Full Stack Development Course Syllabus

Total Duration: 90hrs

Module 1: Frontend Development

Chapter 1: HTML5, CSS3 & Web Foundations

- Semantic HTML and Accessibility Principles
- Forms, Tables, and Multimedia Elements
- CSS Fundamentals: Box Model, Float, Display, Position
- Layout Systems: Flexbox and Grid

Chapter 2: JavaScript Essentials

- ES6+ Syntax: let/const, arrow functions, template literals
- Variables, Functions, Arrays, Objects, Loops
- DOM Manipulation and Event Handling
- Form Validation and Local Storage
- Asynchronous JavaScript: Fetch API, Promises, Async/Await

Chapter 3: Responsive Design & Styling Frameworks

- Mobile-First Design and Media Queries
- Bootstrap 5 Components and Layout System
- Tailwind CSS: Utility-First Styling
- CSS Transitions, Animations, and Effects
- Designing Interfaces in Figma (UI wireframes)

Chapter 4: React.js Development

- JSX Syntax and Project Setup
- Components, Props, and State
- useState and useEffect Hooks
- React Forms and Controlled Components
- React Router for SPA Navigation
- Conditional Rendering and Event Handling

Chapter 5: API Integration & Deployment

- Calling External APIs with Axios

- Handling and Displaying API Data in React

- Debugging with DevTools

- Deploying React Projects (Vercel / Netlify)

- Basic SEO for React Single Page Applications

Chapter 6: Frontend Mini Project

Build a responsive React application, such as:

- Recipe Finder, Weather App, Product Showcase

Requirements:

- Use an external API (e.g., OpenWeather, Spoonacular)

- Implement search, filter, and responsive layout

- Deploy and submit a live link

Module 2: Backend Development & Full-Stack Integration

Chapter 7: Backend Fundamentals

Choose one stack:

Option A: Node.js + Express.js

- Express Project Structure and MVC Pattern

- Routing, Middleware, and Controllers

- REST API Basics

- Sequelize ORM with SQL Database

- Authentication with JWT

- File Uploads using Multer

Option B: Python (Django or Flask)

- Setup and MVC Architecture

- ORM with Models and Relationships

- Django REST Framework or Flask-Restful

- JWT Authentication and Role-Based Access

- File Upload and Media Management

- Deployment: Gunicorn + PostgreSQL + Nginx

Option C: PHP (Laravel)

- Laravel Project Setup and MVC

- Eloquent ORM with Relationships

- Sanctum / Passport for Authentication

- Middleware and Form Validation

- File Upload and Storage

- Deployment via Laravel Forge or Shared Hosting

Chapter 8: Database Management

- SQL Fundamentals: Tables, Relationships, Joins

- Using ORM for CRUD Operations

- Data Seeding and Migration

- Data Validation and Query Optimization

Chapter 9: REST API Development

- RESTful API Design Principles

- CRUD Endpoints with Authentication

- Role-Based Access Control

- Error Handling and Logging

- Testing with Postman or Thunder Client

Chapter 10: Full-Stack Integration

- Connecting React Frontend to Backend API

- Axios Integration and Secure API Calls

- Protected Routes and Authentication Flows

- Dynamic Forms and Server-Side Validation

- Managing Frontend State Based on Backend

Chapter 11: Deployment and Optimization

- Deploying Backend on Render, Railway, or Heroku

- Managing Environment Variables (.env)

- SQL Database Hosting and Connection

- Full-Stack App Hosting and Domain Integration

- SEO, Performance Tuning, and Production Best Practices

Chapter 12: Final Full-Stack Capstone Project

Build and deploy a complete full-stack application, such as:

- E-commerce Store, Blogging Platform, or Task Manager

Project Requirements:

- React for frontend interface

- Custom backend API with selected stack

- SQL database integration using ORM

- Full user authentication with JWT

- Form validation, protected routes, role-based access

- File uploads and dynamic data management

- Clean project structure and optimized code

- Fully deployed with custom domain

- Final submission includes live URL and source code for review