MERN Stack Development Course Notes

Instructor: Mohammad Asad
Presented By: Skills Sikhao Education Academy

Introduction

This course is designed to teach students how to build fullstack web applications using the **MERN Stack**, which includes **MongoDB**, **Express.js**, **React.js**, and **Node.js**. The course starts with modern JavaScript foundations, moves into frontend development using React, covers backend development using Node.js and Express, and finally teaches database integration and deployment techniques.

By the end of the course, students will be able to build and deploy complete web applications using real world tools and technologies.

Course Overview

The course is divided into the following sections:

1. Modern JavaScript for Web Development
2. React.js – Frontend Development
3. Node.js and Express.js – Backend Development
4. MongoDB – NoSQL Database Integration
5. Fullstack Integration – Connecting MERN
6. Deployment – Hosting MERN Applications

---

1. Modern JavaScript for Web Development

Before Exploring React and Node.js, it is crucial to understand advanced JavaScript concepts.

Topics Covered:

- **Execution Context & Call Stack**
  Understand how JavaScript code runs behind the scenes.

- **Closures & Lexical Scope**
  Learn how functions preserve data even after execution.

- **Asynchronous JavaScript**
    - Callbacks
    - Promises
    - Async/Await
      Useful for API calls and server operations.
- **ES6+ Features**
    - Arrow Functions
    - Template Literals
    - Destructuring
    - Spread/Rest Operators
    - Object Enhancements
- **Module Systems**
    - CommonJS (require/module.exports)
    - ESModules (import/export)
- **Error Handling**
  `try/catch`, custom error messages, throwing errors.

---

2. React.js – Frontend Development

React.js is used for building fast, responsive user interfaces. This section covers React from beginner to advanced level.

Topics Covered:

- **What is React?**
  Learn how React manages UI and why it’s so widely used.
- **JSX (JavaScript XML)**
  Write HTML-like syntax inside JavaScript.
- **Functional Components**
  Build UI using functions, not classes.
- **Props and State**
    - Props: Pass data to components
    - State: Store component data
- **Event Handling**
  Handle click events, form inputs, and custom functions.
- **React Hooks**
    - `useState`: Manage state
    - `useEffect`: Run side-effects
    - `useRef`: Access DOM elements

- **React Router DOM**
    - Single Page Application routing
    - Route Parameters
    - Nested Routes
- **Forms and Validations**
  Controlled components and input validation.
- **API Integration**
  Use `fetch` or `axios` to get data from backend.
- **Custom Hooks and Reusability**
  Write custom hooks to organize logic.
- **Best Practices**
  Folder structure, naming conventions, component reusability.
- **React Project (Frontend Only)**
  Build a To-Do App or Weather App.

---

3. Node.js and Express.js – Backend Development

Node.js and Express.js are used to build the backend APIs and server logic for web apps.

Topics Covered:

- **What is Node.js?**
  A runtime environment to run JavaScript on the server.
- **Installing Node and NPM**
  Setup environment and install required packages.
- **Creating a Basic Server**
  Using Node's `http` module and Express.
- **Express.js Framework**
    - Setting up server
    - Creating routes (GET, POST, PUT, DELETE)
- **Middleware**
    - Logging
    - Request parsing
    - Custom middleware functions

- **RESTful API Design**
  Design endpoints to perform Create, Read, Update, Delete (CRUD) operations.
- **Request and Response Handling**
  Understanding `req`, `res`, and status codes.
- **Error Handling in Express**
  Global error middleware and custom error responses.
- **Using Environment Variables**
  Use `.env` files for storing secret keys and configurations.
- **Express Project (Backend Only)**
  Build a simple Blog API or Product API.

---

4. MongoDB – NoSQL Database Integration

MongoDB is a document-based database used to store and manage data in real-time applications.

Topics Covered:

- **What is MongoDB?**
  A NoSQL database that stores data as JSON-like documents.
- **MongoDB Atlas (Cloud Database)**
  - Creating an account
  - Setting up a cluster
  - Getting connection strings
- **Mongoose Library**
  A tool to work with MongoDB using Node.js.
- **Schemas and Models**
  Define structure and constraints for your data.
- **CRUD with Mongoose**
  - Create documents
  - Read documents
  - Update documents
  - Delete documents
- **Validation and Error Handling**
  Use Mongoose validation to ensure clean data.
- **Populating Relationships**
  Connect and fetch data from related collections.

- **MongoDB Project (Backend + Database)**
  Build Notes App with full MongoDB integration.

---

5. Fullstack Integration – Connecting MERN

In this section, the frontend (React) and backend (Express + MongoDB) are integrated into one application.

Topics Covered:

- **Connecting React to Backend API**
  Use `axios` to send requests from React to Node.js.
- **CORS Setup**
  Enable cross-origin access for frontend-backend communication.
- **Authentication and Authorization (Bonus)**
  - JWT (JSON Web Tokens)
  - Password hashing with bcrypt
  - Protected routes
- **React Context or Redux (Optional)**
  Manage global state for user sessions and data.
- **Fullstack Project Examples**
  - Blog App
  - Notes Manager
  - Task Tracker
  - E-commerce CRUD App

---

6. Deployment – Hosting MERN Applications

Learn how to take your fullstack app live for the world to use.

Topics Covered:

- **Git and GitHub Basics**
  Push your code to remote repositories.
- **Hosting Backend on Railway**
  - Create account on Railway
  - Connect GitHub
  - Set environment variables
  - Deploy your Express app

- **Hosting Frontend on Vercel**
    - Create Vercel account
    - Import from GitHub
    - Build and deploy React app
- **Environment Variables in Production**
  Manage `.env` for both backend and frontend securely.
- **Testing Live Project**
  Ensure that both frontend and backend are working correctly on public URLs.

Tools and Technologies

- **Frontend**: React.js, JSX, Axios
- **Backend**: Node.js, Express.js
- **Database**: MongoDB, Mongoose
- **Deployment**: Vercel (Frontend), Railway (Backend)
- **Utilities**: Postman, Git, GitHub, dotenv, VS Code

Final Outcome

After completing this course, students will be able to:

- Build fullstack applications using modern JavaScript tools
- Create and connect RESTful APIs with databases
- Manage state and user interaction in React
- Store and fetch data securely using MongoDB
- Deploy full applications online using free platforms
- Apply best practices in real-world development

Final Words for Students

The topics explained above are just a roadmap. A foundation to help you understand what this course covers.
But in reality, this course is **much more than just a list of technologies**.

You're not just going to **"learn React"** or **"build an API."**
You're going to **build real applications**, **solve real problems**, and write **production-ready code**.

We will:

- Create multiple hands on **projects** together
- Explore **real-world development workflows**
- Discuss how things are used in **actual jobs and freelance projects**
- Learn how to **debug, deploy, and scale** your applications
- And most importantly, **have fun while learning**

By the end of this course, you won’t just know the **MERN stack** —
You’ll be able to **think like a developer**, **build like a professional**, and **launch like a founder**.

So get ready because this journey is going to be full of **learning, building, and growing**.