SEMESTER-VI

COURSE 15 A: MERN STACK

Theory	**Credits: 3**	**3 hrs/week**

Course Objectives:

1. Understand full-stack architecture and the individual roles of the MERN components-MongoDB, Express.js, React.js, and Node.js.
2. Develop interactive front-end applications using React.js and master state management and routing techniques.
3. Model and manipulate NoSQL databases using MongoDB and Mongoose, including CRUD operations and schema validations.
4. Build RESTful APIs using Express.js and integrate server-side logic with frontend and database components.
5. Implement full-stack application features such as authentication, session management, and deployment using modern platforms.

Course Outcomes:

1. Explain the architecture of the MERN stack and configure a Node.js development environment with essential modules and server setup.
2. Develop dynamic user interfaces using React functional components, hooks, forms, and routing for seamless user experiences.
3. Perform database operations using MongoDB and Mongoose, including schema design, data validation, and relational mapping.
4. Construct RESTful backend services using Express.js with routing, middleware, and error handling mechanisms.
5. Integrate frontend and backend components with secure data flow, apply JWT-based authentication, and deploy applications on platforms like Render, Netlify, or Vercel.

Unit 1. Introduction to MERN Stack & Node.js:
Introduction to Full Stack Web Development, Frontend vs Backend, What is the MERN stack? Architecture of MERN Applications

Introduction to Node.js, Installing Node.js & npm, Node.js fundamentals (Modules, Events, Streams), Asynchronous Programming & Event Loop, Node.js File System module, npm modules & custom modules, Setting up a basic server with Node.js

Unit 2. React.js (Frontend Framework):
Introduction to React, Functional Components & JSX, State & Props, Handling Events, useState, useEffect Hooks, Conditional Rendering & Lists, React Router (Routing), Forms in React (Controlled Components), Axios for HTTP requests

Unit 3. MongoDB with Mongoose:
Introduction to NoSQL Databases, MongoDB vs SQL Databases, Installing & using MongoDB (local & Atlas), CRUD Operations with MongoDB Shell & Compass, Mongoose ODM, Models, Schemas, Validation, Relationships (One-to-Many, Many-to-Many), Aggregation

Unit 4. Express.js (Backend Framework):
Introduction to Express.js, Creating RESTful APIs using Express, Routing (GET, POST, PUT, DELETE), Middleware in Express, Error Handling, Connecting to MongoDB using Mongoose, Environment variables and `.env` files

Unit 5. Full Stack Integration & Deployment:
Connecting Frontend (React) with Backend (Express), CORS and Proxy setup, Authentication with JWT, Protected Routes in Frontend, Role-based access control, Deployment: Deploying backend to Render/Heroku, Deploying frontend to Netlify/Vercel, Connecting MongoDB Atlas

Textbooks:

1. Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node, Subramanian, Vasan, 2nd Edition, Apress,
2. Learning React: Functional Web Development with React and Redux, Alex Banks & Eve Porcello, O'Reilly
3. MongoDB: The Definitive Guide, 3rd Edition, Shannon Bradshaw, Eoin Brazil, Kristina Chodorow, O'Reilly

Reference Books:

1. Ultimate Full-Stack Web Development with MERN, Nabendu Biswas, Orange Education Pvt Ltd.
2. Full Stack Development with MERN, Thompson Carter, Lincoln Publishers

Activities:

Outcome: Explain the architecture of the MERN stack and configure a Node.js development environment with essential modules and server setup.

Activity: Set up a basic Node.js project with Express and required modules (express, nodemon, dotenv). Create a simple server that responds with Hello MERN Stack on a browser.

Evaluation Method: Checklist-based review (10-point scale):

- Correct installation of Node.js and modules
- Functional server response
- Use of environment variables
- Folder structure and code clarity

Outcome: Develop dynamic user interfaces using React functional components, hooks, forms, and routing for seamless user experiences.

Activity: Build a multi-page React app (e.g., user profile and contact form) using:

- Functional components
- useState and useEffect hooks
- Controlled form inputs

- React Router for navigation

Evaluation Method: Live demo and rubric (10-point scale):

- Component structure and reusability
- Hook usage and state management
- Form validation and routing functionality
- UI responsiveness and layout

Outcome: Perform database operations using MongoDB and Mongoose, including schema design, data validation, and relational mapping.

Activity: Create a student database using MongoDB and Mongoose. Implement:

- Schema with validation rules
- CRUD operations (Create, Read, Update, Delete)
- Reference between collections (e.g., student and course)

Evaluation Method: Code walkthrough and test cases (10-point scale):

- Schema correctness and validation
- CRUD functionality
- Relational mapping using ref
- Console output and error handling

Outcome: Construct RESTful backend services using Express.js with routing, middleware, and error handling mechanisms.

Activity: Develop a REST API for a task manager app using Express.js. Include:

- Routes for task operations
- Middleware for logging and JSON parsing
- Error handling for invalid routes

Evaluation Method: API testing with Postman(10-point scale):

- Route functionality and status codes
- Middleware implementation
- Error response structure
- Code readability and modularity

Outcome: Integrate frontend and backend components with secure data flow, apply JWT-based authentication, and deploy applications on platforms like Render, Netlify, or Vercel.

Activity: Build a login system with:

- JWT-based authentication
- Protected routes
- Frontend-backend integration
- Deploy frontend on Netlify/Vercel and backend on Render

Evaluation Method: Deployment demo and checklist (10-point scale):

- Token generation and validation
- Secure data flow between client and server
- Working login/logout flow
- Successful deployment and accessibility

SEMESTER-VI

COURSE 15 A: MERN STACK

Practical **Credits: 1** **2 hrs/week**

List of Experiments:

1. Install Node.js and npm; Create and Use Built-in & Custom Node.js Modules
2. Demonstrate Asynchronous Programming Using Callbacks and Promises
3. Implement File Read/Write Operations Using Node.js File System Module
4. Create a React App and Build Functional Components Using JSX
5. Handle Events and Use useState & useEffect Hooks in React
6. Implement Navigation Between Pages Using React Router
7. Create and Validate a Form Using Controlled Components in React
8. Install and Connect to MongoDB (Local and MongoDB Atlas)
9. Perform CRUD Operations Using MongoDB Shell and MongoDB Compass
10. Create Mongoose Schemas and Models, and Connect Them to a Node.js App
11. Create a RESTful API Using Express.js (GET, POST, PUT, DELETE)
12. Use Middleware and Error Handling in an Express App
13. Connect Express App to MongoDB Using Mongoose
14. Implement User Authentication Using JWT in Express and React
15. Create Protected Routes and Role-Based Access Control in React
16. Deploy Backend to Render/Heroku and Frontend to Netlify/Vercel