

# Full Stack Development Curriculum

# **Course Overview**

- Duration: 12 weeks
- Format: Combined lectures, live coding demonstrations, and hands-on practice
- Final Project: Simple but complete web application
- Daily Structure:
  - Live coding demonstration (1 hour)
  - Hands-on practice (2.5 hours) OR
  - Q&A session (1 hour)

# **About Course**

Our Full Stack Development course is designed to equip you with the complete skill set required to build modern web applications. You'll learn both front-end and back-end technologies, including HTML, CSS, JavaScript, React, Node.js, and databases like MongoDB. The course focuses on hands-on projects, giving you practical experience in real-world development. By the end of the course, you'll have the ability to design, develop, and deploy fully functional web applications, making you job-ready for a career in web development.

# **Month 1: Frontend Development Fundamentals**

# Week 1: HTML & CSS Foundations

- Day 1-3: HTML
  - Document structure and elements
  - Text formatting and links
  - Images and media
  - Tables
  - Forms and inputs
  - Practice Project: Personal portfolio structure

#### • Day 4-5: CSS Foundations

- CSS syntax and selectors
- Colors and backgrounds
- Typography and text styling
- Box model basics
- Practice Project: Styling the portfolio

### Week 2: CSS & JavaScript Introduction

- Day 1-2: Advanced CSS
  - Flexbox layout
  - CSS Grid basics
  - Responsive design fundamentals
  - Practice Project: Responsive landing page

#### • Day 3-5: JavaScript Basics

- Variables and data types
- Operators and expressions
- Control structures (if/else, loops)
- Functions introduction
- Practice Project: Basic calculator

# Week 3: JavaScript Core Concepts

- Day 1-2: JavaScript Fundamentals
  - Arrays and array methods
  - Objects and properties
  - Scope and closures
  - Error handling basics

#### • Day 3-4: DOM Manipulation

- Selecting elements
- Modifying content and styles
- Event handling
- Form validation

#### • Day 5: JavaScript Project

- Interactive todo list
- Local storage implementation

### Week 4: JavaScript & React Introduction

#### • Day 1-2: Advanced JavaScript

- ES6+ features
- Promises basics
- Async/await introduction
- Fetch API
- Day 3-5: React Basics
  - React introduction
  - Create React App
  - JSX syntax
  - Components basics
  - Practice Project: Converting todo list to React

# **Month 2: Frontend Framework & Backend Basics**

### Week 5: React Development

#### Day 1-2: React Components

- Functional components
- Props
- useState hook
- Event handling

#### • Day 3-4: React State & Effects

- useEffect hook
- Component lifecycle
- Lists and keys
- Conditional rendering

#### • Day 5: React Project

- Task tracker application
- Component organization

### Week 6: React & Backend Introduction

#### • Day 1-3: Advanced React

- Forms in React
- Data fetching
- Error handling
- Loading states
- Day 4-5: Backend Fundamentals
  - Introduction to Node.js
  - NPM basics
  - Express.js setup
  - Basic routing

# Week 7: Express.js & MongoDB Basics

• Day 1-2: Express.js

- Routing
- Middleware concept
- Request handling
- Response formatting
- Day 3-5: MongoDB
  - Database concepts
  - MongoDB Atlas setup
  - CRUD operations
  - Mongoose basics
  - Practice Project: Basic API development

### Week 8: Full Stack Integration

- Day 1-2: Backend API Development
  - RESTful API principles
  - Route organization
  - Error handling
  - API testing with Postman
- Day 3-5: Frontend-Backend Connection
  - API integration with React
  - Managing API states
  - Error handling in UI
  - Practice Project: Full stack task manager

# **Month 3: Authentication & Projects**

#### Week 9: Authentication

- Day 1-2: Backend Authentication
  - User model
  - Password hashing

• JWT implementation

#### • Day 3-5: Frontend Authentication

- Login/Register forms
- Protected routes
- Authentication state
- Practice Project: Auth implementation

### Week 10: Project Development

- Full stack project development
- User requirements
- Feature implementation
- Database design
- UI/UX considerations

### Week 11-12: Final Project

- Project completion
- Testing and debugging
- Basic deployment
- Project presentations

# **Final Project Options**

- 1. Personal Blog
- 2. E-commerce Product Listing
- 3. Task Management System
- 4. Recipe Sharing Platform

# **Assessment Methods**

- 1. Weekly coding assignments (40%)
- 2. Class participation and quizzes (20%)
- 3. Final project (40%)

# **Learning Resources**

- Course slides and recordings
- Practice code repositories
- Documentation references