

R23 – MINOR (IT) COURSE STRUCTURE

S. No	Course Code	Course Title	Contact Hours/Week			Credits	Scheme of Valuation		
			L	T	P		CIE	SEE	Total
Theory Courses									
1	23ITM1	Linux Essentials	3	0	0	3	30	70	100
2	23ITM2	Database Management System Concepts	3	0	0	3	30	70	100
3	23ITM3	Full Stack Development using React (MERN)	3	0	0	3	30	70	100
4	23ITM4	Introduction to Generative AI and Deep Learning	3	0	0	3	30	70	100
5	23ITM5	Big Data Analytics for Engineering Applications	3	0	0	3	30	70	100
Laboratory Courses									
6	23ITM6	SQL Lab	0	0	3	1.5	30	70	100
7	23ITM7	Full Stack Development using React (MERN) Lab	0	0	3	1.5	30	70	100
8	23ITM8	Data Analytics Lab	0	0	3	1.5	30	70	100

*** Course Selection Requirements for attaining Minor Degree:**

- Theory Courses: **Choose any 3 out of 5.**
- Lab Courses: **Choose any 2 out of 3 that that correspond to the selected theory courses.**
- Additional Requirement: **Complete two 12-week MOOCs/NPTEL courses (3 credits each).**
- ***Total Credits to be attained: 18 Credits***

B.Tech.(IT)
Minor

23ITM1-LINUX ESSENTIALS

L	T	P	Cr.
3	0	0	3

Prerequisites: NIL

Course Objectives: This course meant for the students who want to build their career in Linux System Administration domain. The student who completed this course possesses the fundamental knowledge and proven skills in the area of Linux Essentials.

COURSE OUTCOMES (COs): At the end of this course, the student will be able to

- CO1:** Demonstrate Linux Utilities. **(Knowledge-L1)**
- CO2:** Identify the Basics of using the Linux command line **(Understanding- L2)**
- CO3:** Create, Search and extract data from files in the home directory. **(Apply- L3)**
- CO4:** Familiarity in working with hardware components, server computers, networking configuration. **(Understand- L2)**
- CO5:** Understanding and manipulating file permissions and ownership settings. **(Knowledge-L1)**

UNIT-I: The Linux Community and a Career in Open Source.

Linux Evolution and Popular Operating Systems, Major Open Source Applications, Open Source Software and Licensing, ICT Skills and Working in Linux.

The following is a partial list of the used files, terms and utilities:

Debian, Ubuntu (LTS), CentOS, openSUSE, Red Hat, SUSE, Linux Mint, Scientific Linux, RaspberryPi, Raspbian, Android.

OpenOffice.org, LibreOffice, Thunderbird, Firefox, GIMP, Nextcloud, ownCloud, Apache HTTPD, NGINX, MariaDB, MySQL, NFS, Samba, C, Java, JavaScript, Perl, shell, Python, PHP, dpkg, apt-get, rpm, yum, Copyleft, Permissive, GPL, BSD, Creative Commons, Free Software, Open Source Software, FOSS, FLOSS, Open source business models, Using a browser, privacy concerns, configuration options, searching the web and saving content, Terminal and console, Password issues, Privacy issues and tools, Use of common open source applications in presentations and projects.

UNIT-II: Finding Your Way on a Linux System.

Command Line Basics, Running help commands and navigation of the various help systems. Using Directories and Listing Files, Creating, Moving and Deleting Files

The following is a partial list of the used files, terms and utilities: Bash, echo, history, PATH environment variable, export, type, man, info, Common options for ls, Recursive listings, cd, . and .., home and ~, /usr/share/doc/, locate, mv, cp, rm, touch, mkdir, rmdir

UNIT-III: The Power of the Command Line.

Archiving Files on the Command Line, Searching and Extracting Data from Files, Turning Commands into a Script.

The following is a partial list of the used files, terms and utilities: tar, Common tar options, gzip, bzip2, xz, zip, unzip, grep, less, cat, head, tail, sort, cut, wc, - #! (shebang), /bin/bash, Variables, Arguments, for loops, echo, Exit status.

UNIT-IV: The Linux Operating System

Knowledge of major operating systems and Linux distributions. Familiarity with the components that go into building desktop and server computers. Where various types of information are stored on a Linux system. Querying vital networking configuration and determining the basic requirements for a computer on a Local Area Network (LAN).

The following is a partial list of the used files, terms and utilities: GUI versus command line, desktop configuration, Maintenance cycles, beta and stable, Motherboards, processors, power supplies, optical drives, peripherals, Hard drives, solid state disks and partitions, /dev/sd*, Drivers, ps, top, free, syslog, dmesg, /etc/, /var/log/, /boot/, /proc/, /dev/, /sys/, r route, ip route show, ifconfig, ip addr show, netstat, ss, /etc/resolv.conf, /etc/hosts, IPv4, IPv6, ping, host.

UNIT-V: Security and File Permissions.

Basic Security and Identifying User Types, Creating Users and Groups, Managing File Permissions and Ownership, Special Directories and Files

The following is a partial list of the used files, terms and utilities: /etc/passwd, /etc/shadow, /etc/group, id, last, who, w, sudo, su, /etc/passwd, /etc/shadow, /etc/group, /etc/skel/, useradd, groupadd, passwd, ls -l, ls -a, chmod, chown, /tmp/, /var/tmp/ and Sticky Bit, ls -d, ln -s

Text Books:

1. Linux Essentials, 2nd Edition, Author: Christine Bresnahan , Publisher: Sybex

Reference Books:

1. **Linux Pocket Guide:** Essential Commands Linux Pocket Guide is a book written by Jason Cannon. It provides an organized.
2. **The Linux Command Line:** The Linux Command Line is a book written by William Shotts. The author takes you from your.
3. **Linux for Beginners:** An Introduction to the Linux Operating System and Command Line Linux for Beginners is a book.
4. **Linux Command Line and Shell Scripting Bible,** 3rd Edition Linux Command Line and Shell Scripting Bible is a reference.

E-References:

1. Linux Essentials by CISCO Academy.

B.Tech.(IT)
Minor

**23ITM2- PRINCIPLES OF DATABASE
MANAGEMENT SYSTEMS**

L	T	P	Cr.
3	0	0	3

Pre-requisite: Introduction to Programming, Data Structures.

Course Objective:

This course enables the students to know about Basic concepts of DBMS, Database Languages, Database Design, and Normalization Process.

Course Outcomes (CO): At the end of this course, the student will be able to:

CO1: Demonstrate the Basic Components of Database Management System. (**Understand -L2**)

CO2: Design Entity-Relationship Diagrams for different applications. (**Apply -L3**)

CO3: Examine the relational data model and various constraints. (**Apply- L3**)

CO4: Demonstrate Structured Query Language languages (SQL) to manage relational databases and perform various operations on the data. (**Apply- L3**)

CO5: Employ principles of normalization for effective database design & transaction management. (**Apply- L3**)

UNIT – I

Introduction: Purpose of Database Systems, Database System Vs File System, View of Data, Database Languages, Database Design, Database Architecture, Database users and administrators.

UNIT – II

Entity Relationship Model: ER model- Entity sets, Relationship sets, Attributes, Constraints: Mapping Cardinalities, Participation Constraints, Keys: Concepts of Super Key, Candidate Key, Primary Key, Generalization, Aggregation, Reduction to Relational Schemas.

UNIT – III

Relational Data Model: (Text Book-2): Introduction to the Relational Model, Integrity Constraints over Relations, Enforcing Integrity Constraints, Logical Database Design: ER to Relational, Introduction to Views, Relational Algebra

UNIT – IV

Introduction to SQL: Overview of the SQL Query Language, SQL Data Definition, Basic Structure of SQL Queries, Set Operations, Null Values, Aggregate Functions, Integrity Constraints in SQL, Datatypes in SQL

UNIT – V

Normalization: (Text Book-2): Functional Dependencies, Normal Forms - First, Second, Third Normal Forms, BCNF.

Transaction Management: (Text Book-2): The ACID Properties. Transactions and Schedules, Concurrent Execution of Transactions: Serializability, Lock-Based Concurrency Control: Strict Two-Phase Locking

Text Books:

1. Database System Concepts, 6th edition, Silberschatz, Korth, Sudarsan, TMH.
2. Database Management Systems, 3rd edition, Raghurama Krishnan, Johannes Gehrke, TMH

Reference Books:

1. Database Management System, 6th edition, RamezElmasri, Shamkant B. Navathe, Pearson

B.Tech.(IT)
Minor

**23ITM3-FULL STACK DEVELOPMENT USING
REACT (MERN)**

L	T	P	Cr.
3	0	0	3

Pre-requisites: Nil

Course Objectives (COs) : After completing this course, students will be able to:

- CO1:** Describe and apply the fundamental concepts of HTML and CSS to create structured and visually styled web pages. (**Knowledge-L1**)
- CO2:** Explain and utilize JavaScript functions, arrow functions, and DOM manipulation to implement interactive web elements. (**Understanding -L2**)
- CO3:** Illustrate the working of Node.js and Express.js and develop basic server-side applications. (**Applying-L3**)
- CO4:** Demonstrate the component-based architecture of React.js and build interactive UI components with state management. (**Applying-L3**)
- CO5:** Implement advanced React.js concepts such as hooks, routing, and API integration to develop full-stack web applications. (**Applying-L3**)

Unit 1: Introduction to Web Development - Basics of HTML & CSS

HTML Basics: Structure of an HTML document, Tags, Elements, Attributes, **HTML Forms and Inputs:** Form elements, Form validation, Input types, **CSS Basics-** Selectors, Properties, Box model, Colours, Fonts, **CSS Layouts:** Flexbox, Grid, Positioning, **Responsive Design:** Media queries, Introduction to Bootstrap

Unit 2: JavaScript Fundamentals – JavaScript Basics, Functions, DOM

JavaScript Basics: Purpose of JS, Dynamic and responsiveness, Variables, Data types, Operators, Control Structures, **Functions in JavaScript:** Function declarations, Arrow functions, Callbacks, **JavaScript Objects & Arrays:** Object properties, Array methods, **Document Object Model (DOM):** Selecting elements, modifying elements, Event handling, **Asynchronous JavaScript:** Promises, Async/Await, Event loop

Unit 3: Backend Development: Introduction to Node.js, Express.js, and MongoDB Introduction to

Node.js: Features, Installation, NPM (Node Package Manager),

File System & Modules: Built-in modules, File handling, Creating custom modules, **Express.js Basics:** Routing, Middleware, Request & Response handling , **Working with APIs:** REST API basics, Handling HTTP requests , **Introduction to MongoDB:** NoSQL vs SQL databases, MongoDB architecture and key features, CRUD operations in MongoDB (Create, Read, Update, Delete), Connecting MongoDB with Node.js using Mongoose, Schema design and data validation with Mongoose

Unit 4: Introduction to React.js, Components, State Management

Introduction to React: Features, Virtual DOM, JSX, React Components: Functional vs Class components, Props, Component hierarchy, State & Lifecycle Methods: useState, useEffect, Component lifecycle, Event Handling in React: Handling user inputs, Forms in React,

React Styling: Inline styles, CSS Modules, Styled-components

Unit 5: Hooks, Routing, API Integration

React Hooks: useContext, useReducer, useRef, **React Routing:** React Router, Nested Routes, Dynamic Routing, **Managing Global State:** Redux Basics, Context API, **API Integration in React:** Fetching data using Axios, Handling API responses, **Full Stack Integration:** Connecting React frontend with Node.js backend

Textbooks:

1. "Web Technologies: HTML, JavaScript, Node.js, and React" – Subrahmanyam Allamaraju, R. S. Bichkar – Wiley Publications
2. "Full Stack Web Development with MongoDB, Express, React, and Node" – Shama Hoque – BPB Publications

Reference Book:

1. "Full Stack Web Development for Beginners: Learn MERN Stack Development" – Ishan Goyal – BPB Publications

E-References:

1. Dev.to Community – <https://dev.to/> (Articles and discussions on web development)
2. GeeksforGeeks Web Development Section – <https://www.geeksforgeeks.org/web-development/>

B.Tech.(IT)
Minor

**23ITM4- INTRODUCTION TO GENERATIVE AI
AND DEEP LEARNING**

L	T	P	Cr.
3	0	0	3

Pre-requisite: Probability and Statistics, LATT, Machine Learning

Course Educational Objective: The objective of the course is to make students learn the fundamentals of generative AI, frameworks of deep learning and their applications.

Course Outcomes: At the end of this course, the student will be able to

CO1: Understand the evolution of AI and the significance of Deep Learning (**Understand-L2**)

CO2: Apply the concepts of Convolutional Neural Networks, Recurrent Neural Networks to Natural Language Processing and computer vision applications. (**Apply- L3**)

CO3: Apply the regularization techniques to improve the model performance. (**Apply- L3**)

CO4: Explore various types of generative modelling techniques. (**Apply- L3**)

CO5: Identify the ethical implications of generative AI. (**Understand- L2**)

UNIT 1:

Overview of Generative AI: History and evolution of AI/ML, Deep learning revolution, Transfer learning, History of Neural Natural Language Processing, Structure of Artificial Neural Networks, Steps in Training an Artificial Neural Network, Parameters and Hyper parameters, Back propagation.

Generative AI Applications: Applications in Various Fields: Art and Creativity, Image and Video Generation, Text Generation, Music Composition, Healthcare Finance. Real-world usecases and challenges in deploying generative AI models.

UNIT 2:

Convolutional Neural Networks Motivation, Convolution Operation, Types of layers, Pooling, Architecture of traditional RNN, Types and applications of RNN, Variants of RNNs, Word Embedding using Word2vec, Attention Mechanism in detail - Long Short-Term Memory Network (LSTM).

UNIT 3:

Networks and Models: Regularization for Deep Learning: L1 and L2, Dropout, Data Augmentation, Early Stopping, Autoencoders, Variational Autoencoders, latent space, Generative Adversarial Networks (GANs)- Deep Convolutional GAN (DCGAN), Conditional GAN.

UNIT 4:

Transformers and Large Language Models (LLMs): Language Models, Transformer Architecture, Motivation for Transformer, Architecture, Encoder-Decoder Architecture, Attention, Position-wise Feed-Forward Networks, Advantages and Limitations of Transformer Architecture.

UNIT 5:

Applications and Ethical Implications: Applications - ChatGPT Architecture, Google Bard, Claude 2, Falcon AI, LLaMa 2, Dolly 2, DALL-E 2, Midjourney. Ethics- Bias and Fairness in Generative Models.

Text Books:

1. Deep Learning, Ian Goodfellow, Yoshua Bengio and Aaron Courville, MIT Press, 2016
2. "Generative AI for everyone: Understanding the essentials and applications of this breakthrough technology". Altaf Rehmani .

Reference Books:

1. Artificial Neural Networks, Yegnanarayana, B., PHI Learning Pvt. Ltd, 2009.
2. Matrix Computations, Golub, G.,H., and Van Loan,C.,F, JHU Press,2013.

E-Resources:

1. <https://elearn.nptel.ac.in/shop/iit-workshops/completed/leveraging-generative-ai-for-teaching-programming-courses/?v=c86ee0d9d7ed>
2. <https://elearn.nptel.ac.in/shop/iit-workshops/completed/introduction-to-language-models/?v=c86ee0d9d7ed>
3. Swayam NPTEL: [https://onlinecourses.nptel.ac.in/noc22_cs22/previewB.Tech.\(ArtificialIntelligence and Data Science\) Deep Learning:](https://onlinecourses.nptel.ac.in/noc22_cs22/previewB.Tech.(ArtificialIntelligenceandDataScience)DeepLearning)

Pre-Requisite: Database Management Systems, Data Warehousing and Data Mining.

Course Outcomes: The objectives of this course are to introduce Understanding the process of distributed data (Structured, Semi-Structured and unstructured) that process the Terabytes of data using Hadoop Eco System Tools.

Course Outcomes: At the end of this course, the student will be able to

- CO1:** Identify Big Data and its Business Implications. (**Understand - L2**)
- CO2:** Process of distributed file system using Hadoop (**Apply - L3**)
- CO3:** Illustrate the MapReduce mechanism (**Apply - L3**)
- CO4:** Develop structured data processing tools (**Apply- L3**)
- CO5:** Develop semi/unstructured data processing tools (**Apply – L3**)

Unit I:

Introduction to Big data: Types of Digital Data, Classification of Digital Data, Characteristics of Data, Evolution of Big Data, Definition of Big Data, Challenges with Big Data, What is Big Data?, Other Characteristics of Data Which are not Definitional Traits of Big Data, Why Big Data?, analyzing Data with Unix tools, Best Practices of Big Data Analytics, Analyzing Data with Hadoop, Hadoop Streaming, Hadoop Echo System.

Unit II:

Hadoop Distributed File System: The Design of HDFS, HDFS Concepts, Command Line Interface, Hadoop file system interfaces, Data flow, Data Ingestion with Sqoop and Hadoop archives, Hadoop I/O: Compression, Serialization.

Unit III:

MapReduce Technique: How MapReduce works?, Anatomy of a Map Reduce Job Run, Failures, Shuffle and Sort, Task Execution, Map Reduce Types and Formats, Map Reduce Features.

Unit IV:

Structured Data Processing Tools: Hive: Installation, Running Hive, HiveQL, Tables, Querying Data, User Defined functions Sqoop: Introduction, generate code, Database import, working with imported data, importing large objects, performing an exports.

Unit V:

Semi-structured and unstructured Data Processing Tools: Introduction to Pig, Execution Modes of Pig, Comparison of Pig with Databases, Grunt, Pig Latin, User Defined Functions, Data Processing operators.
HBase: Basics, Concepts, Clients, Example, HBase versus RDBMS.

Text Books:

1. Tom White "Hadoop: The Definitive Guide" Third Edit, O'reily Media, 2012.
2. Big Data and Analytics, 2ed Seema Acharya, Subhashini Chellappan, Wiley 2015.

Reference Books:

1. Michael Berthold, David J. Hand, "Intelligent Data Analysis", Springer, 2007.
2. Jay Liebowitz, "Big Data and Business Analytics" Auerbach Publications, CRC press (2013)
3. Tom Plunkett, Mark Hornick, "Using R to Unlock the Value of Big Data: Big Data Analytics with Oracle R Enterprise and Oracle R Connector for Hadoop", McGraw-Hill/Osborne Media (2013), Oracle press.

Pre-requisite: Database Management Systems, Object Oriented Programming

Course Objectives:

To expose the students to the following:

1. Basics of database management system.
2. How a real-world problem can be mapped to schemas.
3. Solve different industry level problems.

Course Outcomes (CO): At the end of this course, the student will be able to:

CO1: Implement SQL queries using DDL/DML/DCL/TCL commands. **(Apply-L3)**

CO2: Apply different Integrity constraints & Pattern Matching Techniques for effective database design. **(Apply-L3)**

CO3: Implement the GROUP BY, HAVING clauses, Joins, Views and Sequences. **(Apply-L3)**

CO4: Improve individual / teamwork skills, communication & report writing skills with ethical values.

1. Queries on DDL commands (Create, Alter, Drop, Rename)

Task 1:

- a. Create a table with the following schema

Student (sid, sname, saddress, sphone)

- b. Write a query to display structure of the table as student (sid, sname, saddress, sphone, smail)
- c. Write a query to display structure of the table as student (sid, sname, sphone, smail)
- d. Write a query to change the name of the column smail to smailid
- e. Drop the table student

Task 2:

- a. Create a table with the following schema

Faculty (Fid, Fname, Faddress, Fbranch)

- b. Write a query to display structure of the table as faculty (Fid, Fname, Faddress, Fbranch, Fphone)
- c. Write a query to display structure of the table as faculty (Fid, Fname, Fbranch)
- d. Drop the table faculty

2. Queries on DML commands (Insert, update, Delete, select)

- a. Create a table with the following schema

Storeinfo (storename, sales, txn_date, storeaddress)

- b. Insert 10 rows in to the table
- c. Insert 15 rows into the table using single insert statement
- d. Write a query to change the sales of Levis store from 20% to 30%
- e. Write a query to change the address of VanHeusen store to 40-32, Himayat Nagar, Hyderabad.
- f. Write a query to delete the details of Levis store.
- g. Write a query to delete the details of a store with address 20/35, Ameerpet
- h. Write a query to display all the details of storeinfo.
- i. Write a query to retrieve the details of Levis store.

3. Queries on DCL, TCL commands and computations on queries

DCL (Grant, Revoke)

- a. Write queries to create a role called as testing and create permission to testing
- b. Write queries to revoke a create table privilege from testing role and drop the testing role.

TCL (Commit, Rollback, Savepoint)

- a. Perform commit operation to save the changes permanently
- b. Create a schema student (sid, sname) with 3 rows initially. Later insert 3 rows with 3 save points.
- c. perform rollback to savepoint B, savepoint A.

4. Commands on key-constraints (NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY, CHECK)

- a. Create a table persons (id, fname, lname, age) where the null values should not be allowed while inserting the rows in id column.
- b. Create a table persons such that all the values in column id are different.
- c. Create a table student such that the column student id should not allow null values and duplicate values
- d. Create a table persons (pid, fname, lname, age) and orders (id, ordernumber, pid).
- e. Write a query to retrieve the ordered, ordernumber of a person with id 3.
- f. Write a query to retrieve age of a person whose order id is 3.
- g. Create a table votedetails (voterid, name, age, address) which does not allow the details of people whose age is <18

5. Pattern matching queries and SQL queries using oracle functions

- a. Write a query (WAQ) to retrieve the names of students whose names start with S.
- b. WAQ to retrieve the names of the students whose names end with a.
- c. WAQ to retrieve the names of the students whose names start with ca.
- d. WAQ a query to retrieve the names of the students whose names consists of ee.
- f. WAQ to concatenate 2 strings
- g. WAQ to set the first character in uppercase and rest in lower case
- h. WAQ to display the location of DER in Hyderabad.
- i. WAQ to return the length of a string engineering.
- j. WAQ to convert all letters in a string HYDERABAD to lower case and upper case also
- k. WAQ to add NEW to the word HYDERABAD and NAWABS to Hyderabad.
- l. WAQ to extract base from Data base Management systems
- m. WAQ to count number of students in student table
- n. Write queries to demonstrate CEIL, FLOOR, GREATEST, LEAST, MAX, MIN, SUM.
- o. Write queries to demonstrate date functions like ADD_MONTHS, CURRENT_DATE, LAST_DAY, MONTHS_BETWEEN, NEXT_DAY, ROUND, SYSDATE, SYSTIMESTAMP.

5. Implementing Group By, Having, Order by clause

Task 1:

- a. Create a table Northzone (custid, custname, address, city, country)
WAQ to display the number of customers in each country.
- b. WAQ to display the number of customers in each country in descending order
- c. Create a table Northzone (custid, custname, age, address, city, country)
WAQ to display the number of customers in each country whose age is greater than 30.

Task 2:

a. For the schema student

WAQ to display number of students in each section

b. WAQ to display names of students in descending order

c. WAQ to display the names of students in each section whose percentage is >65.

6. Queries on joins (INNER, LEFT OUTER, RIGHT OUTER, FULL JOINS)

a. Create a table orders (orderid, custid, orderdate)

b. Create a table customers (custid, cname, country)

perform all join operations on the given two tables based on conditions.

Sub-Queries

a. WAQ to display list of children taller than 'Mike' from height table.

b. WAQ to display the names of children who are taller and older than 'Jim'

c. WAQ to get the names of the employees who work in department with the highest budget.

d. WAQ to display the names of students whose percentage is greater than 65

e. WAQ to display the names of the employees whose salary is greater than the average of all salaries.

f. WAQ to demonstrate ALL, ANY.

7. Operations on views (Insert, Update, Delete), sequences

a. Create a students above 65 with the details sid, sname, sphone, smailid

b. WAQ to change the phone number of a student whose name is 'XYZ' in students above 65 view.

c. WAQ to drop students above 65 view.

d. WAQ to delete the students details whose sid is 123 from students above 65 view.

8. Sequences

a. Create a sequence seq1 for a table class which starts with 1 increment by 1 and max value is 999 with cycle and without cycle.

b. Insert the values into class using nextval.

TEXT BOOKS/SUGGESTED READING:

1. Oracle: The Complete Reference by Oracle Press

2. Nilesh Shah, "Database Systems Using Oracle", PHI, 2007

3. Rick F Vander Lans, "Introduction to SQL", Fourth Edition, Pearson Education, 2007

Pre Requisite: NIL

Course Outcomes (COs):

CO1: Develop and style web pages using HTML and CSS to create responsive and visually appealing interfaces. **(Apply-L3)**

CO2: Implement JavaScript functionalities—including DOM manipulation and event handling—to create dynamic, interactive web applications. **(Apply-L3)**

CO3: Build backend applications using Node.js, Express.js, and MongoDB and integrate them with frontend solutions for full-stack development. **(Apply-L3)**

CO4: Maintain detailed and organized lab records that document the applied methodologies and outcomes in each experiment. **(Apply-L3)**

Experiment 1: Basic Portfolio Webpage Development

Objective: Apply HTML to create a structured webpage that represents a personal portfolio.

Tasks:

- Develop a multi-section webpage including Home, About, Skills, and Contact sections.
- Use semantic HTML tags to structure content.
- Ensure clarity in the document hierarchy.
- **Outcome:** A fully structured webpage that serves as a personal portfolio.

Experiment 2: Responsive Landing Page with CSS Flexbox and Grid

- **Objective:** Apply CSS techniques to design a responsive landing page that adapts to different screen sizes.
- **Tasks:**
 - Design a landing page layout using CSS Flexbox and Grid properties.
 - Implement media queries to adjust layout for mobile and desktop views.
 - Experiment with colour schemes, typography, and spacing to enhance aesthetics.
- **Outcome:** A visually appealing landing page that works seamlessly on various devices.

Experiment 3: Registration Form with HTML5 and CSS

- **Objective:** Apply form-building techniques and styling to create a user-friendly registration form.
- **Tasks:**
 - Build a form with fields such as name, email, password, etc., using HTML5
 - Style the form using CSS to enhance usability and design consistency.
 - Apply basic form validation (e.g., required fields) using HTML5 attributes.
- **Outcome:** A registration form that demonstrates both structural and visual design skills.

JavaScript & DOM Manipulation

Experiment 4: JavaScript Calculator Application

- **Objective:** Apply JavaScript to build a functional calculator that performs arithmetic operations.
- **Tasks:**
 - Create functions for addition, subtraction, multiplication, and division.
 - Use event listeners to capture user input from buttons.
 - Dynamically update the display based on calculation results.
- **Outcome:** A working calculator that demonstrates core JavaScript logic and DOM manipulation.

Experiment 5: To-Do List Application

- **Objective:** Apply JavaScript and DOM techniques to manage dynamic content on a webpage.
- **Tasks:**
 - Develop an interface to add new tasks, delete tasks, and mark tasks as complete.
 - Use JavaScript arrays and objects to store and manage tasks.
 - Update the DOM in response to user interactions to reflect the current state of the to-do list.
- **Outcome:** An interactive to-do list that responds to user actions in real time.

Experiment 6: Dynamic Image Slider

- **Objective:** Apply JavaScript functions and DOM manipulation to create a dynamic image slider.
- **Tasks:**
 - Develop a slider that cycles through images automatically and via user control (next/previous buttons).
 - Use JavaScript timers and event listeners to control the slider's behavior.
 - Manipulate the DOM to show/hide images as needed.
- **Outcome:** A fully functional image slider that demonstrates applied dynamic content control.

Experiment 7: Setting Up a Basic Node.js & Express.js Server

- **Objective:** Apply server-side development skills by setting up a basic server that serves static content.
- **Tasks:**
 - Install and configure Node.js and Express.js.
 - Create a simple server that responds with an HTML page.
 - Test the server using local development tools.
- **Outcome:** A basic server setup that successfully serves a static webpage.

Experiment 8: REST API Development Using Express.js

- **Objective:** Apply API development skills by creating RESTful endpoints for basic CRUD operations.
- **Tasks:**
 - Design and implement API routes for Create, Read, Update, and Delete
 - Use middleware to handle requests and responses.
 - Test API endpoints using tools like Postman or similar.
- **Outcome:** A set of RESTful API endpoints that can be consumed by frontend applications.

Experiment 9: MongoDB Integration with Node.js Using Mongoose

- **Objective:** Apply database integration techniques by connecting a MongoDB database with a Node.js application.
- **Tasks:**
 - Install and configure MongoDB along with the Mongoose ODM.
 - Design a simple data schema and implement CRUD operations.
 - Test database operations through the API endpoints developed earlier.
- **Outcome:** A Node.js application that successfully integrates with MongoDB, demonstrating applied database management.

Full-Stack Integration**Experiment 10: Building a Login and Registration System**

- **Objective:** Apply full-stack development by integrating frontend and backend for user authentication.
- **Tasks:**
 - Develop backend routes for user login and registration using Node.js, Express.js, and MongoDB.
 - Apply data validation and error handling techniques.
 - Test the authentication process through API endpoints.
- **Outcome:** A secure login and registration system that demonstrates applied authentication techniques.

Experiment 11: Developing a React Frontend for Data Fetching

- **Objective:** Apply React.js to build a frontend interface that interacts with an Express.js API.
- **Tasks:**
 - Set up a React application and create components for displaying data.
 - Use Axios or Fetch API to retrieve data from backend endpoints.
 - Manage state using React's useState and useEffect hooks to reflect fetched data.
- **Outcome:** A responsive React application that effectively displays data from the backend.

Experiment 12: State Management in React using Context API

- **Objective:** Apply React's Context API to manage state across multiple components in a React application.
- **Tasks:**
 - Create a React application with multiple components (e.g., a Navbar, a User Profile, and a Settings panel).

- Implement Context API to store and share user data (e.g., username, theme preference) across components.
- Use the Context Provider to wrap necessary components and pass state values.
- Access the shared state using useContext() inside child components.
- Modify state values through user interactions, such as changing a theme or updating a user name dynamically.
- Test the application by ensuring that changes in one component reflect in others without passing props manually.
- Outcome: A small React app that demonstrates state sharing across components using the Context API instead of prop drilling.

Final Project Integration

- Project Description: The final integrated project (Experiment 12) is designed as a capstone where students combine all previous experiments. They will build a full-stack web application that includes:
 - A front-end developed in React for user interaction and data presentation.
 - A back-end server using Node.js and Express.js to manage business logic and API endpoints.
 - A database layer using MongoDB (with Mongoose) to store user and application data.
 - User authentication and dynamic UI features such as interactive forms and live data updates.
 - State Management in React using Context

Text Books:

1. "Full Stack Development with MongoDB, Express, React, and Node"by Shama Hoque- Packt Publishing
2. "Learning React: Modern Patterns for Developing React Apps"by Alex Banks, Eve Porcello ---O'Reilly Media

Reference Books

1. "Practical Node.js: Building Real-World Scalable Web Apps" by Azat – Apress Publications
2. "MongoDB: The Definitive Guide" by Shannon Bradshaw, Kristina Chodorow - O'Reilly Media

B.Tech.(IT)
Minor

23ITM8- DATA ANALYTICS LAB

L	T	P	Cr.
3	0	0	3

Pre-requisite: Database Management Systems, Object Oriented Programming

Course Educational Objective: The Objective of the course is to provide practical, foundation level training that enables immediate and effective participation in Big Data and other Analytics projects using Hadoop and Data Visualization using Tableau.

Course Outcomes (CO): At the end of this course, the student will be able to:

- CO1:** Demonstrate the installation of Bigdata analytic tools. (**Understand – L2**)
- CO2:** Apply data modelling techniques to large data sets. (**Apply – L3**)
- CO3:** Conduct exploratory data analysis using visualization. (**Understand – L2**)
- CO 4:** Improve individual / teamwork skills, communication & report writing skills with ethical values.

List of Experiments:

1. Refreshing Linux Commands and Installation of Hadoop
2. Implementation of basic Word Count Map Reduce program
3. Implementation of Matrix Multiplication with Hadoop Map Reduce.
4. Implementation of Weather mining by taking weather data set using Map Reduce.
5. Installation of Hive along with practice examples.
6. Downloading and installing Tableau Understanding about importing data, saving, opening, and sharing workbooks.
7. Data Preparation with Tableau
8. Charts: Bar Charts, Legends, Filters, and Hierarchies, Step Charts, Line Charts.
9. Maps: Symbol Maps, Filled Maps, Density Maps, Maps with Pie Charts
10. Interactive Dashboards

Text Books:

1. Seema Acharya, Subhasini Chellappan, "**Big Data Analytics**" Wiley 2015
2. Alexander Loth, "**Visual Analytics with Tableau**", ISBN: 978-1-119-56020-3, Wiley 2019

Reference Books:

1. Tom White, "**Hadoop: The Definitive Guide**", Third Edit on, O'reily Media, 2012
2. Michael Berthold, David J. Hand, "**Intelligent Data Analysis**" Springer, 2007.
3. Jay Liebowitz, "**Big Data and Business Analytics**" Auerbach Publications, CRC press (2013).
4. AnandRajaraman and Jeffrey David Ulman, "**Mining of Massive Datasets**", Cambridge University Press, 2012.
5. ArvindSathi,. "**BigDataAnalytics: Disruptive Technologies for Changing the Game**", MC Press, 2012, 2001.

E-Resources:

1. <http://nptel.ac.in/courses/106104135/48>
2. <http://nptel.ac.in/courses/110106064/>