100 Days of Code Challenge –Full Stack Web Developer--

Days 1-10: Front-End Fundamentals

Day 1-3: HTML Basics

Learn about HTML structure, tags, and elements.

Build simple web pages with headings, paragraphs, lists, and links.

Day 4-6: CSS Basics

Understand CSS selectors, properties, and values.

Style your HTML pages with colors, fonts, margins, and padding.

Day 7-10: Intermediate CSS

Learn about layout techniques, positioning, and responsive design.

Build a responsive webpage using media queries.

Days 11-20: JavaScript Basics

Day 11-14: JavaScript Fundamentals

Learn variables, data types, operators, and control structures.

Practice writing basic scripts to manipulate the DOM.

Day 15-18: Functions and Objects

Understand functions, scope, closures, and the this keyword.

Learn about objects, methods, and prototypes.

Day 19-20: DOM Manipulation

Explore how to select and modify elements on a webpage using JavaScript.

Build interactive elements like buttons and forms.

Days 21-30: Advanced JavaScript & React Basics

Day 21-24: ES6+ Features

Dive into modern JavaScript features like arrow functions, destructuring, and classes.

Update your existing JavaScript code to use ES6+ syntax.

Day 25-28: Introduction to React

Learn about React components, JSX, and rendering.

Build a simple React application with components and props.

Day 29-30: React State and Events

Understand component state and how to handle user events.

Implement state changes and interactivity in your React app.

Days 31-40: React and Front-End Development

Day 31-34: React Hooks

Explore useState, useEffect, and other hooks for managing state and side effects.

Refactor your React app to use hooks.

Day 35-38: React Routing

Learn about React Router and implement navigation in your app.

Create different routes for different components.

Day 39-40: Styling in React

Explore CSS-in-JS libraries like styled-components.

Style your React components using these libraries.

Days 41-50: Back-End Development Basics

Day 41-44: Introduction to Node.js

Learn about Node.js, its asynchronous nature, and the event loop.

Create a basic server using Node.js.

Day 45-48: Express.js Framework

Understand the basics of Express.js, routing, and middleware.

Build API endpoints to handle HTTP requests.

Day 49-50: Integrating Front-End and Back-End

Connect your React front-end to your Express.js back-end.

Implement API calls from the React app to fetch data from the server.

Days 51-60: Databases and Advanced Front-End

Day 51-54: MongoDB Basics

Learn about MongoDB, collections, documents, and CRUD operations.

Set up a connection between your Express.js server and a MongoDB database.

Day 55-58: Advanced React Concepts

Explore more complex React patterns like context, higher-order components, and render props.

Apply these patterns to enhance your React application.

Day 59-60: State Management with Redux (Optional)

If you choose, learn about Redux for managing complex application state.

Implement Redux to manage global state in your React app.

Days 61-70: Authentication and User Management

Day 61-64: User Authentication

Learn about authentication methods like JWT (JSON Web Tokens).

Implement user registration and login functionality in your app.

Day 65-68: Protected Routes

Create protected routes that require authentication to access.

Implement access control for different user roles.

Day 69-70: User Profiles and Settings

Allow users to update their profiles and account settings.

Implement password reset functionality.

Days 71-80: RESTful APIs and Deployment

Day 71-74: Advanced API Design

Dive deeper into RESTful API design principles.

Implement pagination, sorting, and filtering in your API.

Day 75-78: Deployment Basics

Explore deployment options like Heroku, Netlify, and VPS.

Deploy your front-end and back-end applications.

Day 79-80: Error Handling and Validation

Implement error handling mechanisms in your back-end.

Validate user input on both the front-end and back-end.

Days 81-90: Real-Time Communication and Final Projects

Day 81-84: WebSocket Communication

Learn about real-time communication using technologies like WebSockets.

Implement a real-time chat feature in your application.

Day 85-88: Performance Optimization

Explore techniques to optimize your application's performance.

Minimize load times and enhance user experience.

Day 89-90: Finalizing Projects

Spend time polishing and refining your projects.

Implement any remaining features or improvements.

Days 91-100: Portfolio and Beyond

Day 91-94: Create a Portfolio

Build a personal portfolio website showcasing your projects and skills.

Deploy your portfolio and link to your GitHub repositories.

Day 95-98: Continuous Learning

Stay updated with the latest trends in web development.

Explore additional technologies like GraphQL, Docker, etc.

Day 99-100: Reflection and Future Goals

Reflect on your journey and how much you've learned.

Set goals for your future as a Full Stack Web Developer.