MERN Stack Development with AI Integration Syllabus

Building end-to-end applications with MongoDB, Express.js, React and Node.js, plus practical AI. 15 modules, 8 AI topics and a mini project.

Course Overview

This course teaches you to build complete web applications using the MERN stack: MongoDB (database), Express.js and Node.js (backend) and React (frontend), all in JavaScript. It has 15 modules plus 8 AI topics, ending in a mini project.

Why learn MERN Stack?

One language (JavaScript) across frontend, backend and database. Fast development of REST APIs and single-page applications. Real workflow: authentication, testing, version control and deployment. Project-based learning that connects every module into complete applications.

Why add AI to MERN applications?

Add chat, summarisation, document processing and intelligent search. Call OpenAI / Gemini APIs from Node.js and Express. Learn secure API handling, validation and error management.

Course Roadmap

Stage 1 - Frontend Foundations: HTML, CSS, Bootstrap, JavaScript (Modules 1-2)

Stage 2 - React.js: components, hooks, routing, state management, API calls (Modules 3-5)

Stage 3 - Node.js and Express.js: server, REST APIs, middleware (Modules 6-7)

Stage 4 - MongoDB and Mongoose: CRUD, aggregation, schema design (Modules 8-10)

Stage 5 - Security and Quality: JWT auth, testing (Modules 11-12)

Stage 6 - Tools and Deployment: Git, GitHub, Docker, CI/CD, cloud (Modules 13-15)

AI - AI Integration: OpenAI / Gemini, chatbot, semantic search, mini project (AI Topics 1-8)

Part 1 - Frontend Development (Modules 1-5)

1. HTML, CSS and Bootstrap

HTML5: structure, semantic tags, forms, tables, lists, links and multimedia, CSS3: selectors, box model, Flexbox, Grid, positioning and media queries, CSS animations: transform, translate and keyframes, Bootstrap components, grid system, utilities and breakpoints

LAB: Build a responsive landing page

2. JavaScript (ES6+)

Variables, data types, operators and control flow, Functions, arrow functions, scope, closures and this, Arrays, objects, destructuring, spread/rest and array methods (map, filter, reduce), DOM manipulation and event handling, Asynchronous programming: callbacks, promises, async/await and the Fetch API, ES modules, error handling and JSON

LAB: Build a To-Do app with local storage

3. React Fundamentals

React architecture, virtual DOM, Vite / Create React App setup and project structure, JSX, functional components, props and lists/keys, State, event handling, conditional rendering and controlled forms, Hooks: useState and useEffect, component lifecycle with hooks, Styling: CSS modules, Bootstrap / Tailwind CSS

Greens Technologies

4. Advanced React

Hooks: useRef, useContext, useReducer, useMemo, useCallback and custom hooks, React Router: routes, nested routes, parameters, protected routes and navigation, Forms and validation (React Hook Form / Formik), API integration with Axios / Fetch, loading, success and error states, Performance: lazy loading, code splitting and memoization

5. State Management

Props drilling vs Context API, Redux Toolkit: store, slices, actions, reducers and async thunks, Introduction to React Query for server state

LAB: Build an e-commerce frontend with product list, cart and search

Frontend hands-on practice

Create a responsive page, a validated form and reusable React components. Fetch data from a REST API and display loading, success and error states.

Suggested projects: BMI calculator, To-Do app, registration form validation, e-commerce interface, weather dashboard.

Outcome: students can create responsive React interfaces and connect them to backend APIs.

Part 2 - Backend: Node.js, Express.js and MongoDB (Modules 6-10)

6. Node.js

Node.js architecture, V8 engine, event loop and non-blocking I/O, Modules: CommonJS and ES modules, npm and package.json, Core modules: fs, path, os, http and events, Streams, buffers and environment variables (dotenv), Asynchronous patterns: callbacks, promises and async/await

LAB: Build a file-handling and basic HTTP server program

7. Express.js and REST APIs

Express setup, routing, route parameters and query strings, Middleware: built-in, third-party and custom, REST principles, HTTP methods, status codes and API structure (routes, controllers, services), Request validation, error-handling middleware and file uploads (Multer), CORS, API documentation (Swagger) and testing with Postman

LAB: Build a CRUD REST API

8. MongoDB Fundamentals

NoSQL concepts, SQL vs NoSQL, documents, collections and BSON, Installation, MongoDB Atlas and Compass, CRUD: insertOne/Many, find, updateOne/Many, deleteOne/Many, Query and update operators: $gt, $in, $and, $or, $set, $inc, $push, $pull, Projection, sorting, limit and skip (pagination)

9. Aggregation, Indexing and Schema Design

Aggregation pipeline: $match, $group, $project, $sort, $lookup and $unwind, Indexes: single, compound and text, performance with explain(), Schema design: embedding vs referencing, one-to-many and many-to-many, Transactions and backup basics

10. Mongoose ODM

Connecting Express to MongoDB with Mongoose, Schemas, models, data types and validation, CRUD with Mongoose, populate, virtuals and middleware (hooks)

LAB: Connect the REST API to MongoDB with full CRUD

Greens Technologies

Part 3 - Security, Testing and Deployment (Modules 11-15)

11. Authentication and Security

Password hashing with bcrypt, JWT authentication and refresh tokens, protecting routes with middleware, Role-based access control (RBAC), React integration: login flow, protected routes, Axios interceptors and token storage, Security basics: Helmet, rate limiting, input sanitisation, CORS and OWASP top risks

12. Testing

Unit and component testing in React with Jest and React Testing Library, API testing in Node.js with Jest and Supertest, Mocking services and databases

LAB: Write tests for a controller and a React component

13. Git, GitHub and Workflow

Git basics: repository, commit, branch, merge and conflict resolution, Workflow models: feature branch and Gitflow, pull requests and code review, GitHub basics and differences between Git and GitHub

14. Docker and CI/CD Basics

Docker basics: images, containers and docker-compose for the MERN stack, CI/CD introduction with GitHub Actions

15. Deployment

Production builds for React and Node.js, environment configuration, Deploying frontend, backend and MongoDB Atlas to a cloud platform, Process management with PM2 and reverse proxy basics (Nginx)

Part 4 - AI Integration (AI Topics 1-8)

Add real AI capabilities to Node.js / Express applications using OpenAI and Gemini APIs.

AI Topic 1 - Introduction to AI Integration

What AI integration means, prompts, tokens, models and responses, Business use cases: customer support, document workflows and knowledge search, Application flow: React -> Express API -> AI provider -> response, Provider selection: capabilities, pricing, limits and privacy

LAB: Make a first AI API request from Node.js

AI Topic 2 - OpenAI and Gemini APIs with Node.js

API keys, endpoints, request payloads and response formats, Calling APIs with fetch / axios or the official SDKs, Handling errors, timeouts and rate limits, keeping keys in environment variables

LAB: Compare OpenAI and Gemini responses for the same prompt

AI Topic 3 - Express Endpoints for AI Features

Service layer for AI requests and REST endpoints that expose them, Request validation, streaming responses and error handling, Connecting a React front end to the AI endpoint

LAB: Build an endpoint that returns an AI-generated response

AI Topic 4 - AI Chatbot

Chat request/response flow, system prompts and conversation context, Storing conversation history in MongoDB, React chat UI with loading states and failure handling

LAB: Build a customer-support chatbot

Greens Technologies

AI Topic 5 - Text Generation and Summarisation

Prompt basics: instructions, context, examples and output format, Draft emails and descriptions, summarise, rewrite, translate and correct text, Validating generated content

LAB: Build a text summarisation endpoint

AI Topic 6 - Document Ingestion and Indexing

Extract text from PDF, DOCX and HTML sources, Split content into chunks, generate embeddings, retain source text and metadata, Store vectors in MongoDB Atlas Vector Search or another vector database

LAB: Build an ingestion pipeline that extracts, chunks, embeds and indexes documents

AI Topic 7 - Semantic Search and Retrieval

Convert a query to an embedding and search by similarity (cosine), Retrieve relevant chunks, augment the prompt and generate a grounded answer

LAB: Build a semantic-search endpoint over indexed documents

AI Topic 8 - Securing AI Endpoints

Authentication, authorization and usage limits for AI routes, Protecting API keys, handling quota errors and invalid responses safely, Avoiding sensitive data in prompts, reviewing AI outputs

LAB: Secure an AI endpoint and test success, invalid input and provider failure

Mini Project: AI-Powered Customer Support Portal

Users submit questions and receive AI-generated suggested answers.

Frontend: React support request and chat interface

Backend: Node.js / Express REST API with routes, controllers and services

Database: MongoDB (Mongoose) for requests, users and chat history

AI: OpenAI or Gemini through a backend service, with document search

Security: JWT login, role-based access and protected API keys

Testing and delivery: unit and API tests, then GitHub and cloud deployment

Learning outcomes

Build responsive React applications with hooks, routing and state management. Develop REST APIs with Node.js and Express.js. Model and query data with MongoDB and Mongoose. Apply authentication, security, testing and Git-based practices. Integrate AI chat, summarisation, document ingestion and semantic search. Deploy and demonstrate a full-stack AI-enabled application.

Program Details - Tools and Technologies

Frontend: HTML5, CSS3, Bootstrap, JavaScript (ES6+), React, Redux Toolkit, React Router, Axios

Backend: Node.js, Express.js, REST APIs, JWT, bcrypt

Database: MongoDB, MongoDB Atlas, Mongoose

Testing and DevOps: Jest, React Testing Library, Supertest, Postman, Git, GitHub, Docker, GitHub Actions

AI: OpenAI API, Gemini API, embeddings, vector search

Greens Technologies