

MERN Full-Stack Development Learning Pathway

FA-0814 · 12 weeks; two 4-hour classes per week

Software Development · DevOps, Cloud & Infrastructure

Introduction

A three-month, 12-week pathway for fresh graduates and learners with programming foundations. Build a to-do application progressively with HTML/CSS, JavaScript, React, Node.js, Express and MongoDB, then practise authentication and deployment before a group project.

Two four-hour classes per week are supported by assignments and self-study. The pathway develops practical foundations and project experience, not a guarantee of professional mastery, employment or production-grade security. Use supported compatible tool versions and prepared deployment resources.

Learning outcomes

- Create responsive web interfaces with HTML, CSS, JavaScript and React.
- Develop introductory Node/Express REST APIs with CRUD and MongoDB persistence.
- Model application data and connect frontend and backend components.
- Implement a guided authentication workflow and review input, error-handling and security concerns.
- Deploy a learning application using a prepared Ubuntu/NGINX/process-management environment and explain CI/CD basics.
- Use Git/GitHub, code reviews and teamwork to deliver and present a group project.

Prerequisites

Background and programming

A computing-related degree/diploma is helpful but not mandatory where equivalent foundations exist. Familiarity with variables, loops, conditions, functions, data structures and basic OOP is expected.

Web and tools

Basic HTML/CSS/JavaScript and client-server concepts. Ability to install selected tools, use a terminal/package manager and troubleshoot setup; Git familiarity is useful.

Study readiness

Problem-solving, communication and documentation-reading skills, motivation for independent practice, a reliable computer and access to the selected tools/resources. Commit to classes, assignments and teamwork; deployment exercises use prepared approved infrastructure.

Training outline

1. Weeks 1–2 — Foundations and environment

- Objective: Understand the basics of web development and set up the necessary environment.

Topics:

- Introduction to Web Development (HTML, CSS, JavaScript basics).
- Setting up the development environment (Node.js, code editor like VSCode).
- Introduction to Git for version control.
- Assignment: Create a simple static webpage with HTML/CSS and host it on GitHub Pages.

2. Weeks 3–4 — JavaScript and Node.js

- Objective: extend JavaScript skills and introduce Node.js server-side development

Topics:

- Advanced JavaScript (ES6+ features, asynchronous programming).
- Introduction to Node.js (creating a simple server, understanding npm).
- Assignment: build a simple Node.js to-do application with introductory in-memory CRUD; persistence follows later

3. Weeks 5–6 — React frontend

- Objective: Learn to create dynamic user interfaces with React.

Topics:

- React basics (components, state, props).
- React Hooks and an introduction to additional state-management approaches
- Introduction to CSS frameworks (like Bootstrap or TailwindCSS) for UI styling.
- Assignment: Convert the to-do list application into a single-page application using React.

4. Weeks 7–8 — Express and MongoDB

- Objective: Learn to build RESTful APIs and understand database operations.

Topics:

- Building RESTful APIs with Node.js and Express.
- Introduction to MongoDB and Mongoose for database management.
- Connecting the React front-end with the Express back-end.
- Assignment: Add database functionality to the to-do list app, allowing data persistence.

5. Weeks 9–10 — Authentication and API concerns

- Objective: Implement user authentication and understand advanced back-end concepts.

Topics:

- Guided JWT authentication: token verification, expiry and appropriate handling; not a complete security solution
- Understanding middleware, error handling, and API security.
- Assignment: Add user authentication to the to-do list app, allowing user registration and login.

6. Week 11 — Deployment and DevOps basics

- Objective: Learn the basics of deployment, server management, and continuous integration.

Topics:

- Introduction to Linux commands and server management with Ubuntu.
- Deploying applications with NGINX and PM2.
- Basics of continuous integration/continuous deployment (CI/CD) pipelines.
- Assignment: deploy the learning app to a selected prepared provider/environment using an available supported configuration

7. Week 12 – Group project and review

- Objective: integrate the learned concepts in a group project and review next learning needs

Topics:

- Develop a scoped group project incorporating the learned technologies
- Code reviews and team collaboration practices.
- Assignment: complete and demonstrate the scoped group project, deploy where the prepared environment permits, and present the work

Technology roles

- HTML structures content; CSS styles it; JavaScript implements browser behaviour.
- React builds component-based interfaces; Node.js runs JavaScript outside the browser.
- Express supplies server routing/middleware; MongoDB stores document-oriented application data, with Mongoose introduced as a modelling tool.
- Git tracks versions; GitHub hosts repositories and collaboration workflows.
- Ubuntu supplies the selected Linux environment; NGINX and PM2 are introduced for the guided deployment.

Learning and evaluation

Assignments, peer collaboration, code reviews, feedback and short assessments support progress. Use current documentation and selected tutorials for further practice.