

MERN Stack Full-Stack Training (45 Days)

Overview:

This 45-day program is designed for **absolute beginners**, introducing modern web development with the MERN stack (MongoDB, Express.js, React, Node.js). Students will start from the basics of how the web works, then learn front-end (HTML, CSS, JavaScript) and back-end (Node.js/Express, MongoDB) development in tandem.

Course Objectives:

- Understanding client-server architecture, mastering HTML/CSS layouts and responsive design, writing JavaScript logic, building RESTful APIs with Node/Express, and developing dynamic UIs with React.
- By incorporating tools like Git and modern workflows (deployment on cloud), the course ensures graduates can build and deploy complete web applications.
- Hands-on projects are integrated throughout: as CodeQuotient notes, practical projects “allow students to apply their theoretical knowledge to real-world scenarios”, which is one of the most effective learning methods.

Program Structure: The 45 sessions are organized into weekly modules balancing theory and practice. Each week ends with a small project or assessment to reinforce learning.

- **Week 1: Web Basics (HTML & CSS)** – *Day 1:* Course orientation, setting up the development environment (VS Code, Node.js, Git). *Days 2–5:* HTML fundamentals (elements, semantics, forms) and CSS basics (selectors, box model, colors). *Deliverable:* A static portfolio homepage with responsive layout.
- **Week 2: JavaScript Fundamentals** – *Days 6–10:* Core JavaScript syntax and programming concepts (variables, loops, conditionals, functions, ES6 features). DOM manipulation and event handling to make pages interactive. *Deliverable:* A dynamic webpage (e.g. interactive form or quiz app) and a short quiz on JS basics.
- **Week 3: JavaScript & Tooling** – *Days 11–15:* Advanced JavaScript (arrays, objects, ES6 classes), introduction to Node.js fundamentals. Version control with Git (commits, branches) and NPM package management. *Deliverable:* A command-line Node script (e.g. simple calculator) and a Git-based code submission.
- **Week 4: Backend with Node.js and Express** – *Days 16–20:* Building a server-side API. Learn Express.js framework: routing, middleware, REST architecture. Connect a simple in-memory data store. *Deliverable:* A RESTful API (e.g. a JSON “to-do” list service) with testing (using Postman or curl).

- **Week 5: Databases (MongoDB)** – Days 21–25: Introduction to NoSQL and MongoDB (CRUD operations, collections, documents). Use Mongoose for object modeling. Integrate MongoDB with the Express API (storing form data or user info). *Deliverable:* A persistent backend service (e.g. notes app) using MongoDB for data storage.
- **Week 6: React Fundamentals (Front-End)** – Days 26–30: React basics: JSX, components, props/state, event handling, component lifecycle (functional components with hooks). Practice building UI components and styling them. *Deliverable:* A React app (e.g. a todo list UI) with client-side interactivity.
- **Week 7: Full-Stack Integration & Advanced React** – Days 31–35: Connecting front-end to back-end: fetch API calls from React to Express/MongoDB. Advanced React topics: React Router (single-page app navigation), React Context or basic state management. *Deliverable:* A small full-stack MERN project (e.g. a blog or inventory app) where React frontend communicates with Node/Express/MongoDB backend to perform CRUD operations.
- **Week 8: Tools, Deployment, and Testing** – Days 36–40: Cover development tools and deployment. Version control revisited (collaboration with Git), introduction to build tools (Webpack or Create-React-App), and basic testing (unit testing with Jest or manual testing). Deployment on a cloud platform (e.g. Heroku or Netlify) is demonstrated. *Deliverable:* Deploy the Week 7 project to a live URL. Also, mock technical interview questions or quiz.
- **Week 9: Capstone Project & Review** – Days 41–45: Students plan and build a **capstone MERN application** individually or in small teams (e.g. a simple e-commerce site, social feed, or portfolio app). Daily stand-ups and instructor mentoring simulate real team workflow. Final two days are for polishing the app, writing a brief project report, and a presentation/demo. *Deliverable:* A functioning MERN app demonstrating end-to-end skills. Final assessment includes code review, a short quiz covering all topics, and a project evaluation.

Learning Outcomes & Career Readiness

- Upon completion, students will be able to **create and deploy simple full-stack web applications**. They will know how to “develop front end website architecture” and “design user interactions” (using HTML/CSS and React), as well as “develop back-end website applications” (using Node.js/Express) and “create servers and databases for functionality” (using MongoDB).
- In practical terms, learners can design responsive UI layouts, build dynamic pages with React components, implement REST APIs, and connect the front end and back end through JSON/HTTP.
- They will also gain familiarity with version control (Git) and basic deployment. By building multiple mini-projects and a capstone app, students will leave with a small portfolio of work.

- The program equips learners with the “skills to create and manage full-scale web applications” using the MERN technologies, preparing them for entry-level roles or further study in web development.
- By the end of 45 days, students will have a solid foundation in web development. They can **build and deploy a basic full-stack web application** using the MERN stack. Specifically, they will be capable of writing semantic HTML/CSS, implementing JavaScript-driven UI interactivity, designing RESTful APIs with Express, and performing database operations in MongoDB.
- They will also understand developer workflows (using Git) and have experience deploying an app to the cloud. These skills enable graduates to undertake entry-level projects and continue learning advanced topics (e.g. mobile-responsive design, authentication, cloud services). In short, students will leave with practical experience and confidence to pursue roles like Junior Web Developer or to continue with more advanced full-stack studies.