MEAN Stack Development with AI Integration - Course Syllabus

Building end-to-end applications with MongoDB, Express.js, Angular and Node.js, plus practical AI. 15 modules, 8 AI topics, 1 mini project.

Course overview

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

Why learn MEAN Stack?

One language (JavaScript / TypeScript) 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 MEAN 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 - Modules 1-3: Frontend Foundations: HTML, CSS, Bootstrap, JavaScript, TypeScript
Stage 2 - Modules 4-6: Angular: components, routing, forms, services, RxJS
Stage 3 - Modules 7-8: Node.js and Express.js: server, REST APIs, middleware
Stage 4 - Modules 9-11: MongoDB and Mongoose: CRUD, aggregation, schema design
Stage 5 - Modules 12-13: Security and Quality: JWT auth, testing
Stage 6 - Modules 14-15: Tools and Deployment: Git, GitHub, Docker, CI/CD, cloud
AI Stage - AI Topics 1-8: AI Integration: OpenAI / Gemini, chatbot, semantic search, mini project

Part 1 - Frontend Foundations (3 Modules)

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: A To-Do app with local storage

3. TypeScript

Why TypeScript; setup and compiler options, Types, interfaces, enums, generics and type inference, Classes, access modifiers, inheritance and abstract classes, Decorators and modules
LAB: Convert a JavaScript program to TypeScript

Part 2 - Angular (3 Modules)

4. Angular Fundamentals

Angular architecture, Angular CLI and project structure, Components, templates, data binding (interpolation, property, event, two-way), Directives: *ngIf, *ngFor, ngClass, ngStyle and custom directives, Pipes (built-in and custom); component communication with @Input and @Output, Component lifecycle hooks

Groins Technologies

5. Services, Routing and Forms

Services and dependency injection, Angular Router: routes, parameters, child routes, lazy loading and route guards, Template-driven forms and reactive forms with validation, HttpClient: GET, POST, PUT and DELETE calls to REST APIs; interceptors, Handling loading, success and error states

6. RxJS and State Management

Observables, Subjects and BehaviorSubject, Operators: map, filter, switchMap, debounceTime and catchError, Async pipe and unsubscribing safely, State management basics with services; introduction to NgRx
LAB: A product listing app with search, cart and API integration

Part 3 - Backend: Node.js, Express.js and MongoDB (5 Modules)

7. 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: A file-handling and basic HTTP server program

8. 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

9. 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)

10. 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

11. 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

Part 4 - Security, Testing and Deployment (4 Modules)

12. Authentication and Security

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

13. Testing

Unit testing Angular with Jasmine and Karma, API testing in Node.js with Jest and Supertest, Mocking services and databases
LAB: Write tests for a controller and an Angular component

Greens Technologies

14. 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

15. Deployment and DevOps Basics

Production builds for Angular and Node.js; environment configuration, Docker basics: images, containers and docker-compose for the MEAN stack, CI/CD introduction with GitHub Actions, Deploying frontend, backend and MongoDB Atlas to a cloud platform, Process management with PM2 and reverse proxy basics (Nginx)

Part 5 - AI Integration (8 AI Topics)

Course objective

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

AI 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: Angular -> Express API -> AI provider -> response, Provider selection: capabilities, pricing, limits and privacy
LAB: Make a first AI API request from Node.js

AI 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 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 an Angular front end to the AI endpoint
LAB: An endpoint that returns an AI-generated response

AI 4. AI Chatbot

Chat request/response flow, system prompts and conversation context, Storing conversation history in MongoDB, Angular chat UI with loading states and failure handling
LAB: A customer-support chatbot

AI 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: A text summarisation endpoint

AI 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: An ingestion pipeline that extracts, chunks, embeds and indexes documents

AI 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: A semantic-search endpoint over indexed documents

Greens Technologies

AI 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.

Required features

1. Frontend: Angular support request and chat interface, 2. Backend: Node.js / Express REST API with routes, controllers and services, 3. Database: MongoDB (Mongoose) for requests, users and chat history, 4. AI: OpenAI or Gemini through a backend service, with document search, 5. Security: JWT login, role-based access and protected API keys, 6. Testing and delivery: unit and API tests, then GitHub and cloud deployment.

Learning outcomes

Build responsive Angular applications with TypeScript, 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 & Technologies

Frontend

HTML5, CSS3, Bootstrap, JavaScript (ES6+), TypeScript, Angular, RxJS

Backend

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

Database

MongoDB, MongoDB Atlas, Mongoose

Testing & DevOps

Jasmine, Karma, Jest, Supertest, Postman, Git, GitHub, Docker, GitHub Actions

AI

OpenAI API, Gemini API, embeddings, vector search

Greens Technologies