COURSE STRUCTURE AND SYLLABI

M. Tech and M. Tech by

Research in CSE (2025-26)

Centurion

UNIVERSITY

Shaping Lives...
Empowering Communities...

SCHOOL OF ENGINEERING AND TECHNOLOGY
CENTURION UNIVERSITY OF TECHNOLOGY & MANAGEMENT
Andhra Pradesh 535003, India

Website: - www.cutmap.ac.in

Centurion University of Technology and Management
Andhra Pradesh

M. Tech and M. Tech by Research
in
Computer Science and Engineering (CSE)

COURSE STRUCTURE & SYLLABUS

BASKET – I, II, III & IV

Centurion
UNIVERSITY
Shaping Lives...
Empowering Communities...

School of Engineering & Technology

2025-26

Index

Programme Objectives: Job/Higher studies/Entrepreneurship

POs: Engineering Graduates will be able to;

1. **PO1:** Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **PO2:** Problem Analysis: Identify, formulate, review research literature, and analyze complex engineering problems to reach substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **PO3:** Design/Development of Solutions: Design solutions for complex engineering problems and design system components or processes that meet specified needs with appropriate consideration for public health and safety, and cultural, societal, and environmental considerations.
4. **PO4:** Conduct Investigations of Complex Problems: Use research-based knowledge and research methods including the design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **PO5:** Modern Tool Usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **PO6:** The Engineer and Society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to professional engineering practice.
7. **PO7:** Environment and Sustainability: Understand the impact of professional engineering solutions in societal and environmental contexts, and demonstrate knowledge of, and need for, sustainable development.
8. **PO8:** Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **PO9:** Individual and Team Work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **PO10:** Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **PO11:** Project Management and Finance: Demonstrate knowledge and understanding of engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **PO12:** Life-long Learning: Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

PEOs/PSOs

1. **PEO1:** Prepare students to build competency in current technology and its application to meet the industry need for skilled Engineer
2. **PEO2:** Provide students with strong foundational concepts and also domain knowledge to pursue research to build solutions or systems of varying complexity to solve the problems identified
3. **PEO3:** Enable graduates to innovate, bring new ideas, and become entrepreneurs
4. **PSO1:** Graduates will develop hands-on skills related to Manufacturing, Design, Welding, and Automobile field.
5. **PSO2:** Build software competencies for product design, simulation, analysis, and manufacturing.
6. **PSO3:** Become a valuable technocrat to be fit for the industry or entrepreneur through manufacturing and management practices.

Course Outcomes	**Attributes**
CO1	Knowledge
CO2	Analytical skill and Critical Thinking
CO3	Problem Solving and Decision taking ability
CO4	Use of Tool, Design and Development (Hands-on/Technical skill)
CO5	Research
CO6	Environment and Sustainability
CO7	Ethics & Teamwork
CO8	Soft skill

Course Structure
M. Tech in CSE
Semester - I

Course Code	Course Title	T-P-P	Credits
MTAP1013	Advanced Java Technologies	2+2+0	4
MTAP1009	Applied AI and ML using Python	2+2+0	4
MTAP1014	Relational and Distributed Database Design	0+2+0	2
MTAP1015	Principles of Formal Languages & Compiler Design	2+2+0	4
Based on the course	Domain Elective I	2+2+0	4

Semester - II

Course Code	Course Title	T-P-P	Credits
MTAP1016	Android App Development using Kotlin	2+2+0	4
MTAP1017	Generative AI & its Applications	0+2+0	2
MTAP1018	Advanced Data Structures	2+2+0	4
MTAP1019	IoT Networking & Communication Protocols	2+2+0	4
Based on the course	Domain Elective II	2+2+0	4

Semester - III

Course Code	Course Title	T-P-P	Credits
CUTM2378	Research Methodology and IPR	2+2+0	4
MTAP1020	Customer Experience: Design and Development	0+2+0	2
MTAP1021	Dissertation-I / Industrial Project	0+0+4	4
Based on the	Domain Elective III	2+2+0	4

course	Domain Elective IV	2+2+0	4

Semester – IV

Course Code	Course Title	T-P-P	Credits
MTAP1022	Dissertation-II	0+0+14	14

Domain Electives: AI/ML

Course Code	Course Title
MTAP1023	Prompt Engineering
MTAP1024	RAG & Agentic AI
MTAP1025	Predictive Analytics using Machine Learning
MTAP1010	Computer Vision and Deep Learning
MTAP1026	Drone Imaging and Spectral Analysis

Domain Electives: DS

Course Code	Course Title
MTAP1027	DS with Tableau
MTAP1023	Prompt Engineering
MTAP1024	RAG & Agentic AI
MTAP1025	Predictive Analytics using Machine Learning

M. Tech by Research in CSE

Semester I

Course Code	Course Title	T-P-P	Credits
MTAP1009	Applied AI and ML using Python	2+2+0	4
MTAP1010	Computer Vision and Deep Learning	2+2+0	4
MTAP1005	Thesis Research I	0+0+8	8

Semester II

Course Code	Course Title	T-P-P	Credits
MTAP1011	Server-side Programming using NodeJS and MongoDB	2+2+0	4
MTAP1012	Client-side Programming using ReactJS	2+2+0	4
CUTM2378	Research Methodology and IPR	4+0+0	4
MTAP1006	Thesis Research II	0+0+8	8

Semester III

Course Code	Course Title	T-P-P	Credits
MTAP1007	Thesis Research III	0+0+16	16

Semester IV

Course Code	Course Title	T-P-P	Credits
MTAP1008	Publication / Patent	0+0+8	8
MTAP1029	Thesis Submission and Oral Defense	0+0+8	8

Applied AI and ML using Python

Course Code	Course Title	Credits	Type (T+P+P)
MTAP1009	Applied AI and ML using Python	4	2+2+0

Course Description:
This course introduces Artificial Intelligence and Machine Learning concepts and techniques applied to various fields such as medical science, Agriculture, Automobile, mining, and many more. Students will gain hands-on experience in developing and deploying an ML model using Python Programming.

Course Objectives:
1. Understand the meaning, purpose, scope, stages, applications, and effects of AI and ML.
2. Explore important packages of Python, such as numpy, scipy, OpenCV, and scikit-learn.
3. Implement Machine Learning models such as supervised and unsupervised learning applications using modern frameworks.

Course Outcomes (COs):
1. **CO1**: Able to gain knowledge on AI and ML solutions in their respective fields of study. (Understand, Remember)
2. **CO2**: Able to analyze several problems and apply AI and ML techniques to solve them. (Apply, Create)
3. **CO3**: Ability to design prediction and classification models (Apply, Create, and Analyze)

COURSE CONTENTS

Module 1 – Applications of AI & ML and Environmental Setup
- AI vs. Human Intelligence, Introduction to Artificial Intelligence, Applications and limitations of current AI systems.
- Applications of Machine Learning in different fields (Medical science, Agriculture, Automobile, mining, and many more).
- Supervised vs Unsupervised Learning based on problem Definition.
- Understanding the problem and its possible solutions using various datasets.
- Python libraries suitable for Machine Learning (numpy, scipy, scikit-learn, opencv)
- Environmental setup and Installation of important libraries.

Experiments:
- 1.1 Introduction to NumPy module;
- 1.2 Introduction to Pandas module;

Module 2 – Supervised Learning (Regression)
- Linear Regression
- Non-linear Regression
- Polynomial Regression
- Logistic Regression
- Model Evaluation in Regression

- Evaluation Metrics in Regression Models
- Multiple Linear Regression
- Implementation of a regression model on various datasets.

Experiments:

2.1 Implementation of Simple Linear Regression
2.2 Implementation of Non-Linear regression on various datasets;
2.3 Implementation of Multiple Linear Regression on various datasets;
2.4 Comparison between Linear and Non-linear regression;

Module 3 – Supervised Learning (Classification)

- Defining Classification Problem with various datasets.
- Mathematical formulation of the K-Nearest Neighbour Algorithm for binary classification.
- Implementation of the K-Nearest Neighbour Algorithm using sci-kit learn.
- Classification using the Random Forest and the Decision Tree.
- Construction of decision trees based on entropy.
- Implementation of Random Forest and Decision Trees for various datasets.
- Classification using Support Vector Machines.
- SVM for Binary classification
- Regulating different functional parameters of SVM using sci-kit learn.
- SVM for multi-class classification.
- Implementation of SVM using various datasets.
- Implementation of Model Evaluation Metrics using sci-kit learn and IRIS datasets.

Experiments:

3.1 Implementation of the k-Nearest Neighbor algorithm on various datasets;
3.2 Implementation of the Random Forest algorithm on various datasets;
3.3 Implementation of Tree construction using Decision Tree Classifier on various datasets;
3.4 Implementation of Finding Accuracy using Decision Tree on various datasets;
3.5 Implementation of SVM Classification using a Binary class;
3.6 Implementation of SVM Classification using various multiclass datasets;
3.7 Implementation of Evaluation metrics;

Module 4 - Unsupervised Learning

- Defining clustering and its application in ML.
- Mathematical formulation of K-Means Clustering.
- Defining K value and its importance in K-Means Clustering.
- Finding an appropriate K value using the elbow technique for a particular problem.
- Implementation of K-Means clustering for various datasets.
- Dimensionality Reduction using PCA
- Singular Value Decomposition (SVD)

Experiments:

4.1 Implementation of K-Means clustering;
4.2 Implementation of Evaluation metrics;

Module 5 - Ensemble Learning

- Introduction to Ensemble Learning
- Categories of Ensemble Learning
- Types of Ensemble Methods
- Reducing errors with ensemble learning

- Introduction to Recommender Systems

Experiments:

5.1 Implementation of Ensemble Learning for Disease Classification

5.2 Implementation of Ensemble Learning for Sentiment Analysis

Module 6 – Introduction to Neural Networks

- Single Layer Neural Network, Multilayer Perceptron
- Back Propagation Learning, Functional Link Artificial Neural Network
- Recurrent Neural Networks, Deep Learning, Convolutional Neural Networks

Experiments:

6.1 Implementation of a Basic Neural Network Using NumPy;

Textbooks and References:

- Ethem Alpaydin, Introduction to Machine Learning, Second Edition, 2021.
- S Sridhar and M Vijayalakshmi, "Machine Learning", Oxford University Press, 2021.
- M N Murty and Ananthanarayana V S, "Machine Learning: Theory and Practice", Universities Press (India) Pvt. Limited, 2024.

Web Resources:

- https://developers.google.com/machine-learning/crash-course
- https://scikit-learn.org/stable/
- https://www.kaggle.com/learn/intro-to-machine-learning
- https://www.oreilly.com/library/view/hands-on-machine-learning/9781492032632/
- https://www.oreilly.com/library/view/introduction-to-machine/9781449369880/
- https://www.drssridhar.com/?page_id=1053
- https://www.universitiespress.com/resources?id=9789393330697
- https://onlinecourses.nptel.ac.in/noc23_cs18/preview
- https://www.geeksforgeeks.org/machine-learning/
- https://www.w3schools.com/python/python_ml_getting_started.asp
- https://www.tutorialspoint.com/machine_learning/index.htm
- https://www.youtube.com/watch?v=aircAruvnKk
- https://www.youtube.com/watch?v=IHZwWFHWa-w
- https://www.youtube.com/watch?v=Ilg3gGewQ5U

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	1	1	2	-	-	-	-	-	-	1	3	2	2
CO2	3	3	3	2	3	-	-	-	1	-	-	2	3	3	3
CO3	3	3	3	3	3	-	-	-	2	-	-	2	3	3	3

Computer Vision and Deep Learning

Course Code	Course Title	Credits	Type (T+P+P)
MTAP1010	Computer Vision and Deep Learning	4	2+2+0

Course Description:
This course introduces deep learning concepts and techniques specifically applied to image analytics. Students will gain hands-on experience in developing and deploying convolutional neural networks (CNNs) for image classification, object detection, and image generation.

Course Objectives:
1. Understand deep learning fundamentals and CNN architectures.
2. Develop and evaluate CNNs for image classification and object detection.
3. Implement generative models and deploy deep learning applications using modern frameworks.

Course Outcomes (COs):
1. **CO1**: Explain the basic concepts of neural networks, including activation functions and backpropagation. (Understand, Remember)
2. **CO2**: Design and implement CNN architectures for various image analytics tasks. (Apply, Create)
3. **CO3**: Utilize advanced CNN architectures and transfer learning for enhanced performance. (Apply, Analyze)
4. **CO4**: Develop and evaluate models for image classification, object detection, and segmentation. (Apply, Evaluate)
5. **CO5**: Implement generative models and deploy deep learning applications using TensorFlow, Keras, and PyTorch. (Apply, Create)

Syllabus:

Module 1: Introduction to Deep Learning
- Basics of Neural Networks, Activation Functions
- Multilayer Perceptrons (MLPs), Backpropagation, and Gradient Descent
- Feedforward Neural Networks (FFNs), Representation Power of FFNs

Experiments:
1.1 Implement a simple backpropagation network.
1.2 Implement a simple back-propagation network with gradient descent.
1.3 Implement Feedforward Backpropagation Neural Network.

Module 2: Introduction to Computer Vision
- Introduction to Computer Vision, Color models
- Basic image processing in the spatial domain
- Image Preprocessing, Segmentation, and Quality Enhancement

Experiments:
2.1 Install OpenCV
2.2 Basic image processing operations (Image scaling, Image rotation, Thresholding, Histogram processing, Smoothening, Sharpening)

Module 3: Convolutional Neural Networks (CNNs)

- Architecture of CNNs
- Convolutional Layers, Pooling Layers
- Backpropagation in CNNs
- CNN Architecture for Image Classification

Experiments:

3.1 Build a basic CNN for image classification using Pytorch.
3.2 Build a basic CNN for image classification using Tensorflow

Module 4: Advanced CNN Architectures for Image Classification

- Transfer Learning and Pre-trained Models
- ImageNet (Alexnet, VGG, ResNet, ResNetXt, Inception) and EfficientNet Models using either Tesorflow/Pytorch
- Finetuning CNNs and Visualizing CNNs
  - The training can be done either in Google Collab or Kaggle or on Local CPU
  - To be trained to run on GPU
  - Discussion on each model is to be presented before executing the model

Experiments:

4.1 Implement AlexNet for image classification.
4.2 Implement VGG for image classification.
4.3 Implement ResNet for image classification.
4.4 Implement ResNetXt for image classification.
4.5 Implement Inception for Image classification.
4.6 Implement EfficientNet for Image classification.
4.7 Implement MobileNet for Image Classification

Module 5: Segmentation and Object Detection

- CNNs for Object Detection: Two-stage Models
- CNNs for Object Detection: Single-stage Models
- CNNs for Segmentation

Experiments:

5.1 Implement YOLO for semantic segmentation.
5.2 Implement YOLO for real-time object detection

Module 6: Recurrent Neural Networks and their use in Vision

- Recurrent Neural Networks: Introduction
- Backpropagation in RNNs
- LSTMs and GRUs
- Video Understanding using CNNs and RNNs

Experiments:

6.1 Implement and compare LSTM and GRU (Take a case study)

Textbooks and References:

- "Deep Learning" by Ian Goodfellow, Yoshua Bengio, and Aaron Courville
- "Deep Learning with Python" by François Chollet

- Computer Vision: Algorithms and Applications, 2nd ed. Richard Szeliski, The University of Washington, 2022.

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2	1	2	-	-	-	-	-	-	1	3	2	3
CO2	3	3	3	2	2	-	-	-	1	-	-	2	3	3	3
CO3	3	3	3	2	3	-	-	-	1	-	-	2	3	3	3
CO4	3	3	3	3	3	1	1	-	2	-	2	2	3	3	3
CO5	3	3	3	3	3	2	2	3	2	2	2	3	3	3	3

Advanced Java Technologies (60 hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
MTAP1013	Advanced Java Technologies	4	2+2+0

Course Description:
This course delves into advanced topics in Java programming, including multithreading, networking, database connectivity, and Java frameworks. Students will gain a deep understanding of Java's capabilities and how to apply them in real-world applications.

Course Objectives:
1. Master advanced Java programming concepts and techniques.
2. Implement Java-based solutions for web, and database applications.
3. Apply Java frameworks and best practices in real-world scenarios.

Course Outcomes (COs):
1. CO1: Understand and apply advanced object-oriented programming concepts in Java. (Understand, Apply)
2. CO2: Implement servlet to control web applications. (Apply, Analyse)
3. CO3: Develop dynamic web pages using Java Server Page. (Apply, Create)
4. CO4: Integrate Java applications with databases using JDBC and manage transactions. (Apply, Evaluate)
5. CO5: Utilize Java EE and frameworks to build robust enterprise applications. (Apply, Create)

Module Breakdown:

Module 1: Introduction to Web Architecture (9 hours)

Theory
- Overview of Web Architecture, Client-Server Model, HTTP Protocol Basics, Web Servers and Application Servers, Introduction to MVC Architecture.

Practice
- Experiment 1.1: Set up a simple HTTP server and client.
- Experiment 1.2: Create a basic web page and deploy it on a web server.
- Experiment 1.3: Implement a simple MVC pattern in a web application.
- Experiment 1.4: Analyze HTTP request and response headers using browser tools.
- Experiment 1.5: Configure and deploy an application on an Apache Tomcat server.

Module 2: Introduction to GitHub (9 hours)
Theory
Version Control Basics, Git Commands and Workflow, GitHub Repository Management, Branching and Merging, Collaborating with GitHub.

Practice
- Experiment 2.1: Initialize a Git repository and commit changes.
- Experiment 2.2: Push local repository to GitHub and manage remote repositories.
- Experiment 2.3: Create and merge branches.
- Experiment 2.4: Resolve merge conflicts.
- Experiment 2.5: Collaborate with others using pull requests and issues on GitHub.

Module 3: Java Database Connectivity (JDBC) (9 hours)
Theory
- **Topics:** Introduction to Java Database Connectivity (JDBC), JDBC Architecture and Drivers, Establishing Database Connections, Executing SQL Queries and Updates, Handling ResultSets, Prepared Statements and Callable Statements, Batch Processing in JDBC, Transaction Management, Connection Pooling

Practice
- Experiment 3.1: Set up a database and connect to it using JDBC.
- Experiment 3.2: Execute SQL SELECT queries and process the results.
- Experiment 3.3: Perform INSERT, UPDATE, and DELETE operations using JDBC.
- Experiment 3.4: Handle SQL exceptions and errors.
- Experiment 3.5: Develop a simple CRUD application using JDBC.
- Experiment 3.6: Use prepared statements and callable statements
- Experiment 3.7: Implement batch processing in JDBC
- Experiment 3.8: Manage transactions effectively
- Experiment 3.9: Utilize connection pooling for efficient database access.

Module 4: Introduction to Servlets (9 hours)
Theory
- **Topics:** Introduction to Servlets, Servlet Lifecycle, Handling Requests and Responses, Servlet Configuration and Context, Session Management.

Practice
- Experiment 4.1: Create and deploy a simple servlet.
- Experiment 4.2: Handle GET and POST requests in a servlet.
- Experiment 4.3: Implement session tracking using cookies and HTTP sessions.
- Experiment 4.4: Use servlet context and configuration for initialization parameters.
- Experiment 4.5: Develop a login system using servlets and session management.

Module 5: Advanced Servlet Concepts (9 hours)
Theory
Request Dispatching and Redirecting, Servlet Filters, Servlet Listner, Asynchronous Servlets, Error Handling in Servlets, Security and Authentication in Servlets

Practice
- Experiment 5.1: Implement request forwarding and redirection.
- Experiment 5.2: Create and configure a servlet filter.
- Experiment 5.3: Develop an asynchronous servlet for long-running tasks.
- Experiment 5.4: Implement custom error pages for handling different HTTP errors.
- Experiment 5.5: Secure a servlet using basic authentication and HTTPS.

Module 6: Introduction to JSP (9 hours)
Theory
Basics of JavaServer Pages (JSP), JSP Lifecycle, JSP Directives, Scriptlets, and Expressions, JSP Implicit Objects, Using JavaBeans in JSP,JSP Tag Libraries (JSTL), Custom Tags in JSP, Expression Language (EL), JSP and MVC Architecture, Error Handling in JSP

Practice

- Experiment 6.1: Create a basic JSP page and deploy it.
- Experiment 6.2: Use JSP scriptlets to embed Java code in HTML.
- Experiment 6.3: Access JSP implicit objects to handle requests and responses.
- Experiment 6.4: Integrate a JavaBean in a JSP page.
- Experiment 6.5: Create a simple form processing application using JSP and JavaBeans.
- Experiment 6.6: Use JSTL core tags to manage control flow and iteration.
- Experiment 6.7: Create and use custom JSP tags.
- Experiment 6.8: Utilize EL to access Java objects and properties.
- Experiment 6.9: Implement MVC architecture with JSP as the view.
- Experiment 6.10: Develop error handling mechanisms in JSP

Module 7: Introduction to Hibernate (6 hours)

Theory

Basics of Hibernate ORM, Hibernate Architecture, Configuring Hibernate, Mapping Entities to Tables, CRUD Operations with Hibernate

Practice

- Experiment 7.1: Set up Hibernate in a Java project.
- Experiment 7.2: Map a Java class to a database table using annotations.
- Experiment 7.3: Perform CRUD operations with Hibernate.
- Experiment 7.4: Configure Hibernate using XML.
- Experiment 7.5: Implement relationships (one-to-many, many-to-many) in Hibernate

Textbooks and References:

- "Core Servlets and JavaServer Pages" by Marty Hall and Larry Brown

CO-PO-PSO Mapping:

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2	2	3	-	-	-	1	2	2	2	3	2	3
CO2	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO3	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO4	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO5	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3

Relational and Distributed Database Design

Course Code	Course Title	Credits	Type (T+P+Pj)
MTAP1014	Relational and Distributed Database Design	2	0+2+0

Course Objectives

- To strengthen understanding of relational database design and normalization using MySQL.
- To apply distributed database principles for scalability and performance.
- To integrate relational and big data systems for analytical processing.
- To develop hands-on skills in MySQL, Hadoop, Hive, and SparkSQL.

Course Outcomes (COs)

After completing this course, learners will be able to:
CO1: Design normalized relational databases using advanced SQL concepts.
CO2: Implement relational and distributed database design principles.
CO3: Apply data fragmentation, replication, and distributed querying techniques.
CO4: Integrate RDBMS with big data tools for analytics and visualization.
CO5: Perform large-scale data analysis using Hive and SparkSQL.
CO6: Evaluate performance, consistency, and fault tolerance in distributed database systems.

Module 1: Relational Database Design and Implementation using MySQL
Topics:

- Review of DBMS Architecture and Relational Model
- ER Modelling and Conversion to Relational Schema
- Keys, Constraints, and Normalization (up to 3NF)
- Implementation of Database Schemas in MySQL

Experiments:

1. Design and implement an ER diagram for a business application (e.g., online shopping or hospital management) in MySQL.
2. Create tables with primary key, foreign key, and check constraints.
3. Normalize an unnormalized dataset into 3NF and test referential integrity.

Module 2: Advanced SQL and Query Optimization
Topics:

- Subqueries, Views, Joins, Triggers, and Stored Procedures
- Indexing and Query Optimization Techniques
- Transaction Management (COMMIT, ROLLBACK, SAVEPOINT)
- Performance Tuning using EXPLAIN in MySQL

Experiments:

1. Write complex SQL queries using nested subqueries, joins, and views.

2. Create and test triggers and stored procedures for automation.
3. Analyze query performance using indexing and EXPLAIN plan in MySQL.

Module 3: Distributed Database Concepts and Architecture
Topics:

- Distributed Database Architecture (Client-Server, Peer-to-Peer)
- Data Distribution: Fragmentation, Replication, and Allocation
- Distributed DBMS Components and Data Transparency
- Introduction to CAP Theorem and Distributed SQL

Experiments:

1. Simulate distributed database nodes using multiple MySQL instances.
2. Implement horizontal and vertical fragmentation on a large dataset.
3. Demonstrate data replication and synchronization between two MySQL servers.

Module 4: Distributed Query Processing and Transaction Management
Topics:

- Distributed Query Decomposition and Optimization
- Distributed Join Strategies and Cost Models
- Two-Phase Commit Protocol (2PC)
- Concurrency Control and Deadlock Handling

Experiments:

1. Execute distributed joins and measure query execution times across fragmented datasets.
2. Implement a Two-Phase Commit protocol simulation in MySQL or Python.
3. Demonstrate concurrency control using transaction isolation levels and deadlock detection.

Module 5: Integration of RDBMS with Big Data Frameworks
Topics:

- Big Data Concepts and Hadoop Ecosystem Overview
- Data Transfer from RDBMS to HDFS using Sqoop
- Hive as a Data Warehouse over Hadoop
- Connecting MySQL with Hive for Analytical Queries

Experiments:

1. Import relational data from MySQL to HDFS using Apache Sqoop.
2. Create external Hive tables mapped to HDFS files and query data using HiveQL.
3. Perform analytical queries (aggregation, joins, group-by) using Hive connected to MySQL.

Module 6: Distributed Big Data Analytics using SparkSQL
Topics:

- SparkSQL and DataFrames

- Distributed Query Execution in Spark
- Integrating Spark with MySQL (JDBC)
- Case Study: Hybrid Analytical Pipeline

Experiments:

1. Load relational data into Spark DataFrames and perform distributed SQL operations.
2. Integrate Spark with MySQL using JDBC and run analytical queries on joined datasets.
3. Develop an end-to-end mini project combining MySQL, Hive, and Spark for analytics on large data.

Suggested Tools / Software

Category	**Tools / Technologies**
RDBMS	MySQL / PostgreSQL / DBeaver
Big Data Frameworks	Hadoop, Hive, Sqoop, SparkSQL
Programming	Python (PySpark, SQLAlchemy)
Deployment/Simulation	Docker / VirtualBox / Local multi-node setup

Mini Project (Capstone Practical)
Title Examples:

- *Design and Implementation of a Distributed Retail Data Warehouse using MySQL, Hive, and Spark.*
- *Integration of Relational and Big Data Systems for Social Media Analytics.*

Deliverables:

- Database design documentation
- Implementation scripts
- Analytical report with visualized results

Mini Project Suggestions

1. **Retail Data Warehouse:** Design a distributed data warehouse integrating MySQL, Hive, and Spark for sales analytics.
2. **IoT Data Management:** Collect sensor data, store in MySQL, and analyze via SparkSQL for trend patterns.
3. **Social Media Analytics:** Import relational user-post data into Hadoop and analyze engagement patterns using HiveQL.
4. **Healthcare System:** Implement distributed patient data storage and analyze hospital statistics using Spark.

CO–PO and CO–PSO Mapping Matrix

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PSO1	PSO2	PSO3
CO1	3	2	2	1	3	1	2	1	1	3	2	2
CO2	3	3	3	2	3	1	2	2	1	3	2	3
CO3	2	3	3	3	3	1	2	1	1	3	3	3
CO4	2	3	3	2	3	1	3	1	2	2	3	3
CO5	2	2	2	3	3	1	3	1	2	2	3	2
CO6	3	3	2	3	3	2	2	2	2	3	3	3

Principles of Formal Languages and Compiler Design

Course Code	Course Title	Credits	Type (T+P+P)
MTAP1015	Principles of Formal Languages and Compiler Design	4	2+2+0

Course Objectives

1. To understand formal languages, automata, and grammar classifications.
2. To explore the structure and phases of a compiler.
3. To design and implement lexical and syntax analyzers.
4. To understand syntax-directed translation and code generation.
5. To develop a mini compiler integrating all compiler phases.

Course Outcomes (COs)

CO No.	Course Outcome Description
CO1	Design finite automata and classify formal grammars.
CO2	Simplify and transform context-free grammars and implement parsing techniques.
CO3	Develop lexical and syntax analyzers using programming tools.
CO4	Apply syntax-directed translation and generate intermediate code.
CO5	Perform code optimization and generate target machine code.

CO–PO and CO–PSO Mapping Matrix

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PSO1	PSO2
CO1	3	3	2	1	2	–	–	–	1	3	2
CO2	2	3	3	2	2	–	–	–	1	3	2
CO3	2	3	3	2	3	–	1	2	2	3	3
CO4	2	3	3	3	3	–	1	2	2	3	3
CO5	2	3	3	3	3	1	2	2	3	3	3

Detailed Syllabus

Module I – Formal Languages and Automata (10 Hours

Theory Topics:

- Basics of Formal Languages – Alphabets, Strings, Languages
- Grammar and its classifications – Chomsky Hierarchy
- Regular Languages and Regular Expressions
- Finite Automata – DFA, NFA, and ε-NFA
- Conversion between RE, NFA, and DFA
- DFA Minimization

Learning Outcome:
Understand and construct various types of automata for given regular languages.

Experiments:

1. Program to identify valid tokens in a given string using Regular Expressions.
2. Program to convert a Regular Expression to NFA.
3. Program to convert NFA to DFA.
4. Program to minimize a given DFA using state equivalence method.
5. Simulation of string acceptance by DFA.

Module II – Context-Free Grammars and Parsing Techniques (10 Hours)

Theory Topics:

- Context-Free Grammar (CFG) – Definition and Examples
- Derivation Trees, Ambiguity, and Simplification of CFG
- Removal of Null Productions, Unit Productions, and Useless Symbols
- Normal Forms: Chomsky Normal Form (CNF) and Greibach Normal Form (GNF)
- Pushdown Automata (PDA): Definition and Construction
- Parsing: Top-Down and Bottom-Up techniques
- LL(1) and LR(0) parsing introduction

Learning Outcome:
Develop CFGs for simple languages and construct equivalent PDAs.

Laboratory Experiments:
1. Program to remove left recursion and left factoring from a grammar.
2. Program to convert CFG to Chomsky Normal Form (CNF).
3. Construction and simulation of a Pushdown Automaton (PDA).
4. Implementation of Recursive Descent Parser for arithmetic expressions.

Module III – Compiler Design: Lexical and Syntax Analysis (10 Hours)

Theory Topics:

- Phases of Compiler and Compiler Structure
- Lexical Analysis: Role of Lexical Analyzer, Tokens, Lexemes, Patterns
- Specification and Recognition of Tokens using Regular Expressions
- Lexical Analyzer Generators (Lex / Flex)
- Syntax Analysis: Role of Parser, Parse Trees, and Derivations
- Recursive Descent Parser and Operator Precedence Parser
- Error Detection and Recovery

Learning Outcome:
Implement lexical and syntax analysis phases of a compiler.

Laboratory Experiments:
1. Design and implement a Lexical Analyzer using Regular Expressions.
2. Implementation of a Symbol Table for identifiers and their attributes.
3. Program to implement Operator Precedence Parser.
4. Implementation of Error Detection and Recovery techniques during parsing.

Module IV – Syntax Directed Translation and Intermediate Code Generation (10 Hours)

Theory Topics:

- Syntax Directed Definitions (SDD) and Translation Schemes
- Attribute Grammars – Synthesized and Inherited Attributes
- Abstract Syntax Tree (AST) construction
- Intermediate Representations: Quadruples, Triples, and Indirect Triples
- Intermediate Code Generation for Control and Arithmetic Statements
- Type Checking and Type Conversions

Learning Outcome:
Design syntax-directed translation schemes and generate intermediate code.

Laboratory Experiments:
1. Implementation of Syntax Directed Translation for arithmetic expressions.
2. Generation of Three Address Code (TAC) for assignment statements.
3. Construction of Abstract Syntax Tree (AST) for a given expression.

Module V – Code Optimization and Code Generation (10 Hours)

Theory Topics:

- Code Optimization: Need and Objectives
- Basic Blocks and Flow Graphs
- Optimization Techniques: Constant Folding, Dead Code Elimination, Common Subexpression Elimination, Loop Optimization
- Peephole Optimization
- Code Generation: Issues, Register Allocation, and Target Code

- Symbol Table Management

Learning Outcome:
Apply optimization techniques and generate target machine code.

Laboratory Experiments:
1. Implement Constant Folding and Dead Code Elimination.
2. Demonstrate Common Subexpression Elimination and Loop Optimization.
3. Implement Target Code Generation for arithmetic expressions.
4. **Mini Project:** Design and implement a simple compiler that integrates all compiler phases (Lexical → Syntax → Intermediate → Optimization → Code Generation).

Software/Hardware Requirements

- **Programming Languages:** C / C++ / Java / Python
- **Tools:** Lex / Flex, Yacc / Bison (or equivalent)
- **Platform:** Linux / Windows

Text Books

1. Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman – *Compilers: Principles, Techniques and Tools*, Pearson.
2. Peter Linz – *An Introduction to Formal Languages and Automata*, Jones & Bartlett.