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