UX tools (Justinmind Prototype, Pidoco, Marvel ,Figma Prototype) and create a simple design for a given problem definition.
3. Know your client.
   - Design an app that can teach mathematics to children of 4-5 years age in schools in Rural Sector.
   - Design an app that can teach mathematics to children of 4-5 years age in schools in Urban Sector.
   - Design a site that can help people to sell their handmade products in metro cities.
   - Design a site that can connect housewives and keep them engaged. Note : Students should be able to do the following for any given problem statement
   - Analysis of user's/client's behavior eg their preferences, interests etc
   - What kind of interfaces will they like and why?
   - Existing apps - analyze and rate them.
   - What will be your choice of screen elements?
   - How will your app/web design be better than the existing one?
4. Goal-oriented design - Design an experience for passengers whose flight /train is delayed.
5. Design Principles - Understand principles of good UI design by heuristic evaluation. Design UI for a given problem statement.

6. Menus & Navigation - Redesign of a user interface (Suggest and implement changes in Existing User Interface) for a given problem statement.
   a. Windows & Screen controls - Design UI for a given problem statement.
   b. Design a navigator for a student new in your Institute.
   c. Design a navigator for a person new in tourist city/ village.
   d. Motor paralysis for differently able people.
   e. Vaccination App design with localization

7. Icons - Design appropriate icons pertaining to a given domain. (Eg. Greeting cards, Travelling, restaurants, Education, Medical, security at Airport, Malls etc)

8. Colors – Design a personal website for any socio-technical problem. Use color guidelines with statistical graphics for better visualization.

9. Design a Map-based UI(Web User) for the given problem statement. Example: Mumbai Dabbawallas with localization feature. Pet Care New Visitors to Hospital

10. Mini Project:

    SDG 3: Good Health and Well-being

    Design a mobile application that provides accessible mental health resources (e.g., guided meditations, stress management techniques, links to support organizations) with a focus on user-centered design for diverse users, including those with disabilities or limited digital literacy. Interactive System for Promoting Healthy Habits

    SDG 4: Quality Education

    Accessible and Engaging E-Learning Platform for Underserved Communities: Design an e-learning platform with a focus on accessibility for learners in underserved communities, considering factors like low bandwidth, limited device capabilities, and varying levels of digital literacy.

    SDG 11: Sustainable Cities and Communities

    Mobile Application for Promoting Sustainable Transportation Choices: Design a mobile app that encourages users to adopt sustainable transportation options like walking, cycling, and public transport by providing real-time information, route planning, gamified challenges, and rewards.

Any other experiment based on syllabus may be included which would help the learner to understand topic/concept.

Oral examination will be based on the entire syllabus, including the practicals performed during laboratory sessions.

Text Books:

1. Taylor Royce, "When Machines Understand Us: The Human Side of Emotional AI", 2025.

2. Kalbande, Kanade, Iyer, "Galitz's Human Machine Interaction", 1st Edition, Wiley Publications, 2015.

3. Donald A. Normann, "Design of everyday things", Basic Books, 2nd Edition, 2013.

Reference Books:

1. Rogers Sharp Preece, "Interaction Design: Beyond Human Computer Interaction", Wiley, 2019.

2. Guy A. Boy, "The Handbook of Human Machine Interaction", Ashgate publishing Ltd., 2017.

Web Links:

1. NPTEL Course: `https://onlinecourses.nptel.ac.in/noc25_cs38`

2. `https://www.coursera.org/learn/human-computer-interaction`

3. `https://www.edx.org/learn/human-computer-interaction`

4. Human-Computer Interaction (HCI) Course for AI Systems Design

Program: Artificial Intelligence and Machine Learning	T. Y. B.Tech	Semester: V
Advanced Data Structures and Algorithms (RCP23ACPE512)		
Advanced Data Structures and Algorithms Laboratory (RCP23ALPE512)		

Prerequisite: Python Programming, Data Structures, Design & Analysis of Algorithms

Course Objective(s):

1. To provide conceptual and practical knowledge of Advance Data Structures and Algorithms.
2. To Cultivate algorithmic thinking and problem-solving skills through practice and exposure to diverse problem domains.
3. Develop strategies for breaking down complex problems into manageable sub problems and applying suitable algorithms and data structures.
4. Develop skills to analyze problem complexity and choose appropriate data structures and algorithms for efficient solutions.

Course Outcomes:

On completion of the course, the learner will be able to:

CO	Course Outcomes	Blooms Level	Blooms Description
CO1	Understand the concept of time complexity and its importance in analyzing algorithms and to Explore the complexity analysis of popular machine learning algorithms..	L2, L4	Understand, Analyze
CO2	A Explore balanced search tree data structures and spatial data structures used in geometric and spatial applications.	L4	Analyze
CO3	Apply graph algorithms to solve real-world problems related to network flows, matching, and optimization.	L3	Apply
CO4	Understand the complexity classes NP, P, NP-complete, and NP-hard and their significance in algorithm classification and to explore the computational geometry algorithms.	L2, L4	Understand, Analyze

Advanced Data Structures and Algorithms (RCP23ACPE512) Course Contents

Unit-I — 08 Hrs.

Analysis of Algorithm Based on Time:

i. **Amortized Analysis:** Aggregate Method, Accounting Method, Potential Method (for Stack data structure)

ii. **Probabilistic and Randomized Algorithm:** Probabilistic approach to algorithm and Randomized Analysis, Indicator Random Variable (IRV), Analysis of Hiring Problem

Complexity Analysis of Machine Learning Algorithms:

i. Training Time Complexity and Testing Time Complexity

ii. Train/Test Complexity of Linear Regression

iii. Train/Test Complexity of Naïve Bayes Classifier

Unit-II — 10 Hrs.

Balanced Search Trees:

Red-Black Tree, Tango Tree,2-3 Tree, B Tree, B+ Tree, Splay Tree

Unit-III — 06 Hrs.

Advanced Data Structures:

i. **Spatial Data Structure:** KD Tree, R Tree

ii. **Probabilistic Data Structure:** Bloom filter, LogLog and HyperLogLog, Count Min sketch, MinHash with Machine Learning context (Vector Representation)

iii. **Functional Data Structures:** Binomial Tree, Binomial Heap

Unit-IV — 06 Hrs.

Graph Based Algorithms:

i. **Flow Network Introduction:** Residual Network, Augmenting Path, Ford-Fulkerson Method, Edmonds-Karp Method, Push-Relable Algorithm

ii. **Bipartite Matching:** Maximum Bipartite Matching

Unit-V — 07 Hrs.

Classification of Algorithms:

i. **Algorithm Classes:** P, NP, NP Hardness and NP Completeness

ii. **Np Completeness Proofs:** Satisfiability (3 sat), Reducibility, Cook's Theorem, Traveling Salesman Problem

iii. Approximation Algorithms: Vertex Cover Problem, Travelling Salesman problem

Unit-VI

02 Hrs.

Computational Algorithms – Computational Geometry:

Line Segment Properties, Convex Hull Graham's scan algorithm

Advanced Data Structures and Algorithms Laboratory (RCP23ALPE512)

List of Laboratory Experiments

Suggested Experiments: (Any 08)

1. Experiment on Amortized Analysis.
2. To perform and implement Hiring Problem.
3. Experiment on Randomized Algorithms (Randomized Quick Sort)
4. To implement Red Black Tree creation.
5. To implement Red Black Tree deletion.
6. To implement KD Tree.
7. To implement Ford Fulkerson Algorithm.
8. To implement Approximation Algorithms (Vertex Cover)
9. Experiment on Computational Geometry Algorithms (Graham Scan)

Minimum eight experiments from the above suggested list or any other experiment based on syllabus will be included, which would help the learner to apply the concept learnt.

Oral examination will be based on the entire syllabus, including the practicals performed during laboratory sessions.

Text Books:

1. Thomas H Cormen, Charles E. Leiserson, Ronald L Rivest, Clifford Stein, "Introduction to Algorithms", 3rd Edition, 2009.
2. S. Sridhar, "Design and analysis of algorithms", 1st Edition, Oxford, 2014.
3. Horowitz, Sahani and Rajsekaran, "Fundamentals of Computer Algorithms', 2nd Edition, Galgotia, 1998.
4. Harsh Bhasin, "Algorithms Design and Analysis", 1st Edition, Oxford, 2015.

5. Giuseppe Bonaccorso, "Machine Learning Algorithms", Packt, 2019.

Reference Books:

1. Rajeev Motwani, "Prabhakar Raghavan, Randomized Algorithm", Cambridge University, 2004.

2. Vijay V. Vajirani, "Approximation Algorithms", Springer, 2003.

3. "Computational Complexity", Stanford University, 2010.

4. Jason Brownlee, "Master Machine Learning Algorithms", Machine Learning Mastery, 2020.

Web Links:

1. Train/Test Complexity and Space Complexity of Linear Regression — by Writuparna Banerjee — Level Up Coding (gitconnected.com) https://levelup.gitconnected.com/train-test-complexity-and-space-complexity-of-linear-regression-26b604dcdfa3

2. Computational Complexity of ML Models — by Paritosh Kumar — Analytics Vidhya | Medium https://medium.com/analytics-vidhya/time-complexity-of-ml-models-4ec39fad2770

3. Importance of Understanding the Complexity of a Machine Learning Algorithm — by Baran Köseoğlu — Towards Data Science https://towardsdatascience.com/importance-of-understanding-the-complexity-of-a-machine-learning-algorithm-9d0532685982

4. Probabilistic Data Structures Decoded: Enhancing Performance in Modern Computing — by Naman Agrawal — Towards Data Science https://towardsdatascience.com/probabilistic-data-structures-decoded-enhancing-performance-in-modern-computing-17f700e6ea47/

5. ML Security Pro Tips: Understanding MinHash in a Security Context — by Melanie Beck — AI/ML at Symantec — Medium. https://medium.com/ai-ml-at-symantec/ai-ml-security-pro-tips-understanding-minhash-in-a-security-context-3dd0dd2ffe8

6. Bloom Filters and when to use them — by Janko Krstic — The Little Bit Ninja — Medium https://medium.com/the-little-bit-ninja/bloom-filters-and-when-to-use-them-ab64028996d4

7. The Power of Bloom Filters: A Comprehensive Guide — by Chiranjeet Baruah — Medium https://medium.com/@chiranjeetbaruah/the-power-of-bloom-filters-a-comprehensive-guide-e7d209774b5f

Program: Artificial Intelligence and Machine Learning	T. Y. B.Tech	Semester: V
Recommendation Systems (RCP23ACPE513)		
Recommendation Systems Laboratory (RCP23ALPE513)		

Prerequisite: Statistics for Data Science, and Machine Learning.

Course Objective(s):

This course aims to provide a comprehensive understanding of recommendation system techniques, including collaborative, content-based, knowledge-based, and hybrid models. It equips students with the skills to design, implement, and evaluate personalized recommender systems across various applications using real-world data.

Course Outcomes:

On completion of the course, the learner will be able to:

CO	Course Outcomes	Blooms Level	Blooms Description
CO1	Understanding the architecture and working of Collaborative Filtering, Content based recommendation systems.	L2	Understand
CO2	Analyze and differentiate various collaborative and content-based filtering techniques, including their algorithms, similarity functions, and limitations.	L4	Analyze
CO3	Apply knowledge-based and hybrid recommendation models to develop personalized recommendation solutions using constraint-based, case-based, or hybridization strategies.	L3	Apply
CO4	Evaluate the performance of different types of recommender systems using offline and online evaluation paradigms and appropriate metrics.	L5	Evaluate

Recommendation Systems (RCP23ACPE513)
Course Contents

Unit-I — 06 Hrs.

Introduction to Recommender System:

Introduction to Recommendation System, Framework of recommendation systems, Eliciting Ratings and other Feedback Contributions, Implicit and Explicit Ratings, Recommender system functions. Applications of recommendation systems, Issues with recommender system

Unit-II — 08 Hrs.

Collaborative filtering-based Recommender System:

Architecture of Collaborative Filtering, User-based nearest neighbour recommendation: Similarity Function, User-Based Algorithms, Item-based nearest neighbour recommendation: Similarity Function, Item-Based Algorithms, Comparing User-Based and Item-Based recommendations, data drift and concept drift.

Unit-III — 07 Hrs.

Content-based Recommender System:

Architecture of content-based systems, Content representation and content similarity, Item profiles, Discovering features of documents, Obtaining item features from tags, Representing item profiles, Methods for learning user profiles, Similarity based retrieval, The Role of User Generated Content in the Recommendation Process. Bayes classifier for recommendation, Regression based recommendation system. Advantages and drawbacks of content-based filtering.

Unit-IV — 06 Hrs.

Knowledge based recommendation:

Knowledge representation and reasoning, Constraint based recommenders, Case based recommenders, Persistent Personalization in Knowledge-Based Systems, Conversational Recommendation. Search based recommendation, Navigation-based recommendation.

Unit-V — 06 Hrs.

Hybrid Recommendation System:

Opportunities for hybridization, Monolithic hybridization design: Feature combination, Feature augmentation, Parallelized hybridization design: Weighted, Switching, Mixed, Pipelined hybridization design: Cascade Meta level, Limitations of hybridization strategies.

Unit-VI

06 Hrs.

Evaluating Recommendation System:

Evaluation Paradigms, General Goals of Evaluation Design, Design Issues in Offline Recommender Evaluation, Online Recommender evaluation techniques. Comparison between evaluation design of classification model and recommendation system, Error metrics, Decision-Support metrics, User Centered metrics. Comparative analysis between different types of recommendation systems.

Recommendation Systems Laboratory (RCP23ALPE513)

List of Laboratory Experiments

Suggested Experiments: (Any 08)

1. Build a Recommendation Engine with Item-Based Collaborative Filtering.
2. Build a Recommendation Engine with User-Based Collaborative Filtering.
3. Build Content-based recommendation engine on different datasets.
4. Build recommender system using association rule mining.
5. Implement Recommendation System using K-Nearest Neighbours.
6. Build Context-Aware Recommender Systems.
7. Build Constraint-based Recommenders.
8. Implement knowledge-based recommender system.
9. Implement a Monolithic hybridization design.
10. Evaluate the recommendation system with evaluation matrix.
11. Compare the performance of different recommender systems.
12. Mini Projects:

    1) Build a recommender system that suggests SDG-related content (articles, actions, or products) based on user preferences or content similarity.

    2) Build a recommender system that suggests eco-friendly products or services aligned with specific SDGs.

    3) Build a recommender system that suggests books aligned with specific SDGs, helping users find educational or informative literature related to sustainability, equality, or climate action.

Any other experiment based on syllabus may be included which would help the learner to understand topic/concept.

Oral examination will be based on the entire syllabus, including the practicals performed during laboratory sessions.

Text Books:

1. C.C. Aggarwal, "Recommender Systems: The Textbook", 1st Edition, Springer, 2016.
2. Jannach D., Zanker M. and FelFering A., "Recommender Systems: An Introduction", 1st Edition, Cambridge University Press, 2011.
3. Ricci, F., Rokach, L., & Shapira, B., "Introduction to Recommender Systems Handbook", Springer, Boston, MA 2011.

Reference Books:

1. P. Pavan Kumar, S. Vairachilai, Sirisha Potluri, "Recommender Systems: Algorithms and Applications", 1st Edition, CRC Press, 2021.
2. Kim Falk, "Practical Recommender Systems", 1st Edition, Manning, 2019.
3. Rounak Banik, "Hands-On Recommendation Systems with Python: Start building", Ingram short title, 2018.
4. M. D. Ekstrand, J.T. Riedl, J.A. Konstan, "Collaborative filtering recommender systems", 1st Edition, Now publishers, 2011.
5. P J. Leskovec, A. Rajaraman and J. Ullman, "Mining of massive datasets", 2nd Edition, Cambridge, 2012.
6. Rounak Banik, "Hands-On Recommendation Systems with Python: Start building powerful and personalized, recommendation engines with Python", 2018.

Web Links:

1. https://tryolabs.com/blog/introduction-to-recommender-systems
2. https://medium.com/@deepapandithu/recommender-system-user-collaborative-filtering-37613f0c6a9
3. https://www.analyticsvidhya.com/blog/2015/08/beginners-guide-learn-content-basedrecommender-systems/
4. https://www.researchgate.net/publication/2378325Knowledge-Based_Recommender_Systems
5. https://medium.com/analytics-vidhya/7-types-of-hybrid-recommendation-system3e4f78266ad8
6. https://www.researchgate.net/publication/226264572EvaluatingRecommendationSystems

Program: Artificial Intelligence and Machine Learning	T.Y. B.Tech	Semester:V
Environmental Science Tutorial (RCP23ITHSX06)		

Prerequisite: Interest in Environment and its impact on Human Calculus.

Course Objective(s):

1. Familiarise students with environment related issues such as depleting resources, pollution, ecological problems and the renewable energy scenario.
2. Give overview of Green Technology options.

Course Outcomes:

On completion of the course, the learner will be able to:

CO	Course Outcomes	Blooms Level	Blooms Description
CO1	Understand how human activities affect environment.	L2	Understand
CO2	Understand the various technology options that can make a difference.	L2	Understand

Environmental Science Tutorial (RCP23ITHSX06) Course Contents

Unit-I Air Pollution — 01 Hr.

Sources of Air pollution. Definition of Air Quality Index and how it is measured.

Unit-II Water Pollution — 01 Hr.

Sources of water pollution. Ground water pollution and eutrophication.

Unit-III Noise Pollution — 01 Hr.

Noise pollution and sources. Decibel limits for hospital, library, silence zone.

Unit-IV Biodiversity loss — 01 Hr.

Value of Biodiversity. Endangered species.

Unit-V Deforestation — 01 Hr.

Product and services provided by forests. Relationship between forests and climate change.

Unit-VI Renewable Energy sources — 01 Hr.

Our energy needs and global energy crisis. Renewable energy sources.

Unit-VII Climate change — 01 Hr.

Greenhouse gases and climate change.

Unit-VIII Green Technology — 01 Hr.

Data Center Energy Efficiency, Thin-Client and Energy Efficiency.

Textbooks:

1. R. Rajagopalan, "Environmental Studies from Crisis to Cure", 2nd Edition, Oxford University Press.
2. Erach Bharucha, "Textbook of Environmental Studies for Undergraduate Courses" for University Grants Commission, New Delhi & Bharti Vidyapeeth Institute of Environment Education and Research.
3. P. Narayanan, "Environmental Pollution: Principles, Analysis and Control", CBS Publis

4. Mohammad Dastbaz, Colin Pattinson, Babak Akhgar, Morgan and Kaufman, Elsevier, "Green Information Technology: A Sustainable Approach".

Reference Books:

1. Paulina Golinska, Marek Fortsch, Jorge Marx-Gómez, "Information Technologies in Environmental Engineering: New Trends and Challenges", Springer, 2011.

Websites:

1. CITES: www.cites.org
2. Convention on Biological Diversity: www.biodiv.org
3. Kalpvriksh: www.kalpvriksh.org
4. Water pollution: http://en.wikipedia.org/wiki/Water_polution
5. Ecosan: www.eco-solutions.org

List of Tutorials:

1. Case study on Smog.
2. Presentation on Water Pollution (Industrial, Sewage) explaining any specific case.
3. List effects of noise pollution on human health. Measure decibel level in college library, canteen, classroom.
4. Case study on effect of pollution on Biodiversity loss.
5. Debate for and against to promote Economic Growth Deforestation is required.
6. Presentation on different Renewable Energy Technologies.
7. Report on major impact of Global warming on Environment giving real examples.
8. Report on advantages and examples of Green Building for Sustainable development, Sustainable Software Design.

The final certification and acceptance of term work will be subject to satisfactory laboratory work performance and fulfilling minimum passing criteria in the term work.

Program: Artificial Intelligence and Machine Learning	T. Y. B.Tech	Semester: V
Semester Project-III (RCP23IPSC501)		

Course Objective(s):

Students are expected to design, simulate/implement a project based on the knowledge acquired from current semester subjects.

Course Outcomes:

On completion of the course, the learner will be able to:

CO	Course Outcomes	Blooms Level	Blooms Description
CO1	Conduct a survey of several available literatures in the preferred field of study.	L4	Analyze
CO2	Demonstrate various/alternate approaches to complete a project.	L2	Understand
CO3	Ensure a collaborative project environment by interacting and dividing project work among team members.	L3	Apply
CO4	Present their project work in the form of a technical report / paper and thereby improve the technical communication skill.	L3	Apply
CO5	Demonstrate the ability to work in teams and manage the conduct of the research study.	L2	Understand

Semester Project:

The purpose of semester project is to provide exposure to students with a variety of projects based on the knowledge acquired from the semester subjects. This activity is supposed to enrich their academic experience and bring enough maturity in student while selecting the project. Students should take this as an opportunity to develop skills in implementation, presentation and discussion of technical ideas/topics. Therefore, proper attention shall be paid to the content of semester project report which is being submitted in partial fulfillment of the requirements of the Third Year and it is imperative that a standard format be prescribed for the report.

Each student shall work on project approved by departmental committee approved by the Head of Department, a group of 03 to 05 students (max allowed: 5 students in extraordinary cases, subject to the approval of the department committee and the Head of the department) shall be allotted for each Semester Project. Each group shall submit at least 3 topics for the Semester Project. The departmental committee shall finalize one topic for every group. Semester Project Title or Theme should be based on knowledge acquired during semester. The project work shall involve sufficient work so that students get acquainted with different aspects of knowledge acquired from semester subjects.

Student is expected to:

- Select appropriate project title based on acquired knowledge from current semester subjects.
- Maintain Log Book of weekly work done(Log Book Format will be as per Table 1).
- Report weekly to the project guide along with log book.

Assessment Criteria:

- At the end of the semester, after confirmation by the project guide, each project group will submit project completion report in prescribed format for assessment to the departmental committee (including project guide).
- Assessment of the project (at the end of the semester) will be done by the departmental committee (including project guide).

Prescribed project report guidelines:

Size of report shall be of minimum 25 pages. Project Report should include appropriate content for:

- Introduction
- Literature Survey
- Related Theory
- Implementation details

- Project Outcomes
- Conclusion
- References

Assessment criteria for the departmental committee (including project guide) for Continuous Assessment:

Guide will monitor weekly progress and marks allocation will be as per Table 2.

Assessment criteria for the departmental committee (including project guide) for End Semester Exam:

Departmental committee (including project guide) will evaluate project as per Table 3.

Each group shall present/publish a paper based on the semester project in reputed/peer reviewed Conference/Journal/TechFest/Magazine before end of the semester.

Table 1: Log Book Format

Sr	Week (Start Date:End Date)	Work Done	Sign of Guide	Sign of Coordinator
1				
2				

Table 2: Continuous Assessment Table

Sr	Exam Seat No	Name of Student	Student Attendance	Log Book Maintenance	Literature Review	Depth of Understanding	Report	Total
			5	5	5	5	5	25

Table 3: Evaluation Table

Sr	Exam Seat No	Name of Student	Project Selection	Design/ Simulation/ Logic	Hardware/ Programming	Result Verification	Presentation	Total
			5	5	5	5	5	25

Program: Artificial Intelligence and Machine Learning	T. Y. B.Tech	Semester: V
Employability Skill Development Program-I (RCP23ALSC506)		

Prerequisite: Basic knowledge of C programming.

Course Objectives:

1. To implement and apply linear and non-linear data structures to solve real-world problems.
2. To design, code, and test data structures and algorithms with efficiency in mind.
3. To analyze time and space complexity through hands-on problem-solving.
4. To implement and compare searching and sorting techniques for different datasets.
5. To apply data structures in practical tasks or mini-projects for software solutions.

Course Outcomes:

On completion of the course, the learner will be able to:

CO	Course Outcomes	Blooms Level	Blooms Description
CO1	Analyze and compute time and space complexity for various algorithms	L3	Apply
CO2	Implement linear and non-linear data structures in a programming language.	L3	Apply
CO3	Solve real-world coding problems using suitable data structures.	L4	Analyze
CO4	Implement and compare searching and sorting algorithms on different datasets.	L3	Apply

Employability Skill Development Program-I (RCP23ALSC506) Course Contents

Unit-I Time Complexity — 03 Hrs.

- Introduction to time complexity
- Applying time complexity
- Log and square root time complexity
- Practice time complexity

Unit-II Linked Lists — 04 Hrs.

- Introduction to Arrays
- Arrays Practice
- Singly linked list
- Deletion in linked lists
- Circular and doubly linked list

Unit-III Stacks and Queues — 04 Hrs.

- How to implement a stack
- How to implement a queue
- Stack and Queue applications

Unit-IV Trees and Binary trees — 05 Hrs.

- Basic tree concepts
- Trees: Practice problems
- Binary trees
- BT: Practice problems
- Binary search trees
- BST: Practice problems

Unit-V Graphs — 05 Hrs.

- Basic concepts in Graphs

- Graph types
- Graph representation
- Graph Traversal
- Dijkstra's algorithm

Unit-VI Searching, Sorting and Hashing 05 Hrs.

- Searching
- Practice Searching
- Sorting
- Bubble sort
- Selection sort
- Insertion sort
- Merge sort
- Quick sort
- Radix sort
- Hashing

Suggested List of Experiments:

1. Implementation of Linked List operations.
2. Implementation of different operations on Linked List –copy, concatenate, split, reverse, count no. of nodes etc.
3. Implementation of polynomials operations (addition, subtraction) using Linked List.
4. Implementation of stack.
5. Implementation of Infix to Postfix conversion.
6. Implementation of Prefix and Postfix evaluation using stack.
7. Implementation of parenthesis checker using stack.
8. Implementation of Linear queue.
9. Implementation of Circular queue.
10. Implementation of Double ended queue.

11. Implementation of Priority queue program using array and Linked list.
12. Implementation of Binary Tree.
13. Implementation of Binary Tree Traversal.
14. Implementation of BST using following operations – create, delete, display.
15. Implementation of various operations on tree like – copying tree, mirroring a tree, counting the number of nodes in the tree, counting only leaf nodes in the tree.
16. Implementation of Graph traversal (DFS & BFS).
17. Implementations of Selection and Radix sort.
18. Implementation of Heap & Heap Sort.
19. Implementation of Merge Sort and Quick Sort.
20. Implementation of Bubble Sort.
21. Implementation of searching methods (Index Sequential, Fibonacci search, Binary Search)
22. Implementation of hashing functions with different collision resolution techniques.
23. Implementation of Insert and Delete operation in an array.
24. Implement the program to perform sum of array elements.
25. Implement the program to find maximum of array elements.
26. Implement the program to print all the leaders in the array.
27. Implement the program to find the sub array with the largest sum, and print its sum.
28. Implement the program to remove duplicates elements from Linked List.
29. Implement the program to Insert the node at the end in Circular Linked List.
30. Implement the program to delete the node from Circular Linked List.
31. Implement the program to insert the node in Double Linked List.
32. Implement the program to delete the node from Double Linked List.
33. Implement a stack using arrays to solve a classic problem: converting an integer from decimal to binary using the "divisor-remainder" method.
34. Implement a program to print numbers in a specific pattern using a queue.
35. Implement a program that simulates a to-do list manager using a queue.

36. Implement a program to find whether a given string is palindrome or not using stack.

37. Implement a program to find whether a given string is palindrome or not using queue.

38. Implementation of Adjacency Matrix for Tree.

39. Implementation of Adjacency List for Tree.

40. Implementation of Depth-First Search (DFS) tree traversal algorithm.

41. Implementation of Breadth-First Search (BFS) tree traversal algorithm.

42. Implement a program find the number of leaf nodes using DFS algorithm.

43. Implement a program to find the height of the tree for undirected connected tree.

44. Implement a program to print all the nodes at the level K from root node in sorted order given undirected connected tree.

45. Implement a program to check if there exists a path from root to any leaf whose sum is equal to targetSum for undirected connected tree.

46. Implement a program to count the number of neighbours of the node v for undirected connected tree.

47. Implement a program to find the subtree sum of each node i for undirected connected tree.

48. Implement a program to find the length of the diameter of the tree for undirected connected tree.

49. Implement a program to find the lowest common ancestor (LCA) for undirected connected tree.

50. Implement a program to find the distance between two nodes for undirected connected tree.

51. Implement a program to find the Kth ancestor of the node v for undirected connected tree.

52. Implement a program to find the largest node value at each level for undirected connected tree.

53. Implement a program to determine for each node the number of distinct colours in the subtree of the node for an undirected connected tree with N coloured nodes.

54. Implement a program to determine the minimum time in minutes you have to spend to collect all coins in the tree, starting at root node 1 and coming back to it for an undirected connected tree with N nodes and some of the nodes has gold coins in it.

55. Implement a program to find the height of it given connected binary tree with N nodes.

56. Implement a program to determine binary tree is height-balanced.

57. Implement a program to check given two undirected binary trees are the identical or not.

58. Implement a program to print the level order traversal of binary tree nodes.

59. Implement a program to print the zigzag level order traversal of binary tree nodes.

60. Implement a program to print the top view of the binary tree.

61. Implement a program to print binary tree minimum depth.

62. Implement a program to construct and return a binary tree using two integer arrays, preorder and inorder.

63. Implement a program to find the sum of leaves at deepest level in a binary tree.

64. Implement a program to check if the second tree T2 exists as a subtree in the first tree T1 in the given two undirected binary trees T1 and T2.

65. Implement a program to find the lowest common ancestor of the three nodes in a given binary tree with three nodes.

66. Implement a program to retrieve all root-to-leaf paths where the sum of the node values along the path equals the given integer target. Print each such path on separate lines for a binary tree and an integer target.

67. Implement a program to search if X exists in the BST or not.

68. Implement a program to print its nodes in sorted order in a binary search tree.

69. Implement a program to find the maximum/largest node in a binary search tree.

70. Implement a program to return the minimum difference between the values of any two different nodes in a binary search tree.

71. Implement a program to check if the given binary tree is a valid Binary Search Tree (BST).

72. Implement a program to find the Kth largest node in BST.

73. Implement a program to determine the inorder predecessor of a given node X in a binary search tree.

74. Implement a program to check if there exist two elements in the BST such that their sum is equal to S.

75. Implement a program to recover the tree without changing its structure for given a binary search tree, where exactly two nodes of the tree were swapped by mistake.

76. Implement a program to find the LCA (lowest common ancestor) of these two nodes using BST.

77. Implementation of Adjacency Matrix for Graph.

78. Implementation of Adjacency List for Graph.

79. A Cloud Provider's network spans across n locations with m connections between them. Your task is to determine whether Chef can send a message to Chefina. If possible, you need to find the minimum number of servers on such a route.

80. Given an undirected and unweighted graph and two nodes x and y, Implement a program to find the length of the shortest path between the two nodes. If no path exists, return -1.

81. Given a string s1, a character c1, and an integer k, Implement a program to find and print the position of the kth occurrence of the character c1 in the string s1. If the kth occurrence does not exist, print -1.

82. Implement a program to find the smallest and largest elements in an array of integers.

83. Implement a program to find the element in an array with the smallest absolute difference from a given integer k. If there are multiple elements with the same minimum difference, print the smallest of these elements.

84. Implement a program to find and print all pairs of integers from a list of n pairs where the sum of each pair is divisible by k.

85. Implement a program that reads an integer n followed by n pairs of integers. Given two additional integers left and right, the program should print all pairs whose sum and product fall within the inclusive range [left, right].

86. Implementation a program to find frequency of each element in the array.

87. Implementation a program to find frequency of elements using Hashing.

88. Implementation a program to Count Beautiful Pairs using Hashing.

89. Implementation a Hashing function using Division method.

Practical examination will be based on the entire syllabus including, the practicals performed during laboratory sessions.

Reference Books:

1. R. F. Gilberg and B. A. Forouzan, "Data Structures – A Pseudocode Approach with C", 2nd Edition, Cengage Learning, 2005.

2. Ellis Horowitz, Sartaj Sahni and Susan Anderson-Freed, "Fundamentals of Data Structures in C", 2nd Edition, W. H. Freeman, and Company 2008.

3. Mark A. Weiss, "Data Structures and Algorithm Analysis in C", 4th Edition, Pearson, 2014.

4. M. T. Goodritch, R. Tamassia, D. Mount, "Data Structures and Algorithms in C++", 2nd Edition, Wiley, 2011.

5. Kruse, Leung, Tondo, "Data Structures and Program Design in C", 2nd Edition, Pearson Education, 2013.

6. Tenenbaum, Langsam, Augenstein, "Data Structures using C", 2nd Edition Pearson, 2015.

7. Reema Thareja, "Data Structures using C", Oxford, 2017.

8. Seymour Lipschutz, "Data Structures, Schaum's Outline Series", 1st Edition, Tata McGraw-Hill, 2014.

Shirpur Education Society's

R. C. Patel Institute of Technology, Shirpur

(An Autonomous Institute)

Course Structure and Syllabus

Third Year B. Tech

Artificial Intelligence and Machine Learning

With effect from Year 2025-26

Shahada Road, Near Nimzari Naka, Shirpur, Maharashtra 425405
Ph: 02563 259 802, Web: www.rcpit.ac.in

Third Year B. Tech Artificial Intelligence and Machine Learning Semester-VI (w.e.f. 2025-26)

Sr	Course Category	Course Code	Course Title	Teaching Scheme			Evaluation Scheme					Total	Credit	
							Continuous Assessment (CA)							
							TA	Term Test 1 (TT1)	Term Test 2 (TT2)	Average of (TT1 & TT2)	ESE			
				L	T	P	[A]			[B]	[C]	[A+B+C]		
1	PC	RCP23ACPC601	Deep Learning	3			25	15	15	15	60	100	3	4
	PC	RCP23ALPC601	Deep Learning Laboratory			2	25				25	50	1	
2	PC	RCP23ACPC602	Large Language Models	3			25	15	15	15	60	100	3	4
	PC	RCP23ALPC602	Large Language Models Laboratory			2	25				25	50	1	
3	PC	RCP23ALPC603	Big Data Analytics Laboratory			2	25				50	75	1	1
4	PC	RCP23ALPC604	Advanced Java Laboratory			2	25				50	75	1	1
5@	PE	RCP23ACPE611	Advanced System Design	3			25	15	15	15	60	100	3	4
		RCP23ALPE611	Advanced System Design Laboratory			2	25				25	50	1	
		RCP23ACPE612	IoT Foundations	3			25	15	15	15	60	100	3	4
		RCP23ALPE612	IoT Foundations Laboratory			2	25				25	50	1	
		RCP23ACPE613	Cybersecurity and Ethical Hacking	3			25	15	15	15	60	100	3	4
		RCP23ALPE613	Cybersecurity and Ethical Hacking Laboratory			2	25				25	50	1	
6#	PE	RCP23ACPE621	Agentic and Explainable AI	3			25	15	15	15	60	100	3	4
		RCP23ALPE621	Agentic and Explainable AI Laboratory			2	25				25	50	1	
		RCP23ACPE622	Time Series & Financial Analysis	3			25	15	15	15	60	100	3	4
		RCP23ALPE622	Time Series & Financial Analysis Laboratory			2	25				25	50	1	
		RCP23ACPE623	Probabilistic Graph Models	3			25	15	15	15	60	100	3	4
		RCP23ACPE623	Probabilistic Graph Models Laboratory			2	25				25	50	1	
7	MD	RCP23ACMD601	DevOps and MLOps	2			25	15	15	15	60	100	2	3
	MD	RCP23ALMD601	DevOps and MLOps Laboratory			2	25				25	50	1	
9	EL	RCP23IPEL601	Project Stage-I			4	50				50	100	2	2
8	HS	RCP23ICHSX07	Constitution of India	1									A	A
			Total	15		18	350			75	575	1000	23	

@ Any1 Programme Elective Course from Set-1 # Any1 Programme Elective Course from Set-2

Prepared by:
Dr. N. G. Shinde

Checked by:
Dr. S. P. Patil

Prof. Dr. D. R. Patil
BOS Chairman

Prof. S. P. Shukla
C.O.E.

Prof. Dr. P. J. Deore
Dean Academics/Dy. Director

Prof. Dr. J. B. Patil
Director

R. C. Patel Institute of Technology, Shirpur
SES
Autonomous

Artificial Intelligence and Machine Learning
T. Y. B. Tech.
Semester - VI

Program: Artificial Intelligence and Machine Learning	T. Y. B.Tech.	Semester: VI
Deep Learning (RCP23ACPC601)		
Deep Learning Laboratory (RCP23ALPC601)		

Prerequisite: Natural Language Processing, Machine Learning.

Course Objectives:

1. To introduce fundamental concepts of artificial neural network and different learning algorithms: supervised and unsupervised neural networks.
2. Develop in-depth understanding of the key techniques in designing Deep Network and GAN.
3. To expose Deep Network based methods to solve real world complex problems.
4. To explore applications and challenges in deep learning.

Course Outcomes:

CO	Course Outcomes	Blooms Level	Blooms Description
CO1	Explain the fundamentals of deep neural networks and their training mechanisms.	L2	Understand
CO2	Apply optimization and regularization techniques to improve model performance.	L3	Apply
CO3	Demonstrate CNN models for supervised learning tasks.	L3	Apply
CO4	Develop solutions for sequence learning applications using recurrent networks.	L6	Create
CO5	Analyze unsupervised learning techniques for dimensionality reduction and data reconstruction.	L4	Analyze
CO6	Evaluate recent trends in adversarial networks and generative models.	L5	Evaluate

Deep Learning (RCP23ACPC601)
Course Contents

Unit-I 07 Hrs.

Supervised Learning Networks Feedforward DNN:

Perceptron: Representational power of Perceptron, The Perceptron Training Rule, Multilayer perceptron: Delta training rule; Multilayer Networks: A differentiable Threshold Unit (Sigmoid Neurons), Representational Power of Feedforward Networks; Activation functions: Tanh, Logistic, Linear, Softmax, ReLU, Leaky ReLU, Loss functions: Squared Error loss, Cross Entropy, Choosing output function and loss function

Unit-II 06 Hrs.

Optimization: Learning with backpropagation: EBPTA, Learning Parameters: Gradient Descent (GD), Stochastic and Mini Batch GD, Momentum Based GD, Nesterov Accelerated GD, AdaGrad, Adam, RMSProp, Convergence and local minima, stopping criteria.

Regularization: Regularization for Deep Learning: Parameter Norm Penalties, Dataset Augmentation, Noise Robustness, Early Stooping, Sparse Representation, Dropout.

Unit-III 06 Hrs.

Convolutional Neural Networks: Convolution operation, Padding, Stride, Relation between input, output and filter size, CNN architecture: Convolution layer, Pooling Layer, Weight Sharing in CNN, Fully Connected NN vs CNN, Variants of basic Convolution function, 2D Convolution.

ConvNet Architectures: LeNet: LeNet: Architecture, AlexNET: Architecture, ResNet : Architecture, ConvNeXt, EfficientNET.

Applications: Object detection and recognition tasks, medical image analysis, image classification

Unit-IV 07 Hrs.

Sequence Modelling: Sequence Learning Problem, Unfolding Computational graphs, Recurrent Neural Network, Bidirectional RNN, Backpropagation Through Time (BTT), Limitation of "vanilla RNN", Vanishing and Exploding Gradients, The Long Short-Term Memory, GRU, Deep recurrent Networks.

Applications: Sentiment analysis, stock prices or market trends

Unit-V 08 Hrs.

Unsupervised Learning Networks: Kohonen Self-Organizing Feature Maps – architecture, training algorithm

Autoencoders: Introduction, comparison with PCA, Linear Autoencoder, Undercomplete Autoen-

coder, Overcomplete Autoencoders, Regularization in Autoencoders, Denoising Autoencoders, Sparse Autoencoders, Contractive Autoencoders, Variational Autoencoders (VAEs)

Applications: image compression, feature extraction, risk assessment and fraud detection

Unit-VI 08 Hrs.

Adversarial Networks: Generative Vs Discriminative Modeling, Generative Adversarial Networks (GAN) Architecture, GAN challenges: Oscillation Loss, Mode Collapse, Uninformative Loss, Hyper-parameters, Tackling GAN challenges, Wasserstein GAN, Cycle GAN, Neural Style Transfer,

Diffusion Models: Introduction, Comparison with GANs.

Applications: image synthesis or style transfer, Data Augmentation.

Deep Learning Laboratory (RCP23ALPC601)

List of Laboratory Experiments (Any 10)

1. Implement Boolean gates using perceptron.
2. Implement representation power of perceptron.
3. Implement backpropagation algorithm from scratch.
4. Train CNN Models for Image Classification Tasks.
5. Evaluate the Effect of Optimizers SGD on Model Performance.
6. Evaluate the Effect of Optimizers Adam on Model Performance.
7. Compare the Performance of PCA and Autoencoders on Dimensionality Reduction Tasks.
8. Sequence Classification Using RNN or GRU (e.g., Sentiment Analysis or Activity Recognition).
9. Anomaly detection using Self-Organizing Network.
10. Compare the performance of PCA and Autoencoders on a given dataset.
11. Train Variational Autoencoders (VAEs) for Image Reconstruction.
12. Build Generative adversarial model for fake (news/image/audio/video) prediction.
13. Generate Synthetic Data Using Diffusion Models and Evaluate Results.
14. Mini Project

Oral examination will be based on the entire syllabus including, the practicals performed during laboratory sessions.

Text Books:

1. Asaton Zhang, Zhacary Lipton, Mu Li and Alex Smola, "Dive into Deep Learning", December 2023.
2. Simon Prince, "Understanding Deep Learning", MIT Press, December 2023.
3. Simon Haykin, "Neural Networks and Learning Machines", Pearson Prentice Hall, 3^{rd} Edition, 2010.
4. S. N. Sivanandam and S. N. Deepa, "Introduction to Soft Computing", Wiley India Publications, 3^{rd} Edition, 2018.
5. M. J. Kochenderfer, Tim A. Wheeler, "Algorithms for Optimization", IT Press.
6. David Foster, "Generative Deep Learning", O'Reilly Media, 2019.
7. Denis Rothman, "Hands-On Explainable AI (XAI) with python", Packt, 2020.

Reference Books:

1. Ian Goodfellow and Yoshua Bengio and Aaron Courville, "Deep Learning", An MIT Press, 2016.
2. François Chollet, "Deep Learning with Python", Manning Publication, 2017.
3. Josh Patterson, Adam Gibson, "Deep Learning: A Practitioner's Approach", O'Reilly Publication, 2017.
4. Andrew W. Trask, Grokking, "Deep Learning", Manning Publication, 2019.
5. John D. Kelleher, "Deep Learning", MIT Press Essential Knowledge series, 2019.
6. Douwe Osinga, "Deep Learning Cookbook", O'REILLY, SPD Publishers, Delhi.

Web Resources:

1. Deep Learning: https://vlab.spit.ac.in/ai/#/experiments
2. Deep learning book: https://www.deeplearningbook.org/
3. Deep learning all videos: https://www.cse.iitm.ac.in/ miteshk/CS6910.html
4. Deep Learning Specialization: https://www.coursera.org/specializations/deep-learning

Online Resources:

1. Deep Learning, IIT Ropar NPTEL course by Prof. Sudarshan Iyengar, Dr. Padmavati https://nptel.ac.in/courses/106106184

Program: Artificial Intelligence and Machine Learning	T. Y. B.Tech.	Semester: VI
Large Language Models (RCP23ACPC602)		
Large Language Models Laboratory (RCP23ALPC602)		

Prerequisite: Natural Language Processing, Deep Learning.

Course Objectives:

1. Introduce the fundamental concepts and applications of Generative AI.
2. Provide in-depth understanding of Transformer architecture, the core building block of most Large Language Models (LLMs).
3. Explore various LLM architectures and techniques like BERT, prompt engineering, and fine-tuning.
4. Introduce students to Multimodal LLMs that can process and understand different data modalities.

Course Outcomes:

CO	Course Outcomes	Blooms Level	Blooms Description
CO1	Describe the fundamental concepts and applications of Generative AI and to provide in-depth understanding of Transformer architecture, the core building block of Large Language Models (LLMs).	L2	Explain
CO2	Explain various LLM architectures and techniques like BERT, GPT-3, T5 and Large Reasoning Models.	L3	Apply
CO3	Apply prompt engineering techniques for effective LLM interaction and understand the concept of Retrieval Augmented Generation (RAG) and its role in LLMs.	L3	Apply
CO4	Summarize the different data modalities using Multimodal Architectures.	L5	Evaluate

Large Language Models (RCP23ACPC602)
Course Contents

Unit-I 07 Hrs.

Introduction to Generative AI & Transformer Architecture:

Domains of Generative AI, Text Generation, Image Generation, Music Generation, Video Generation. Limitations of RNN & LSTM, Tokenization, Transformer Architecture : encoders, decoders, attention mechanisms - types, Self-attention vs Flash Attention, feed-forward layer, RETRO Transformer, Reinforcement Learning with AI Feedback (RLAIF), Reinforcement Learning from Human Feedback (RLHF)

Unit-II 06 Hrs.

Language Models - Unveiling the Power of Words: BERT (Bidirectional Encoder Representations from Transformers) and its applications, exploring other notable LLM architectures (e.g.GPT-3, T5), Mixture of Experts (MoE), various benchmarks to evaluate LLMs – perplexity, BLEU score, Needle in a Haystack

Unit-III 06 Hrs.

Large Reasoning Models: Deep seek- v2: Multi head Latent Attention, Deep seek MoE, Knowledge Distillation, Mistral 7-B Architecture: sliding Window attention, Grouped Query Attention, Titans: Learning to memorize at test time, Knowledge Distillation, QWQ models.

Unit-IV 07 Hrs.

Prompt Engineering and Agentic AI: Introduction to prompt, examples of prompt, prompt engineering, prompt techniques, zero shot, one shot, few-shot learning, Agentic AI- a chain of thought , ReAct , self-consistency, Tree of thought, Multimodal CoT, Graph prompting, Large Action Models(LAMs), LLM based Agents, Auto Gen

Unit-V 08 Hrs.

Retrieval Augmentation & Generation (RAG) and Fine-tuning for LLMs: Understanding Retrieval and vector, vector storage: vector indexing and retrieval Algorithms: Annoy, HSNV, Inverted File System, LSH, vector quantization techniques: Scalar, Product, Binary, vector libraries, vector databases, Loading and retrieving in Lang Chain, Document loaders, Retrievers in Lang Chain. Fine-tuning: Quantization, PEFT, Full-Fine-tuning vs LoRA vs QLoRA, Fine-Tuning LLMs for different downstream tasks.

Unit-VI 08 Hrs.

Multimodal Architectures - Beyond Text: Introduction to Multimodal LLMs, Exploring architectures for Multimodal LLMs: Vision Transformer, Next GPT, GPT-4V, **Vision-Language Fusion Models:** BLIP-2, Flamingo, LLaVA, MiniGPT-4 **Temporal & Cross-Modal Models:** VideoLLaMA, ImageBind, Empowering Time Series Analysis with Large Language Models.

Large Language Models Laboratory (RCP23ALPC602)

List of Laboratory Experiments (Any 10)

1. Case study on Applications of Generative AI
2. Case study on role of Artificial Intelligence in achieving the Sustainable Development Goals
3. Fine Tuning Pre-trained Model On Custom Dataset (synthetic data) Using Transformer
4. Build your own LLM from scratch.
5. Query PDF using Lang Chain and Pine cone
6. Fine Tune Mistral7-B With Custom Dataset Using LoRA And QLoRA Techniques
7. Using in-built tools and creating custom tools for ReAct agent in Langchain.
8. Question Answering Application using LLM based agents.
9. Understanding various retrievers in Langchain.
10. Case study on comparison of Large Reasoning Models
11. Understanding multimodal models like Gemini vision
12. Build a simple multimodal generative model that combines text and image inputs to generate captions
13. Mini project

Oral examination will be based on the entire syllabus including, the practicals performed during laboratory sessions.

Text Books:

1. Ben Auffarth, "Generative AI with LangChain: Build large language model (LLM) apps with Python, ChatGPT, and other LLMs" by Packt Publishing, 2023.

2. Valentina Alto, “Modern Generative AI with ChatGPT and OpenAI Models”, by Packt Publishing, 2023.

3. Jay Alammar, Maarten Grootendorst, “Hands-On Large Language Models”, by O’Reilly, 2023

4. Thushan Ganegedara, “Natural Language Processing with TensorFlow”, by Packt Publishing, Second Edition, 2022.

Reference Books:

1. David Foster, “Generative Deep Learning”, O’Reilly, 2020.

2. Lewis Tunstall, Leandro von Werra & Thomas Wolf,” Natural Language Processing with Transformers”, 2022.

3. Sebastian Raschka, “Build a Large Language Model (From Scratch)”, ISBN 9781633437166.

Web Resources Blogs and Websites:

1. Mixture of Experts: Mixture of Experts Explained (huggingface.co)

2. PEFT:Efficient Model Fine-Tuning for LLMs: Understanding PEFT by Implementation — by Shivansh Kaushik — Medium

3. Various benchmarks to evaluate LLMs: LLM Benchmarks: Understanding Language Model Performance (humanloop.com)

4. Types of attention mechanism: Understanding and Coding the Self-Attention Mechanism of Large Language Models From Scratch (sebastianraschka.com)

5. Agents— RAG: Intro to LLM Agents with Langchain: When RAG is Not Enough — by Alex Honchar — Mar, 2024 — Towards Data Science

6. React— Agent: Teaching LLMs to Think and Act: ReAct Prompt Engineering — by Bryan McKenney — Medium

7. LLM based Agents : Superpower LLMs with Conversational Agents — Pinecone

8. RAGAS: Evaluating RAG pipelines with Ragas + LangSmith (langchain.dev)

9. Model distillation: LLM distillation demystified: a complete guide — Snorkel AI

10. Sentence classifier —BERT: Classify text with BERT — Text — TensorFlow

Program: Artificial Intelligence and Machine Learning	T. Y. B.Tech.	Semester: VI
Big Data Analytics Laboratory (RCP23ALPC603)		

Prerequisite: Java, Python, SQL.

Course Objectives:

To equip students with the necessary theoretical knowledge and practical expertise to design, implement, and manage scalable data solutions using both traditional and modern data technologies. Students will learn to leverage the Hadoop ecosystem (HDFS, MapReduce, Hive, Pig) for large-scale data processing, utilize specialized databases like PostgreSQL and MongoDB, and master the high-speed processing capabilities of Apache Spark for real-time analytics and machine learning applications, culminating in data visualization for business insights.

Course Outcomes:

CO	Course Outcomes	Blooms Level	Blooms Description
CO1	Analyze and apply the MapReduce paradigm using Hadoop and Spark for efficient distributed processing of large datasets.	L3	Apply
CO2	Differentiate and utilize structured (PostgreSQL) and various NoSQL databases (Document, Column-family, Graph) based on data model requirements.	L4,L5	Analyze, Evaluate
CO3	Design and implement Big Data ingestion, cleaning, and modeling pipelines for analytical purposes.	L3	Apply
CO4	Evaluate and select appropriate Big Data tools and cloud services for real-world application case studies.	L5	Evaluate

Big Data Analytics Laboratory (RCP23ALPC603) Course Contents

Unit-I 04 Hrs.

Hadoop & MapReduce Programming: Big Data Characteristics (4V's), Big Data vs. Traditional Databases. Hadoop: Architecture, HDFS (Design, File Operations, NameNode/DataNode Monitoring). Hadoop Ecosystem overview (YARN, Pig, Hive, HBase). **MapReduce Programming:** Job Run Anatomy, Mapper, Reducer, Combiner, Partitioner, Shuffle & Sort. Programming Examples: Word Count, Set Operations, Matrix Multiplication.

Unit-II 04 Hrs.

Relational & NoSQL Systems: PostgreSQL (Relational): pgAdmin - Schemas, Basic and Advanced Queries, Joins, Views. **NoSQL Introduction:** Types, CAP Theorem. **MongoDB:** Database and Collection creation. CRUD Operations - Document structure, Insert, Query, Update, and Delete documents, MongoDB Compass.

Unit-III 04 Hrs.

Apache Spark: Introduction to Spark: Architecture, Components (Spark Core, SQL, Streaming, MLlib, GraphX). Resilient Distributed Datasets (RDDs): Creation, Transformations (Narrow & Wide), Actions. Spark DataFrames and Spark SQL basics.

Unit-IV 04 Hrs.

Advanced NoSQL Databases: Cassandra (Column-Family): Architecture, Data Model (Keyspace, Column Family, Row, Column), CQL (Cassandra Query Language) basics, CRUD operations. **Neo4j (Graph):** Introduction to Graph Databases, Nodes, Relationships, Properties, Cypher Query Language fundamentals.

Unit-V 07 Hrs.

Data Ingestion & Processing Pipeline: ETL/ELT Concepts for Big Data. Introduction to Data Ingestion tools (e.g., Flume, Sqoop). Data Cleaning and Filtering techniques in Big Data context. Data modeling for Analytical processing..

Unit-VI 05 Hrs.

Visualization & Cloud Services: Introduction to Data Visualization for Big Data (e.g., Tableau, Power BI, or Python Libraries). Overview of Cloud Big Data Services (e.g., AWS EMR, Google Cloud Dataproc, Azure HDInsight). Case Studies on Big Data Applications.

Laboratory Experiments (Any 10)

1. Implement the following HDFS File Operations in Hadoop:

   - Adding files and directories
   - Retrieving files
   - Deleting file

   Use the Hadoop Web UI (50070) to check the NameNode status, view DataNode health, and observe file blocks and replication factors for a file uploaded.

2. i. Implement word count/frequency programs using MapReduce using a large text file (e.g., a novel) in Java or Python.

   ii. Implementing algorithm in MapReduce Matrix multiplication, Aggregates, joins, sort.

3. Install MongoDB/Compass. Practice CRUD operations. Create various types of indexes to optimize query performance.

4. Use the MongoDB Aggregation Framework to perform complex data analysis (e.g.grouping, counting, and calculating averages).

5. Install Cassandra. Design a keyspace, and create tables adhering to Cassandra's data denormalized model. Execute basic CQL commands.

6. Design a Cassandra schema suitable for time-series data and demonstrate insertion and querying based on time ranges.

7. Write a PySpark/Scala program to create RDDs and apply various Transformations (e.g., groupByKey, reduceByKey) and Actions.

8. Load data into a Spark DataFrame. Perform analytical queries and joining operations using Spark SQL.

9. Implement a Natural Join using Spark DataFrames or RDDs on two large related datasets.

10. Install Neo4j. Model a real-world scenario (e.g., a supply chain). Use Cypher for basic pattern matching.

11. Execute built-in graph algorithms in Neo4j (e.g., Shortest Path or PageRank) on your modeled data.

12. Experiment with Sqoop to transfer data between HDFS/Hive and a relational database (PostgreSQL/MySQL), or use Flume for log data collection.

13. Use a visualization tool (e.g., Tableau, Power BI, or Matplotlib/Seaborn in Python) to load and visualize analytical results obtained from Spark or MongoDB.

14. A major e-commerce platform needs to shift from legacy RDBMS to a Big Data architecture to handle billions of daily clickstream events. The objective is to design a hybrid analytics pipeline using Hadoop for storage, MapReduce/Spark for calculating key metrics (e.g., top products, page views), and a mix of PostgreSQL,MongoDB, Cassandra, and Neo4j to store diverse results. This system must ultimately identify complex user-to-user purchase relationships and generate personalized product recommendations efficiently.

15. Mini Project:

    SDG 13: Climate Change Impact Analysis

    - Analyse temperature and air quality data to identify trends and patterns.
    - Use social media data to assess public sentiment towards climate change.
    - Visualize the impact of climate change on specific regions.

    SDG 3: Healthcare Access Analysis

    - Analyze data on healthcare access, disease prevalence, and mortality rates.
    - Identify disparities in healthcare access across different populations.
    - Visualize the impact of healthcare interventions.

    SDG 4: Student Dropout Prediction

    - Analyze large datasets of student records, attendance, test scores, and socio-economic background.
    - Use a Spark ML classification model to predict which students are at high risk of dropping out.

    SDG 13: Greenhouse Gas (GHG) Emissions Tracking

    - Load historical industry, transport, and energy consumption data into Hive.
    - Use HiveQL to calculate and trend total GHG emissions per capita and per sector, visualized on a dashboard.

Oral examination will be based on the entire syllabus including, the practicals performed during laboratory sessions.

Text Books:

1. Mayank Bhushan "Big Data and Hadoop: Fundamentals, tools, and techniques for data-driven success", Edition 2nd ed. ISBN-13978-9355516664, 2024.
2. Jugnesh Kumar, Anubhav Kumar, Rinku Kumar "Big Data and Analytics: The key concepts and practical applications of big data analytics", ISBN-13978-9355516176, 2024.
3. Greyson Chesterfield "Mastering Apache Spark: Real-Time Big Data Analytics: Build Large-Scale Data Processing Pipelines with Apache Spark", 2024
4. Mrs A S R Sulthana & Mr Micheal Yeboah Frimpong "Demystifying the NoSQL", 2021.
5. Kristina Chodorow, Michael Dirolf, "MongoDB: The Definitive Guide Paperback", O'Reilly Publications, 2020.
6. Radha Shankarmani, M Vijayalakshmi, "Big Data Analytics", Wiley Publications, 2019.

Reference Books:

1. Simhadri Govindappa "Ultimate Big Data Analytics with Apache Hadoop: Master Big Data Analytics with Apache Hadoop Using Apache Spark Hive, and Python", 2024.
2. Dr Sudhakar Ranjan, Dr Anurag Bharatwal, Dr Amit Kumar Goel, "Hadoop in Action: Real World Applications and Best Practices for Big Data Processing Kindle" Edition,), 2024.
3. Ali Soofastaei, "Advanced Analytics in Mining Engineering: Leverage Advanced Analytics in Mining Industry to Make Better Business Decisions", 2023.
4. Herbert Jones, "Data Analytics: The Ultimate Guide to Big Data Analytics for Business, Data Mining Techniques, Data Collection, and Business Intelligence Concepts", 2020.

Online References:

1. NPTEL Course: https://onlinecourses.nptel.ac.in/noc20_cs92
2. MongoDB:https://www.mongodb.com/developer/products/mongodb/learn-mongodb-university-online-free-mooc/ msockid=07258624254b619719bf95cb24b36065
3. https://www.youtube.com/watch?v=S2MUhGA3lEw
4. https://www.youtube.com/watch?v=XhjIJGsAMsQlist=PL6UwySlcwEYKiC-EjEmN4f33c5fVpbzha

Program: Artificial Intelligence and Machine Learning	T. Y. B.Tech.	Semester: VI
Advanced Java Laboratory (RCP23ALPC604)		

Prerequisite: Object Oriented Programming using Java.

Course Objectives:

1. Build advanced Java programming and enterprise application development skills aligned with current industry practices.
2. Apply modern frameworks and APIs, especially Spring Boot, for secure and scalable software solutions.
3. Design and implement multi-tier applications with database integration, web services, and concurrency handling.
4. Gain practical exposure to DevOps tools and workflows for real-world software development and deployment.

Course Outcomes:

CO	Course Outcomes	Blooms Level	Blooms Description
CO1	Apply advanced Java programming concepts—including Generics, Streams, concurrency utilities, and modern Java (21+) features—to design efficient and parallel data processing applications.	L3	Apply
CO2	Develop and deploy enterprise-level applications using Spring Boot, incorporating RESTful APIs, dependency injection, and microservice-based architecture.	L6	Create
CO3	Design interactive and responsive web applications by integrating Spring MVC with frontend frameworks and ensuring effective database management using JDBC and ORM tools like Hibernate/JPA.	L6	Create
CO4	Apply software engineering best practices using design patterns, unit testing, version control, and DevOps tools to build secure, scalable, and high-performance Java solutions.	L3	Apply

Advanced Java Laboratory (RCP23ALPC604) Course Contents

Unit-I 06 Hrs.

Core Java Essentials & Data Handling: Java Collections Framework (ArrayList, HashMap, etc.); Generics with type parameters, wildcards, and erasure; Lambda Expressions and Functional Interfaces; Stream API (filtering, mapping, reducing, parallel streams); Introduction to Java 21+ features (records, sealed classes, pattern matching).

Unit-II 04 Hrs.

Database Integration & ORM: JDBC Architecture (drivers, connections, Statements, PreparedStatements, ResultSets); Transaction management, batch processing, connection pooling; Introduction to ORM with Hibernate/JPA (entities, mappings, queries); Spring Data JPA basics; Practical examples with MySQL/PostgreSQL, including CRUD operations and metadata handling.

Unit-III 06 Hrs.

Web Development Fundamentals: HTTP Protocol and web architecture; RESTful APIs design principles; JSON processing (parsing, serialization); Introduction to Spring Boot (auto-configuration, starters); Creating REST controllers, handling requests/responses, error management; Basic frontend integration (e.g., with Thymeleaf or API calls to React).

Unit-IV 06 Hrs.

Enterprise Frameworks & Architecture: Spring Framework core (IoC, Dependency Injection, Application Context); Spring MVC patterns; Advanced Spring Boot (microservices basics, actuators, profiles); Security fundamentals (Spring Security, JWT, OAuth); File handling, internationalization, and session management in web apps.

Unit-V 06 Hrs.

Concurrency & Advanced Concepts: Multithreading (threads, synchronization, executors); Java Concurrency utilities (Locks, Semaphores, Completable Future); Virtual threads (Project Loom); Reflection API (dynamic invocation, annotations); Performance optimization (garbage collection tuning, profiling). Design, Testing and DevOps.

Laboratory Experiments (Any 10)

1. Implement ArrayList, HashMap, and basic file I/O to create a console-based student record management system with search and sort functionality.
2. Create generic classes and methods to handle different data types, implement custom comparators, and use wildcards for flexible data processing.
3. Process CSV files using Stream API for filtering, mapping, and collecting data. Generate reports with statistical analysis using lambda expressions.
4. Build a console application using JDBC to manage books, members, and transactions with MySQL database. Implement CRUD operations with proper exception handling.
5. Develop RESTful web services using Spring Boot for employee management. Implement GET, POST, PUT, DELETE operations with JSON request/response handling.
6. Create a web application using Spring Boot and Thymeleaf to display products with search, filter, and pagination features. Include responsive web design.
7. Implement complete user registration, login, and session management system with password encryption and form validation using Spring Security basics.
8. Build a web-based file upload/download system with file type validation, storage management, and user access control. Include progress indicators.
9. Enterprise Report Generation using JasperReports and Spring Boot
10. Real-Time Event Processing using Apache Kafka
11. Use Java multithreading and Completable Future to simulate a task scheduler (e.g., email notifications), ensuring proper synchronization and error handling.
12. Implement Singleton and Factory patterns for a simple calculator app. Write JUnit tests to verify functionality and deploy using Maven to a local server.
13. Full-Stack Application with Spring Boot and React
14. Mini Project

Practical and Oral examination will be based on the entire syllabus including, the practicals performed during laboratory sessions.

Text Books:

1. Java: The Complete Reference (13th Edition): Herbert Schildt & Danny Coward, McGraw-Hill. Year: 2024 (Jan).
2. Core Java Volume I – Fundamentals: Cay S. Horstmann, Pearson / Oracle Press. For example, the 11th edition 2019.
3. Effective Java (3rd Edition): Joshua Bloch, Addison-Wesley Professional. Year: 2017.

Reference Books:

1. Mark Heckler, Spring Boot: Up and Running – Building Cloud-Native Java and Kotlin Applications, O'Reilly Media / Shroff Publishers (Indian edition), 2021.
2. Christian Ullenboom, Spring Boot 3 and Spring Framework 6, Rheinwerk / SAP Press, 2024.
3. K. Siva Prasad Reddy & Sai Upadhyayula, Beginning Spring Boot 3: Build Dynamic Cloud-Native Java Applications and Microservices (2nd Edition), Apress, 2023.
4. Marten Deinum, Spring Boot 3 Recipes: A Problem-Solution Approach for Java Microservices and Cloud-Native Applications (2nd Edition), Apress, 2024.

Web Links:

1. Getting started with Spring MVC + Thymeleaf: Tutorial Point – Spring Boot Thymeleaf
2. Spring Boot CRUD application with Thymeleaf: Baeldung – Spring Boot CRUD Thymeleaf
3. Official Thymeleaf + Spring tutorial: Thymeleaf.org – Thymeleaf + Spring

Program: Artificial Intelligence and Machine Learning	T. Y. B.Tech.	Semester: VI
Advanced System Design (RCP23ACPE611)		
Advanced System Design Laboratory (RCP23ALPE611)		

Prerequisite: Operating Systems, DBMS, Computer Networking & Protocols.

Course Objectives:

1. Design and justify high-level distributed architectures based on requirements.
2. Master low-level techniques for data distribution, caching, and performance optimization.
3. Design highly available and observable systems using SRE principles and asynchronous patterns.
4. Synthesize all knowledge to solve common industry design problems under interview pressure.

Course Outcomes:

CO	Course Outcomes	Blooms Level	Blooms Description
CO1	Accurately estimate scale and propose a resilient HLD (High-Level Design) with justified core components.	L2	Understand
CO2	Implement Consistent Hashing and justify caching policies (LRU/LFU) to optimize system latency and throughput.	L3	Apply
CO3	Design a decoupled system using Message Queues and implement fault-tolerance patterns (e.g., Circuit Breakers).	L3	Apply
CO4	Solve complex design problems (e.g., News Feed) by articulating trade-offs and low-level choices using the interview blueprint.	L3	Apply

Advanced System Design (RCP23ACPE611)
Course Contents

Unit-I 04 Hrs.

System Design Fundamentals & Core Concepts:

Introduction to System Design, Core Architectural Patterns, Back-of-the-Envelope Estimation, Availability & Scalability Metrics

Unit-II 06 Hrs.

Data Modeling and Storage Systems:

Relational Databases (SQL), NoSQL Databases, Data Partitioning and Sharding, Storage Technologies Filesystems, Parallel Database design & query evaluation.

Unit-III 08 Hrs.

Performance, Caching, and Load Balancing: Caching at Scale: Low-Level Concepts: Cache eviction policies (LRU, LFU), Write-through, Write-back, Cache-Aside patterns. Distributed Caching (e.g., Redis Cluster). Cache Invalidation, Load Balancing Techniques: Layer 4 vs. Layer 7, Algorithms: Round Robin, Least Connections, Weighted. Proxy servers (Forward/Reverse), Performance Optimization: Connection pooling, compression (Gzip), reducing I/O, and database query optimization. Throttling and rate limiting

Unit-IV 08 Hrs.

Asynchronous Processing and Communication: Message Queues & Brokers, Kafka vs. RabbitMQ concepts, Publish-Subscribe vs. Point-to-Point models., Data Ingestion and Stream Processing, Kafka Streams or Flink, Communication Protocols: REST vs. gRPC (Protocol Buffers),WebSockets for real-time applications, Polling, Long Polling, and Server-Sent Events (SSE).

Unit-V 10 Hrs.

Security, Monitoring, and Reliability: System Monitoring and Logging, Telemetry, Metrics (Prometheus), Logging (ELK/Graylog), and Alerting. Tracing (Jaeger/Zipkin) for microservices. SRE (Site Reliability Engineering) Basics, Security Fundamentals, Authentication (OAuth 2.0, JWT), Authorization, Transport Layer Security (TLS/SSL). Defenses against common attacks (DDoS, SQL Injection, XSS), Fault Tolerance and Disaster Recovery.

Unit-VI 06 Hrs.

Scaling Real-World Systems: Designing a URL Shortener (TinyURL), Distributed Shopping Cart Design, Designing a News Feed or Twitter Timeline, Designing a Distributed Key-Value Store.

Advanced System Design Laboratory (RCP23ALPE611)

List of Laboratory Experiments (Any 10)

1. **Back-of-the-Envelope Calculator:** Create a spreadsheet or simple program to quickly calculate storage (TB/year) and QPS for a given user base (e.g., 100M Daily Active Users).
2. **SQL vs. NoSQL Schema Modeling:** Design the schema for a simple E-commerce cart: one using a normalized SQL model and one using a de-normalized Document/NoSQL model.
3. **Consistent Hashing Simulation:** Implement a small simulation of Consistent Hashing (using a language of choice, e.g., Python) to observe key redistribution when 1 node is added and 1 node is removed from an initial 3-node cluster.
4. **Data Sharding Design:** Propose 3 different sharding keys (e.g., user_id, geolocation, date) for a highly-used logging service and analyze the pros/cons of each (e.g., data skew, query access pattern).Implement Boolean gates using perceptron.
5. **LRU Cache Implementation:** Implement a simple in-memory LRU (Least Recently Used) Cache using a standard library (like a linked hash map or equivalent structure).
6. **Cache-Aside vs. Write-Through Analysis:** Document the read/write steps and potential consistency issues for the Cache-Aside and Write-Through patterns for a user profile service.
7. **Load Balancer Algorithm Selection:** Given a scenario (e.g., homogenous servers, stateful sessions, high heterogeneity), recommend and justify a Load Balancing algorithm (Round Robin, Least Connections, Weighted).
8. **API Rate Limiter Logic:** Design the logic (using a Leaky Bucket or Token Bucket algorithm concept) to limit a user to 5 requests per second on a login API.
9. **Asynchronous Task Design:** Design the interaction flow between a Web App, a Message Queue (MQ), and a Worker Service to process a video upload and encoding task.
10. **Observability Stack Design:** Propose a 3-tool monitoring stack (Metrics, Logs, Tracing) for a microservices architecture and explain how it helps debug a P99 latency spike.
11. **Circuit Breaker Implementation:** Design a component (pseudocode) that wraps a remote service call and implements the Circuit Breaker pattern (Closed, Open, Half-Open states).
12. **JWT vs. Session Design:** Compare and contrast the Authentication workflow using traditional session cookies versus using JSON Web Tokens (JWT) for a mobile-first application.

13. **URL Key Generation Algorithm:** Implement the logic (Base-62 or a simple Hash with collision handling) for generating and ensuring uniqueness for a 7-character short key.

14. **Fan-out on Write vs. Read Cost Analysis:** Calculate the estimated write and read cost (in simple units) for a News Feed system with 1M users, comparing the Fan-out on Write vs. Fan-out on Read model assumptions.

15. **Distributed Key-Value Store API:** Design the API (interfaces and methods) for a simple distributed K-V store, including methods for replication and failure handling (e.g., put(key, value, quorum), get(key, quorum)).

Oral examination will be based on the entire syllabus including, the practicals performed during laboratory sessions.

Text Books:

1. 1. Martin Kleppmann, "Designing Data-Intensive Applications (DDIA)", O'Reilly 2017.

2. Alex Xu, "System Design Interview – An Insider's Guide (Volumes 1 & 2)", Independently published 2020/2021.

3. Mark Richards, Neal Ford, "Fundamentals of Software Architecture", O'Reilly 2020.

Reference Books:

1. George Coulouris, Jean Dollimore, Tim Kindberg, Gordon Blair, John Friday, "Distributed Systems: Concepts and Design", Addison-Wesley 2021.

2. Google (Various Authors), "Site Reliability Engineering (SRE)", O'Reilly 2016.

NPTEL Courses:

1. Cloud Computing and Distributed Systems Cloud Computing - Course

2. Distributed Systems Cloud Computing and Distributed Systems - Course

3. Introduction to Database Systems Introduction to Database Systems - Course

Program: Artificial Intelligence and Machine Learning	T. Y. B.Tech.	Semester: VI
IoT Foundations (RCP23ACPE612)		
IoT Foundations Laboratory (RCP23ALPE612)		

Prerequisite: Microcontrollers and Sensor Fundamentals.

Course Objectives:

1. To learn concepts of sensors and their interfacing.
2. To introduce different architectures used for connected smart devices.
3. To study integration of AI with IoT and various protocols used in the IoT environment.

CO	Course Outcomes	Blooms Level	Blooms Description
CO1	Explain the fundamentals and architecture of IoT and IIoT and interface sensors, actuators, and embedded devices for IoT applications.	L2	Understand
CO2	Implement IoT communication protocols and integrate data with cloud platforms.	L3	Apply
CO3	Apply IoT analytics and visualization techniques for smart solutions.	L3	Apply
CO4	Analyze industrial IoT frameworks, standards, and security mechanisms.	L4	Analyze

IoT Foundations (RCP23ALPE612)
Course Contents

Unit-I — 05 Hrs.

Introduction to IoT: Definition, evolution, and characteristics of IoT. Components: Things, Internet, Data, Cloud. IoT Ecosystem and Reference Architectures (3-layer, 5-layer). IoT Communication Models: Device-to-Device, Device-to-Cloud, Cloud-to-Cloud. Applications: Smart Home, Smart City, Healthcare, Agriculture, Energy.

Unit-II — 07 Hrs.

IoT Hardware and Software Components: Sensors, actuators, and embedded platforms (Arduino, Raspberry Pi, ESP32). Communication Modules: Wi-Fi, Bluetooth, ZigBee, LoRa, RFID, NFC. Edge devices and gateways. IoT Operating Systems: Contiki, RIOT, FreeRTOS..

Unit-III — 07 Hrs.

IoT Communication Protocols: IPv6, 6LoWPAN, MQTT, CoAP, HTTP/HTTPS, AMQP. Shared memory vs. message passing. Cloud Integration: AWS IoT, Azure IoT Hub, Google Cloud IoT. Case Studies: Smart Home, Smart Agriculture.

Unit-IV — 07 Hrs.

IoT Data Management and Analytics: IoT data lifecycle and characteristics. IoT cloud storage (ThingSpeak, Firebase). Edge and Fog Computing concepts. IoT Analytics – real-time dashboards, anomaly detection, data visualization. Case Study: Smart Energy Management.

Unit-V — 10 Hrs.

Industrial Internet of Things (IIoT): Introduction to IIoT and Industry 4.0. IIoT Architecture and Reference Models (IIRA, RAMI 4.0). Industrial Connectivity: OPC-UA, Modbus, PROFINET. M2M Communication. Use Cases: Manufacturing, Energy, Transportation, Robotics.

IoT Security, Standards and Governance: IoT Security challenges, Authentication, Authorization, and Encryption. Privacy and data protection mechanisms. IoT Standards and Frameworks: IEEE, IETF, ITU, ISO, NIST. Ethical and Governance aspects in IoT.

Unit-VI — 07 Hrs.

Emerging Trends and Case Studies: Digital Twins, Cyber-Physical Systems, AI and ML in IoT, 5G and IoT, Sustainable IoT, and Green Computing. Case Studies: Smart Manufacturing, Smart Grid, Connected Vehicles.

IoT Foundations Laboratory (RCP23ALPE612)

List of Laboratory Experiments (Any 10)

1. Introduction to Arduino/Raspberry Pi – setup and environment configuration.
2. Interfacing temperature and motion sensors with Arduino/Raspberry Pi.
3. IoT-based LED control using MQTT protocol (Node-RED or Mosquitto).
4. Real-time data acquisition and visualization using ThingSpeak
5. Smart Home automation using Wi-Fi and cloud dashboard
6. Implementation of MQTT and CoAP communication protocols.
7. Edge computing demo using ESP32 and local data processing.
8. Industrial data communication using Modbus protocol (simulation).
9. IoT security demonstration using SSL/TLS in MQTT communication.
10. Implement Cloud Integration and Data Logging using Firebase
11. Implement Smart Agriculture Monitoring using LoRa Communication
12. Deploy a simple machine learning model (e.g., vibration anomaly detection) on an ESP32 or Arduino Nano etc.
13. Digital Twin Simulation for a Smart Factory
14. Temperature and Humidity Monitoring with Edge Data Filtering
15. Mini-project: Design a prototype IoT/IIoT solution integrating sensors, cloud, and analytics.

Oral examination will be based on the entire syllabus including, the practicals performed during laboratory sessions.

Text Books:

1. Arshdeep Bahga & Vijay Madisetti, Internet of Things: A Hands-on Approach, Universities Press, ISBN 9788173719547.
2. Raj Kamal, Internet of Things: Architecture and Design Principles, McGraw Hill, ISBN 9789352605224.
3. Alasdair Gilchrist, Industry 4.0: The Industrial Internet of Things, Apress, ISBN 9781484220467.
4. Olivier Hersent et al., The Internet of Things: Key Applications and Protocols, Wiley, ISBN 9781119958352.

5. NPTEL Course: IoT and Applications by Prof. Sudip Misra, IIT Kharagpur.

Reference Books:

1. Pethuru Raj, Anupama C. Raman, The Internet of Things: Enabling Technologies, Platforms, and Use Cases, CRC Press.
2. Dirk Slama et al., Enterprise IoT: Strategies and Best Practices for Connected Products and Services, O'Reilly Media.
3. Jan Holler et al., From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence, Academic Press.
4. Aloknath De, Industrial Internet of Things (IIoT): Applications, Challenges and Standards, McGraw Hill.

Web Links:

1. Cisco Networking Academy – Introduction to IoT - https://www.netacad.com/courses/iot/introduction-iot
2. ThingSpeak IoT Analytics Platform (MathWorks) - https://thingspeak.com/
3. AWS IoT Core Developer Guide - https://docs.aws.amazon.com/iot/latest/developerguide/what-is-aws-iot.html
4. IBM: What is Industrial IoT (IIoT)? - https://www.ibm.com/topics/iiot
5. NIST Cybersecurity for IoT Program - https://www.nist.gov/programs-projects/cybersecurity-iot-program

Program: Artificial Intelligence and Machine Learning	T. Y. B.Tech.	Semester: VI
Cybersecurity and Ethical Hacking (RCP23ACPE613)		
Cybersecurity and Ethical Hacking Laboratory (RCP23ALPE613)		

Prerequisite: Computer Network.

Course Objectives:

1. To prepare students with the technical knowledge and skills needed to protect and defend cyber-attacks and understand ethical hacking methodology
2. To analyse and understand security attacks, web server vulnerabilities and prevention techniques to avoid exploitation.

Course Outcomes:

CO	Course Outcomes	Blooms Level	Blooms Description
CO1	Explain the breaching process of cyber threats.	L2	Understand
CO2	Apply the knowledge of different cryptographic algorithm.	L3	Apply
CO3	Identify different Cyber-attacks and apply Cyber Security mechanisms.	L3	Apply
CO4	Analyze and evaluate the cyber security needs of an organization.	L4	Analyze
CO5	Categorize a web-based security attack and demonstration of attack detection techniques.	L4	Analyze
CO6	Evaluate the performance and troubleshoot cyber security systems.	L5	Evaluate

Cybersecurity and Ethical Hacking (RCP23ACPE613) Course Contents

Unit-I 07 Hrs.

Introduction to Cyber Security:

Overview of Cyber Security, Internet, Governance – Challenges and Constraints, Cyber Threats, Vulnerabilities and its types Need for a Comprehensive Cyber Security Policy, Need for a Nodal Authority, Need for an International convention on Cyberspace.

Cyber Security Safeguards: Cyber security safeguards- overview, access control, audit, authentication, biometrics, cryptography, deception, denial of service filters, ethical hacking, firewalls, intrusion detection systems, response,scanning, security policy, threat management.

Unit-II 08 Hrs.

Cryptography: Introduction to Cryptography, stream and block cipher, Symmetric key (DES and AES) and Asymmetric key Cryptography (RSA), Key exchange protocol (DH), Hash algorithm(MD5 and SHA1, SHA256), Message Authentication code(MAC), Public key Infrastructure(PKI), Digital Signatures

Unit-III 07 Hrs.

Introduction to AI Security,Overview of AI Security: Need, challenges, and importance,Threat landscape in AI systems Multi-Agent Systems (MAS) and Security: Overview of Multi-Agent Systems (MAS): Agents, environment, communication Security challenges in MAS: Trust, deception, coordination, and privacy

Unit-IV 07 Hrs.

Overview of Ethical hacking Methodology: Ethical hacking process, Hacker's behaviour & mindset, Maintaining Anonymity, Hacking Methodology, Information Gathering, Active and Passive Sniffing, Physical security vulnerabilities and countermeasures. Internal and External testing.

Unit-V 06 Hrs.

Attacks and countermeasures: Social Engineering attacks and countermeasures. Password attacks, Privilege Escalation and Executing Applications, Network Infrastructure Vulnerabilities, IP spoofing, DNS spoofing, Wireless Hacking: Wireless footprint, Wireless scanning and enumeration, Gaining access (hacking 802.11), WEP, WPA, WPA2.

Unit-VI 07 Hrs.

Web server vulnerabilities and exploitation: DoS attacks, Web server and application vulnerabilities, SQL injection attacks, Vulnerability Analysis and Reverse Engineering, Buffer overflow attacks. Client-side browser exploits, Exploiting Windows Access Control Model for Local Elevation Privilege. Exploiting vulnerabilities in Mobile Application..

Cybersecurity and Ethical Hacking Laboratory (RCP23ALPE613)

List of Laboratory Experiments (Any 10)

1. Hands-on Cyber Lab Environment Setup with Cyber Ethics & Legal Response Analysis
2. Packet capture and protocol analysis: Analyzing Network Traffic through Packet Capture Techniques
3. Nmap scanning: Network Scanning and Enumeration Using Nmap
4. Passive reconnaissance & OSINT: Open-Source Intelligence (OSINT) Based Passive Information Gathering
5. Password cracking basics: Exploring Basic Password Cracking Methods and Tools
6. Web app mapping & Burp Suite basics: Web Application Structure Mapping Using Burp Suite Tools
7. SQL Injection – discovery and exploitation: Discovering and Exploiting SQL Injection in a Secure Practice Setup
8. Malware analysis (static): Static Malware Inspection and Feature Extraction
9. Wireless security (WEP/WPA2 testing): Testing and Analyzing Wireless Security Protocols (WEP/WPA2)
10. Log analysis & SIEM basics: Log Management and SIEM Analysis Techniques
11. Password Strength Assessment. Implement AI tools to evaluate and improve password policies
12. Automated Web Application Security Testing. Scan for vulnerabilities like SQL injection or XSS using AI-enhanced scanners
13. Generative Adversarial Networks (GANs) for Security. Use GANs both to create synthetic threats for testing defense systems and to detect fake content perpetrated by attackers.
14. Threat Modeling: Analysis and Evaluation of System Security through Threat Modeling

Oral examination will be based on the entire syllabus including, the practicals performed during laboratory sessions.

Text Books:

1. Cryptography and Network Security, Atul Kahate, Tata Mc Graw Hill.
2. Baloch, R., Ethical Hacking and Penetration Testing Guide, CRC Press, 2015.

Reference Books:

1. Sunit Belapure Nina Godbole, Cyber Security, Wiley India Pvt. Harish Chander, Cyber Laws
2. Davidoff, S. and Ham,J., Network Forensics Tracking Hackers through Cyberspace, Prentice Hall, 2012

Web Link:

1. Cybrary: Cybersecurity Fundamentals : https://www.cybrary.it/course/intro-to-it-and-cybersecurity/
2. Ethics & Law: https://www.eccouncil.org/ethics/

Online Courses: NPTEL / Swayam:

1. An advanced course focusing on cybersecurity concepts, technologies, and practices with a managerial perspective.: https://onlinecourses.nptel.ac.in/noc23cs127/preview?utm_source=chatgpt.com
2. Ethical Hacking: https://onlinecourses.nptel.ac.in/noc23cs44/preview?utm_source=chatgpt.com
3. Practical Cyber Security for Cyber Security Practitioners: https://onlinecourses.nptel.ac.in/noc25cs120/preview?utm_source=chatgpt.com

Program: Artificial Intelligence and Machine Learning	T. Y. B.Tech.	Semester: VI
Agentic and Explainable AI (RCP23ACPE621)		
Agentic and Explainable AI Laboratory (RCP23ALPE621)		

Prerequisite:

1. Basic Python programming and familiarity with libraries like numpy, pandas, and matplotlib.
2. Understanding of Machine Learning and Deep Learning fundamentals (e.g., model training, evaluation, and architecture concepts).
3. Introductory knowledge of AI concepts such as reasoning, environment, and search algorithms.
4. Familiarity with API integration and simple web frameworks (Flask or Streamlit) is recommended but not mandatory.

. Course Objectives:

1. Introduce the evolution of AI from traditional systems to agentic architectures with autonomy, reasoning, and goal-oriented behavior.
2. Explore Explainable AI (XAI) principles, techniques, and frameworks for building transparent and ethical AI systems.
3. Provide hands-on experience with OpenAI tools and APIs for developing intelligent, multimodal, and explainable agents.
4. Integrate concepts of agentic design and explainability to create trustworthy, real-world AI applications.

Course Outcomes:

CO	Course Outcomes	Blooms Level	Blooms Description
CO1	Explain the principles and architecture of agentic AI systems, including perception, reasoning, memory, and goal-setting.	L2	Understand
CO2	Apply Explainable AI (XAI) techniques such as LIME, SHAP, and Grad-CAM to interpret and evaluate AI model decisions.	L3	Apply
CO3	Demonstrate agentic systems and chat-based applications using OpenAI APIs and frameworks such as LangChain or AutoGen.	L3	Apply
CO4	Develop transparent, ethical, and accountable AI solutions by integrating explainability into agentic decision-making.	L3	Apply

Agentic and Explainable AI (RCP23ACPE621) Course Contents

Unit-I 07 Hrs.

Foundations of Agentic AI :

Evolution of AI: From Rule-based → Machine Learning → Generative → Agentic AI, What makes an AI agentic: autonomy, perception, reasoning, memory, goal-setting, planning, and reflection, Components of agent architecture (environment, sensors, actuators, policies), Categories: Reactive agents, Deliberative agents, Hybrid models, Frameworks and tools: LangChain, AutoGen, CrewAI, HuggingFace Agents, LlamaIndex Case examples: personal assistants, autonomous research bots, recommender agents.

Unit-II 07 Hrs.

Multi-Agent Systems and Coordination : Introduction to multi-agent environments, Communication and coordination: messaging, negotiation, task allocation, Emergent behavior, cooperation vs competition, Role-based and hierarchical agent design (planner, executor, verifier), Applications in distributed AI, simulations, and research agents, Frameworks: AutoGen multi agent, LangGraph, CrewAI orchestrations.

Unit-III 07 Hrs.

Explainable AI (XAI) Foundations: Motivation and need for explainability in AI, Key concepts: interpretability, transparency, accountability, Ethics and fairness in AI, Techniques: LIME, SHAP, Grad-CAM, Counterfactual explanations, Interpreting ML/DL models (decision trees, CNNs, transformers), XAI for Large Language Models (LLMs), Evaluation of explainability metrics.

Unit-IV 07 Hrs.

OpenAI Ecosystem and API Integration: Overview of OpenAI's tools and APIs: GPT (text), DALL·E (image), Whisper (audio), Embeddings, Codex, OpenAI API structure and authentication, Using API for NLP, vision, and multimodal tasks, Building chatbots, tutors, and assistants using GPT models, Prompt design and engineering, Fine-tuning overview and parameter-efficient methods, Integration with Python, Flask, or Streamlit.

Unit-V 07 Hrs.

Integrating XAI with Agentic AI : Explainable decision-making within agents, Designing agents that justify their actions, Generating and surfacing "reasoning traces" safely, Case study: AI Tutor that explains learning path, AI Recommender that clarifies suggestions, Combining XAI frameworks (SHAP, LIME) with agent reasoning, Frameworks and patterns for XAI + Agentic design

Unit-VI 07 Hrs.

Applications and Future Directions : Design and implementation of Explainable AI Agents using OpenAI tools, Example projects:AI Healthcare Assistant with Explainability, Financial Advisory Bot with Transparent Recommendations, Autonomous Research Assistant using OpenAI + XAI, Local Data Chatbot with Explainable Responses, Handling local/private data securely, Trends: self-reflective agents, goal memory, retrieval augmented reasoning, ethical AI, Open-source & enterprise deployment (LangServe, FastAPI)

Agentic and Explainable AI Laboratory (RCP23ALPE621)

List of Laboratory Experiments (Any 10)

1. Implement a simple reactive AI agent (rule-based)
2. Create a goal-based agent with memory and reasoning
3. Build a multi-agent chat (planner & executor) using LangChain
4. Implement communication between agents (task delegation)
5. Train and interpret a decision tree using LIME
6. Use SHAP to interpret a neural network
7. Visualize CNN activations using Grad-CAM
8. Query OpenAI GPT API for text summarization and explain results
9. Build a custom chatbot with memory using OpenAI + LangChain
10. Add explainability layer to chatbot (show reasoning path)
11. Create an explainable recommendation system (e.g., course or movie)
12. Design a local data QA chatbot with vector embeddings
13. Implement a healthcare advisor agent with explainable suggestions
14. Integrate Whisper (speech-to-text) + GPT (response) for an explainable voice agent
15. Mini project: Multi-agent explainable system (research assistant or advisor)

Oral examination will be based on the entire syllabus including, the practicals performed during laboratory sessions.

Text Books:

1. Anjanava Biswas, Wrick Talukdar, “Building Agentic AI Systems”, O’Reilly 2025.
2. Mayuri Mehta, Vasile Palade, Indranath Chatterjee, “Explainable AI (XAI): Foundations, Methodologies and Applications”, Springer 2023.
3. Michael Munn, David Pitman, “Explainable AI for Practitioners”, O’Reilly 2022.

Reference Books:

1. Michael Lanham, “AI Agents in Action”, Manning 2025.
2. Nicole Koenigstein, “AI Agents: The Definitive Guide”, O’Reilly 2025.

NPTEL Courses:

1. An Introduction to Artificial Intelligence https://onlinecourses.nptel.ac.in/noc22_cs56/preview
2. Artificial Intelligence: Concepts and Techniques https://onlinecourses.nptel.ac.in/noc25_cs159/preview
3. Fundamentals of Artificial Intelligence https://onlinecourses.nptel.ac.in/noc22_ge47/preview
4. Programming with Generative AI https://onlinecourses.nptel.ac.in/noc25_cs137/preview
5. Responsible & Safe AI Systems https://onlinecourses.nptel.ac.in/noc24_cs132/preview

Program: Artificial Intelligence and Machine Learning	T. Y. B.Tech.	Semester: VI
Time Series & Financial Analysis (RCP23ACPE622)		
Time Series & Financial Analysis Laboratory (RCP23ALPE622)		

Prerequisite: Basic Statistics and Probability, Knowledge of Regression Analysis, Fundamentals of Financial Concepts — returns, volatility, and risk measures.

Course Objectives:

1. To introduce fundamental concepts of time series data, its structure, components, and patterns with emphasis on financial and economic datasets.
2. To develop an understanding of methods for identifying trends, seasonality, and randomness in time series and their statistical properties.
3. To apply smoothing and regression-based techniques for forecasting and model building
4. To Explore models incorporating trend and volatility such as ARCH and GARCH, and understand their applications in financial data modelling.
5. To Integrate modern computational tools for data acquisition, pre-processing, and forecasting of financial time series, emphasizing real-world case studies.

Course Outcomes:

CO	Course Outcomes	Blooms Level	Blooms Description
CO1	Describe the fundamental components, types, and properties of time series data and explain concepts of stationarity and stochastic processes.	L2	Explain
CO2	Apply descriptive and smoothing techniques (moving averages, exponential smoothing, regression) for trend and seasonal analysis.	L3	Apply
CO3	Construct and estimate parameters of AR, MA, ARMA, and ARIMA models using statistical tools.	L4	Analyze
CO4	Evaluate models with trend and intervention effects; perform VAR and Monte Carlo simulation for multivariate time series.	L4	Analyze
CO5	Implement and interpret nonlinear time series models such as ARCH and GARCH for financial volatility modelling.	L4	Analyze
CO6	Acquire, pre-process, and forecast financial data using time series and machine learning based approaches.	L4	Analyze

Time Series & FinancialAnalysis (RCP23ACPE622) Course Contents

Unit-I 08 Hrs.

Introduction to Time series: Definition of a time series, Interpolation vs Extrapolation, Components of time series, Types of Time Series, Types of time series patterns, different types of data, simple descriptive techniques, Trends in time series (Parametric trends, differencing, non-parametric methods, noise), measurement of trends, seasonality, seasonal indices, stochastic processes. Stationary Time Series: the sample mean and its standard error, Stationary processes (weak and strict), statistical inference of time series.

Unit-II 08 Hrs.

Smoothening Methods: Regression analysis and forecasting- Least square estimation, Regression models for General Time Series Data, Autocorrelation detection using Durbin Watson Test, Estimating Parameters in Time Series. Exponential Smoothening Methods- 1st Order exponential smoothening, 2nd Order exponential smoothening, higher order exponential smoothening, Forecasting, exponential smoothening for seasonal data. .

Unit-III 08 Hrs.

Autoregressive Integrated Moving Average: Autocorrelation function (ACF) and Partial Autocorrelation function (PACF) plot, Linear time series and MA models, theoretical properties of time series with a MA (1) and MA (2) model, The AR model, simulating from an autoregressive process, The ARMA model, The ARIMA model Forecasting ARIMA processes, SARIMA model.

Unit-IV 06 Hrs.

Models with Trend: Removing trend, Unit Root (Augmented Dickey Fuller Test) and Regression Residuals, The Monte Carlo Method, Multi-equation Time Series Models: Intervention Analysis, Estimating the Intervention Effect, ADLs and Transfer Functions, Introduction to VAR Analysis. .

Unit-V 06 Hrs.

Non Linear Time series: The ARCH model: Feature of an ARCH, interpretation of ARCH model, The GARCH model: Existence of stationary solution of a GARCH(1,1) and Bilinear models. .

Unit-VI 06 Hrs.

Financial Data Analysis using Time Series: Obtaining and exploring Financial Data, preprocessing financial data for deep learning, adding quantities of Interest, scaling, building and training

forecasting models. Case studies on Financial forecasting using: Smoothening, Autoregressive models and Nonlinear time series.

Time Series & Financial Analysis Laboratory (RCP23ALPE622)

List of Laboratory Experiments (Any 10)

1. Detecting Trend in Time Series Data
2. Detecting Seasonality in Time Series Data
3. Time Series Decomposition
4. Exploratory Data Analysis of Time Series Data
5. Data Wrangling and Pre-processing for Time Series
6. Stationarity Testing and Making data stationary
7. Smoothening methods in Time Series
8. 2nd order and Higher Order Exponential Smoothening
9. Autoregressive Moving Average Models
10. ARIMA and SARIMA Models
11. VAR Models
12. ARCH and GARCH Models
13. Review of Research articles
14. Mini Project

Oral examination will be based on the entire syllabus including, the practicals performed during laboratory sessions.

Text Books:

1. Introduction to Time Series Analysis and Forecasting, Douglas C. Montgomery, Cheryl L. Jennings, Murat Kulahci, 2nd Edition, Wiley
2. Practical Time Series Analysis Prediction with Statistics & Machine Learning, Aileen Nielsen, O'reilly

Reference Books:

1. Time Series Analysis with Python Cookbook: Practical recipes for exploratory data analysis, data preparation, forecasting, and model evaluation, Tarek A. Atwan, Packt Publishing Limited, 2022

2. Hands-on Time Series Analysis with Python, B. V. Vishwas and Ashish Patel, APress 2020

3. Time Series Analysis and Its Applications : With R Examples, Robert H. Shumway and David S. Stoffer, 4th Edition, Springer

4. Time Series Databases: New Ways to Store and Access Data, Ted Dunning and Ellen Friedman, First Edition, O'Reilly, 2019

Web Links:

1. MIT Open courseware:
   https://ocw.mit.edu/courses/14-384-time-series-analysis-fall-2013/pages/lecture-notes/

2. Penn state Eberly College of Science:
   https://online.stat.psu.edu/stat510/

3. Forecasting: Principles and Practice (3rd ed):
   https://otexts.com/fpp3/

4. NIST Engineering Statistic Handbook:
   https://www.itl.nist.gov/div898/handbook/pmc/section4/pmc4.html

Program: Artificial Intelligence and Machine Learning	T. Y. B.Tech.	Semester: VI
Probabilistic Graph Models (RCP23ACPE623)		
Probabilistic Graph Models Laboratory (RCP23ALPE623)		

Prerequisite: Machine learning, Probability

Course Objectives:

1. To model problems using graphical models.
2. To design inference algorithms.
3. To learn the structure of the graphical model from data.

Course Outcomes:

CO	Course Outcomes	Blooms Level	Blooms Description
CO1	Explain the basic fundamentals of probabilistic graph theory.	L2	Understand
CO2	Illustrate various principles of graph theory and algorithms.	L3	Apply
CO3	Integrate core theoretical knowledge of graph theory to solve problems.	L4	Analyze

Probabilistic Graph Models (RCP23ACPE623)
Course Contents

Unit-I 06 Hrs.

Fundamentals: Overview and Motivation of Probabilistic Graphical Models, Structured Probabilistic Models, Marginal and Joint Distributions, Independence and Conditional Independence, Factors.
Representation of Bayesian networks: Semantics and Factorization, Reasoning Patterns, Flow of Probabilistic Influence, Conditional Independence, Independence in Bayesian Networks, Naïve Bayes, Applications.

Unit-II 07 Hrs.

Temporal Models of Bayesian Network: Overview of Temporal Models, Dynamic Bayesian Networks (DBN), Hidden Markov Model (HMM), Plate Models. Representation of Structured CPDs: Overview of Structured CPDs, Tree-Structured CPDs, Independence of Casual Influence, Continuous Variable, Applications.

Unit-III 06 Hrs.

Representation of Markov networks: Pairwise Markov Networks, General Gibbs Distribution, Conditional Random Fields, Independencies in Markov Networks, Imaps and perfect maps.

Unit-IV 07 Hrs.

Exact inference: Conditional Probability Queries, MAP Inference, Analysis of Complexity, Sum- and Max-product algorithms, Variable elimination, Belief propagation (message passing) on trees, Clique tree.

Unit-V 07 Hrs.

Inference and sampling methods : Simple Sampling, MCMC method, Gibbs sampling Algorithm, Importance sampling, Particle filtering.

Unit-VI 09 Hrs.

Learning Parameter Estimation: Learning Overview, Maximum Likelihood Estimation for Bayesian Networks, Bayesian Estimation, Bayesian Prediction, Bayesian Estimation for Bayesian Networks. Maximum Likelihood for Log-Linear Models, Maximum Likelihood for MRFs and CRFs. Structure Learning: Overview, Likelihood Scores, BIC and Asymptotic Consistency, Bayesian Scores, Learning Tree Structured Networks, Learning General Graphs: Heuristic Search.

Probabilistic Graph Models Laboratory (RCP23ACPE623)

List of Laboratory Experiments (Any 10)

1. Implement Discrete Bayesian Networks
2. Implementation of Alarm Bayesian Network
3. Implementation of Linear Gaussian Bayesian Networks (GBNs).
4. Implementation of Monty Hall Problem using Bayesian Network
5. Implementation of Exact inference in Bayesian Networks.
6. Implementation of Inference in Discrete Bayesian Network
7. Implementation of Causal Inference
8. Implement Approximate Inference using MCMC.
9. Implementation of Parameter Learning in Discrete Bayesian Networks.
10. Implementation of Dynamic Bayesian Networks (DBNs) for Temporal Modeling
11. Simulation of Interventional Experiments and Validation of Do-Calculus
12. Implementation of Nonlinear Conditional Dependencies in Continuous Bayesian Networks
13. Comparative Analysis of Exact and Approximate Inference Techniques
14. Mini project

Oral examination will be based on the entire syllabus including, the practicals performed during laboratory sessions.

Text Books:

(a) Koller, D. and Friedman, N. "Probabilistic Graphical Models: Principles and Techniques," MIT Press, 2009.

Reference Books:

(a) Jensen, F. V. and Nielsen, T. D. "Bayesian Networks and Decision Graphs. Information Science and Statistics," 2nd edition, Springer, 2002.

(b) Luis Enrique Sucar, "Probabilistic Graphical Models Principles and Application", Advances in Computer Vision and Pattern Recognition, Second Edition, Springer, 2021

(c) Kevin P. Murphy, "Machine Learning: A Probabilistic Perspective," 4th Printing. MIT Press, 2013.

(d) Barber, D. "Bayesian Reasoning and Machine Learning," 1st edition, Cambridge University Press, 2011.

(e) Bishop, C. M. "Pattern Recognition and Machine Learning (Information Science and Statistics)". 2nd printing, Springer, 2011.

(f) Wainwright, M. and Jordan, M. "Graphical Models, Exponential Families, and Variational Inference," Foundations and Trends in Machine Learning, 2008.

(g) Ankur Ankan, Abinash Panda Mastering Probabilistic Graphical Models with Python Packt Publishing

Web Resources Blogs and Websites:

(a) https://www.coursera.org/specializations/probabilistic-graphical-models?msockid=0fba0991cc8d64af281a1ae9cd3f657a

(b) https://createmomo.github.io/2019/01/07/Probabilistic-Graphical-Models-Revision-Notes/

(c) https://github.com/mcharrak/probabilistic-graphical-models-PGM-representation-coursera daphne-koller

Online Resources:

(a) Probabilistic Graphical Models : https://girishvarma.in/teaching/prob-graph-models/

(b) COL 776: Learning Probabilistic Graphical Models: https://www.cse.iitd.ac.in/p̃arags/teaching/2015/au15/col776/

Program: Artificial Intelligence and Machine Learning	T. Y. B.Tech.	Semester: VI
DevOps and MLOps (RCP23ACMD601)		
DevOps and MLOps Laboratory (RCP23ALMD601)		

Prerequisite: Machine Learning.

Course Objectives:

(a) To understand the need for DevOps as a software engineering practice.

(b) To know and understand the concept of Continuous Integration Continuous Delivery (CICD).

(c) To learn the concept of continuous deployment and monitoring strategies.

(d) To learn various tools used in DevOps.

(e) To comprehend the concepts in MLOps.

(f) To comprehend the concepts and deployment strategies in MLOps.

CO	Course Outcomes	Blooms Level	Blooms Description
CO1	Explain the fundamental concepts of DevOps.	L2	Understand
CO2	Describe the concept of continuous integration and continuous delivery.	L2	Understand
CO3	Compare various stages of continuous deployment and monitoring strategies.	L5	Evaluate
CO4	Explain various tools to implement concepts in DevOps.	L2	Understand
CO5	Describe the concepts used in the automation of Machine Learning life cycle phases.	L2	Understand
CO6	Analyze deployment strategies in MLOps.	L4	Analyze

DevOps and MLOps (RCP23ACMD601)
Course Contents

Unit-I 06 Hrs.

Introduction: Software Engineering- process framework, Software Development Life Cycle (SDLC) Process Models: Incremental and Evolutionary models.

Fundamentals of Agile Process: Need of Agile software development, Overview of Agile Development Models: Scrum, Extreme Programming, Feature Driven Development.

Devops: Introduction to Devops, History of Devops, Devops definition, DevOps Main Objectives, Continuous Integration & Deployment, Containers and Virtual Development, Configuration Management Tools.

Unit-II 06 Hrs.

Source Code Management: Version Control: GIT Features, 3-Tree Architecture, GIT – Clone /Commit / Push, GIT Hub Projects, GIT Hub Management, GIT Rebase & Merge, GIT Stash, Reset, Checkout, GIT Clone, Fetch, Pull, Membership GITHUB.

Continuous Integration and Continuous Delivery: Jenkins Architecture, Integrating Source code management, Continuous delivery to a staging environment or the pre-production, environment, Self-healing systems.

Unit-III 06 Hrs.

Continuous Deployment and Continuous Monitoring: Implementing a testing Strategy: Types of Tests, Integration testing, managing defect backlogs,

Continuous Deployment: Trade-offs in the deployment pipeline, Basic Deployment pipeline, Deployment pipeline practices & Commit stage, Automated Acceptance Test Gate,

Factors involved in monitoring systems: white-box and black-box monitoring, building a monitoring system, Site reliability engineering, SRE and DevOps, roles, and responsibilities of SRE, common tools used by SREs.

Unit-IV 06 Hrs.

Configuration Management: The Process of Configuration in Devops. Docker introduction, Docker Image, working with Docker Containers, Devops Monitoring Tool: Introduction to Nagios, Architecture. Virtualization and Containerization Difference between orchestration: Introduction to Kubernetes and automation.

Unit-V 04 Hrs.

Introduction to MLOps and ML Model Deployment: MLOps Motivation, MLOps (ML

Engineering + Operations), Machine Learning Life Cycle, MLOps Vs DevOps.

Data Management, Model Development and Training for MLOps: Model Development and Training for MLOps, Data versioning and reproducibility, Data preprocessing and feature engineering pipelines, Data validation and monitoring, Data quality assurance and governance, Model versioning and tracking. Creating and deploying ML/AI models, MLOps: Testing, Monitoring and Maintenance.

DevOps and MLOps Laboratory (RCP23ALMD601)

List of Laboratory Experiments (Any 10)

(a) Develop a web application and Deploy on the github. To Study DevOps: Principles, Practices, and DevOps Engineer Role and Responsibilities

(b) To configure Scrum and Kanban boards in JIRA, create sprints, and visualize issue progress.

(c) To carry out Version Control System / Source Code Management on the web application, install git and create a GitHub account.

(d) To Perform various GIT operations on local and Remote repositories using GIT Cheat-Sheet

(e) Continuous Integration, install and configure Jenkins with Maven/Ant/Gradle to setup a build Job on the above web application.

(f) To Build the pipeline of jobs using Maven / Gradle / Ant in Jenkins, create a pipeline script to Test and deploy an above application over the tomcat server.

(g) To Setup and Run Selenium Tests in Jenkins Using Maven on on the above web application.

(h) To study Docker Architecture and Container Life Cycle, install Docker and execute docker commands to manage images and interact with containers.

(i) To study Dockerfile instructions, build an image for a sample web application using Dockerfile.

(j) To install and Configure Pull based Software Configuration Management and provisioning tools using Puppet.

(k) To perform Software Configuration Management and provisioning using Puppet Blocks (Manifest, Modules, Classes, Function).

(l) Setting up a Version Control System (VCS) for ML Projects:

   i. Experiment with popular VCS tools like Git and create a repository for ML projects.

   ii. Learn to track code changes, collaborate with team members, and manage different branches.

(m) Creating a Continuous Integration (CI) Pipeline:

i. Build a CI pipeline using tools like Jenkins, Travis CI, or GitLab CI.

ii. Automate the process of building, testing, and validating ML models with each code commit.

(n) Containerization with Docker:

i. Containerize ML models and their dependencies using Docker.

ii. Experiment with Docker images, containers, and Dockerfile configurations

(o) Orchestrating ML Workflows with Kubernetes:

i. Deploy ML models as scalable and resilient services using Kubernetes.

ii. Experiment with deploying, managing, and scaling ML workloads in Kubernetes clusters.use tool like Microsoft Azur/AWS

(p) Experiment Tracking and Management:

i. Use tools like MLflow or Neptune.ai to track experiments, log metrics, and manage model versions.

ii. Explore features like hyperparameter tuning, model registry, and experiment reproducibility.

(q) Mini Project: Example like

- AI Model Monitoring and Explainability
- Post-Deployment Security and Maintenance.
- Model Metrics Logging with MLflow
- Detecting Data Drift Using EvidentlyAI
- Building a Grafana Dashboard for AI Model Monitoring
- Local Kubernetes deployment
- Deploying to AKS and CI/CD etc...

Oral examination will be based on the entire syllabus including, the practicals performed during laboratory sessions.

Text Books:

(a) Karl Matthias & Sean P. Kane, "Docker: Up and Running", O'Reilly Publication, 2^{nd} Edition, 2018.

(b) PierluigiRiti, "Pro DevOps with Google Cloud Platform", Apress, ISBN: 978-1-4842-3896-7.

(c) Gene Kim, Kevin Behr, George Spafford, "The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business", It Revolution Press publication, 2018.

(d) Gene Kim, Patrick Debois, John Willis, Jez Humble, "The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations", IT Revolution Press, 2021.

(e) Noah Gift, "Practical MLOps: A Guide to Building Real-World Machine Learning Systems", O'Reilly, 1st Edition, September 2021.

Reference Books:

(a) Viktor Farcic, "The DevOps 2.0 Toolkit: Automating the Continuous Deployment Pipeline with Containerized Microservices".

(b) Jennifer Davis and Katherine Daniels, "Effective DevOps: Building a Culture of Collaboration, Anity, and Tooling at Scale", O'Reilly Media, Inc., ISBN: 978-1-491-92630-7.

(c) Sanjeev Sharma and Bernie Coyne, "DevOps for Dummies", John Wiley & Sons, Inc., 2nd IBM Limited Edition, ISBN: 978-1-119-04705-6.

(d) Sridhar Alla, Suman Kalyan Adari, "Beginning MLOps with MLFlow: Deploy Models in AWS SageMaker, Google Cloud, and Microsoft Azure".

(e) Mark Treveil, Nicolas Omont, "Introducing MLOps: How to Scale Machine Learning in the Enterprise", O'Reilly Media, 1st Edition, January 5, 2021.

(f) Chris Fregly, Antje Barth, "Data Science on AWS: Implementing End-to-End Continuous Machine Learning Pipelines", O'Reilly, 1st Edition, 9 May 2021.

Web Links:

(a) https://www.redhat.com/en/resources/cloud-native-container-design-whitepaper

(b) https://www.redhat.com/en/topics/cloud-native-apps/what-is-serverless

(c) https://www.redhat.com/en/topics/automation/what-is-orchestration

(d) https://www.atlassian.com/continuous-delivery/continuous-integration

(e) https://www.flagship.io/glossary/site-reliability-engineer/

(f) https://docs.microsoft.com/en-us/learn/paths/intro-to-vc-git/

(g) https://www.javatpoint.com/kubernetes

(h) https://www.javatpoint.com/docker-tutorial

(i) https://www.javatpoint.com/jenkins

(j) 1https://www.javatpoint.com/jenkinss

(k) https://www.javatpoint.com/ansible

(l) https://www.javatpoint.com/selenium-tutorial

(m) https://prometheus.io/docs/introduction/overview/

(n) https://www.javatpoint.com/jira-tutorial

(o) https://www.geeksforgeeks.org/what-is-elastic-stack-and-elasticsearch/

(p) Coursera: Machine Learning Engineering for Production (MLOps) Specialization by Andrew Ng

(q) Coursera: "Machine Learning Engineering for Production (MLOps)" by deeplearning.ai. This course provides a comprehensive introduction to MLOps, covering topics like data and model versioning, deployment, monitoring, and more.

(r) Udacity: "Machine Learning Deployment" by Google Cloud.This course focuses on deploying and scaling machine learning models using Google Cloud technologies and covers MLOps principles.

Program: Artificial Intelligence and Machine Learning	T.Y. B.Tech.	Semester: VI
Project Stage-I (RCP23IPEL601)		

Course Objectives:

(a) To understand the basic concepts and principles of project development.

(b) To formulate/identify the problem statement.

(c) To implement the solution as per the problem statement.

(d) To develop the team building, writing, logical reasoning and management skills.

(e) To provide the connections between the designs and concepts across different disciplinary boundaries.

(f) To encourage students to become independent personnel, critical thinkers and lifelong learners.

Course Outcomes:

On completion of the course, the learner will be able to:

COs	Course Outcomes	Blooms Level	Blooms Description
CO1	Analyze the problem statement and produce solution of the problem considering cultural, social, environmental and economic factors using appropriate tool and method.	L4	Analyze
CO2	Interpret project based learning that allows students to transfer existing ideas into new applications.	L2	Understand
CO3	Apply the ability to work in teams and manage to conduct the project development activity.	L3	Apply
CO4	Use different perspectives from relevant disciplines which help them to get internships, jobs, and admission for higher studies.	L3	Apply
CO5	Explain the project development in the form of technical writing, and interpret what constitutes plagiarism and the use of proper referencing styles.	L2	Understand

Syllabus:

Domain knowledge (any beyond) needed from the following areas for the effective implementation of the project:

Database Management System, Networking and Internet of Things, Embedded Systems, Data science and Big data, Web and Application Development, Robotics, AI and Machine Learning, etc.

The above areas can be updated based on the technological innovations and development needed for specific project.

Guidelines: The main purpose of this activity is to improve the students' technical skills, communication skills by integrating writing, presentation and teamwork opportunities.

- Each student shall work on project stage-I as approved by the departmental committee, a group of 03 to 04 students (max allowed: 5 students in extraordinary cases, subject to the approval of the departmental committee) shall be allotted for each Project stage-I. The departmental committee shall include Head of Department, project coordinator and guide(s).
- Each group shall submit at least 3 topics for the Project stage-I. The departmental committee shall finalize one topic for every group.
- Each group is required to maintain separate log book for documenting various activities of the project stage-I (Refer Table 1).
- Each group will be reviewed twice in a semester and marks will be allotted based on the various points mentioned in the evaluation scheme.
- In the first review of this semester, each group is expected to complete 30% of project stage-I.
- In the second review of this semester, each group is expected to complete 50% of project stage-I.
- Interaction with alumni mentor will also be appreciated for the improvement of project stage-I.

Assessment Criteria:

- At the end of the semester, after confirmation by the project guide, each project group will submit project completion report in prescribed format for assessment to the departmental committee.
- Assessment of the project (at the end of the semester) will be done by the departmental committee.
- Oral examination shall be conducted by Internal and External examiners. Students have to give presentation and demonstration based on their project stage-I.

Prescribed project report guidelines:

Size of report shall be of minimum 30 pages (excluding cover and front pages). Project stage-I report should include appropriate content for:

- **Abstract**
- **Introduction**
  - Background
  - Motivation
  - Problem Statement
  - Objectives
  - Scope
- **Literature Survey**
  - Review of Existing System(s)
  - Limitations of Existing System(s)
- **Proposed System**
  - Analysis/Framework/ Algorithm
  - Details of H/W and S/W required
  - Design details
  - Methodology (your approach to solve problem)
- **Implementation Plan for Project Stage-II**
- **Conclusion**
- **References**

Assessment criteria for the departmental committee for Continuous Assessment:

Guide will monitor weekly progress and marks allocation will be as per Table 2.

Assessment criteria for the departmental committee for End Semester Exam:

Departmental committee will evaluate project as per Table 4.

Table 1: Log Book Format

Sr	**Week (Start Date:End Date)**	**Work Done**	**Sign of Guide**	**Sign of Coordinator**
1				
2				

Table 2: Continuous Assessment Sheet

Sr	**P.R.N.**	**Name of Student**	**Student Attendance (10)**	**Log Book Maintenance (10)**	**Literature Review (10)**	**Depth of Understanding (10)**	**Report (10)**	**Total (50)**
			10	10	10	10	10	50

Table 3: Evaluation Sheet

Sr	**P.R.N.**	**Name of Student**	**Project Stage - I Selection (10)**	**Design /Methodology /Logic (10)**	**Fabrication /Modelling /Simulation (10)**	**Result Verification (10)**	**Presentation (10)**	**Total (50)**
			10	10	10	10	10	50

Program: Artificial Intelligence and Machine Learning	T. Y. B.Tech.	Semester: VI
Constitution of India (RCP23ITHSX06)		

Prerequisite: NA

Course Objectives:

(a) To provide basic information about Indian constitution.

(b) To identify individual role and ethical responsibility towards society.

(c) To understand human rights and its implications.

Course Outcomes:

CO	Course Outcomes	Blooms Level	Blooms Description
CO1	Illustrate knowledge and legal literacy to take up competitive examinations.	L3	Apply
CO2	Explain state and central policies, fundamental duties.	L2	Understand
CO3	Describe Electoral Process, special provisions.	L2	Understand
CO4	Discuss powers and functions of Municipalities, Panchayats and Co-operative Societies.	L2	Understand
CO5	Associate Engineering ethics and responsibilities of Engineers.	L2	Understand
CO6	Interpret Engineering Integrity & Reliability.	L2	Understand

Constitution of India (RCP23ITHSX06)
Course Contents

Unit-I 02 Hrs.

Introduction to the Constitution of India:

The Making of the Constitution and Salient features of the Constitution. Preamble to the Indian Constitution. Fundamental Rights & its limitations.

Unit-II 03 Hrs.

Directive Principles of State Policy: Relevance of Directive Principles, State Policy, Fundamental Duties. Union Executives President, Prime Minister, Parliament, Supreme Court of India.

Unit-III 03 Hrs.

State Executives: Governor, Chief Minister, State Legislature, High Court of State. Electoral Process in India, Amendment Procedures, 42nd, 44th, 74th, 76th, 86th & 91st Amendments.

Unit-IV 02 Hrs.

Special Provisions: For SC & ST, Special Provision for Women, Children & Backward Classes, Emergency Provisions.

Unit-V 02 Hrs.

Human Rights: Meaning and Definitions, Legislation Specific Themes in Human Rights, Working of National Human Rights Commission in India, Powers and functions of Municipalities, Panchayats and Co- Operative Societies.

Unit-VI 02 Hrs.

Scope & Aims of Engineering Ethics: Responsibility of Engineers and Impediments to Responsibility. Risks, Safety and liability of Engineers. Honesty, Integrity & Reliability in Engineering.

Text Books:

(a) Durga Das Basu, "Introduction to the Constitution on India", (Students Edition) Prentice Hall EEE, 19th / 20th Edition, 2001.

(b) Charles E. Haries, Michael S. Pritchard and Michael J. Robins, "Engineering Ethics", Thompson Asia, 2003.

Reference Books:

(a) M. V. Pylee, "An Introduction to Constitution of India", Vikas Publishing, 3rd Edition, 2003.

(b) M. Govindarajan, S. Natarajan, V. S. Senthilkumar, "Engineering Ethics", Prentice Hall of India Pvt. Ltd. New Delhi, 2013.

(c) Brij Kishore Sharma, "Introduction to the Constitution of India", PHI Learning Pvt. Ltd., New Delhi, 7th Edition 2015.

(d) Latest Publications of Indian Institute of Human Rights, New Delhi.

Online Resources:

(a) www.nptel.ac.in

(b) www.hnlu.ac.in

(c) www.nspe.org

(d) www.preservearticles.com