Roadmap: Full-Stack Development

September 2024 - April 2025

HTML, CSS, and CSS Frameworks

September 2024 - October 2024: 2 hours daily

Week 1: HTML Basics (Sep 1 - Sep 7)

- **Day 1-2: Introduction to HTML**
  - What is HTML?
  - HTML syntax and structure
  - Basic tags: `<html>`, `<head>`, `<body>`
  - Headings (`<h1>` to `<h6>`)
  - Paragraphs, lists (`<ul>`, `<ol>`, `<li>`), block vs. inline elements
- **Day 3-4: Working with Links and Images**
  - Anchor tag (`<a>`), adding links
  - Image tag (`<img>`)
  - Linking images and texts
  - Absolute vs. relative URLs
- **Day 5-6: Forms and Input Elements**
  - Form basics (`<form>`, `<input>`, `<label>`)
  - Common input types (text, email, password, checkbox, radio, etc.)
  - Form validation attributes (required, placeholder, etc.)
  - Buttons (`<button>`, `<input type="submit">`)
- **Day 7: Semantic HTML**
  - Why use semantic HTML?
  - Tags: `<section>`, `<article>`, `<nav>`, `<header>`, `<footer>`, `<aside>`, `<main>`
  - HTML5 new elements

Week 2: Advanced HTML & Introduction to CSS (Sep 8 - Sep 14)

- **Day 8-9: Multimedia in HTML**
  - Embedding videos (`<video>`)
  - Embedding audio (`<audio>`)
  - Working with `<iframe>` for embedding external content
- **Day 10-11: Introduction to CSS**
  - What is CSS? How it works
  - Inline, internal, external CSS
  - CSS syntax: selectors, properties, and values
  - Colors in CSS: color names, hex, RGB, HSL

- **Day 12-14: Text and Fonts in CSS**
    - Styling text: color, font-size, font-family
    - Text alignment, line height, text decoration
    - Using Google Fonts or custom fonts

Week 3: CSS Layouts and Flexbox (Sep 15 - Sep 21)

- **Day 15-16: Box Model**
    - Understanding the box model
    - Margins, padding, borders, and content
    - `box-sizing` property
- **Day 17-18: CSS Positioning**
    - Static, relative, absolute, fixed, and sticky positioning
    - `z-index` and stacking context
    - Floating elements and clearing floats
- **Day 19-21: Flexbox**
    - Flexbox introduction: `display: flex`
    - Main axis, cross axis, `flex-direction`
    - Aligning items, `justify-content`, and `flex-grow`/`flex-shrink` properties
    - Building layouts with Flexbox

Week 4: Responsive Design and CSS Grid (Sep 22 - Sep 28)

- **Day 22-23: Responsive Design Concepts**
    - What is responsive design?
    - Viewport, media queries
    - Mobile-first design
- **Day 24-26: CSS Grid Layout**
    - Introduction to CSS Grid: `display: grid`
    - Creating grid containers and items
    - Grid template areas, rows, and columns
    - Aligning and placing elements in the grid
- **Day 27-28: Advanced Responsive Design**
    - Fluid layouts, responsive typography
    - Using media queries for different breakpoints
    - Flexbox vs. Grid for responsive layouts

Week 5: Introduction to CSS Frameworks (Sep 29 - Oct 5)

- **Day 29-30: Introduction to Bootstrap**
    - What is Bootstrap? Why use it?
    - Setting up Bootstrap
    - Bootstrap Grid system: rows, columns
    - Bootstrap Containers: `container`, `container-fluid`
- **Day 31-32: Bootstrap Components**

    - Bootstrap navigation bars, buttons, forms
    - Bootstrap cards, modals, and dropdowns
    - Using Bootstrap utilities (margins, paddings, colors)
- **Day 33-35: Responsive Design with Bootstrap**
    - Bootstrap responsive utilities
    - Responsive breakpoints (`col-sm`, `col-md`, `col-lg`)
    - Building a responsive layout with Bootstrap grid

Week 6: Tailwind CSS and Advanced CSS (Oct 6 - Oct 12)

- **Day 36-38: Introduction to Tailwind CSS**
    - What is Tailwind CSS? Why use it?
    - Setting up Tailwind CSS in a project
    - Utility-first CSS: classes and custom styling
    - Building layouts using Tailwind's grid and flex utilities
- **Day 39-40: Tailwind CSS Responsive Design**
    - Handling responsive layouts in Tailwind
    - Breakpoints and customizations
    - Tailwind typography and spacing utilities
- **Day 41-42: Tailwind Components**
    - Buttons, forms, and cards in Tailwind
    - Using Tailwind’s pre-built components
    - Building reusable components in Tailwind

Week 7: Responsive Web Design and Best Practices (Oct 13 - Oct 19)

- **Day 43-45: Advanced Responsive Design Techniques**
    - Mobile-first responsive design
    - CSS frameworks for responsive grids
    - Optimizing images and media for responsiveness
- **Day 46-47: Cross-browser Compatibility**
    - Ensuring CSS works across all major browsers
    - Using CSS resets and normalize.css
    - Testing responsiveness and compatibility
- **Day 48-49: Web Accessibility**
    - What is web accessibility?
    - Semantic HTML and ARIA roles for accessibility
    - Ensuring forms and media are accessible
    - Accessibility testing tools

Week 8: Projects and Final Review (Oct 20 - Oct 31)

- **Day 50-55: Project Implementation**
    - Building 2-3 mini-projects combining HTML, CSS, Bootstrap, and Tailwind
    - Focus on responsive design

    - Testing across multiple devices
- **Day 56-60: Final Review**
    - Revising all key concepts: HTML, CSS, responsive design
    - Testing projects for responsiveness and cross-browser compatibility
    - Ensuring best practices for accessibility and responsiveness

Professional Project Ideas:

1. **Responsive Portfolio Website:** Create a personal portfolio showcasing projects, experience, and contact details.
2. **E-commerce Homepage:** Build a responsive homepage for an e-commerce website with product grids and responsive navigation.
3. **Restaurant Website:** Design a fully responsive website for a restaurant, including menus, location info, and reservation forms.
4. **Landing Page for a Startup:** Create a modern, responsive landing page for a tech startup, with sign-up forms and interactive components.
5. **Blog Website:** Build a responsive blog layout with different article previews, categories, and an article page.

JavaScript Full-Stack Programmer Roadmap

September 2024 - April 2025: 8 hours daily

Month 1: JavaScript Fundamentals

- **Week 1: Introduction to JavaScript (Sep 1 - Sep 7)**
    - **Day 1-2:**
        - Overview of JavaScript
        - Setting up Development Environment (Node.js, VS Code)
        - Understanding ES6+ features
    - **Day 3-4:**
        - Syntax, Variables, Constants, Data Types (Primitive, Non-Primitive)
        - Declaring variables (`var`, `let`, `const`)
    - **Day 5-7:**
        - Operators: Arithmetic, Assignment, Comparison, Logical, Bitwise
        - Control Structures (if-else, switch, loops)
- **Week 2: Functions and Scope (Sep 8 - Sep 14)**
    - **Day 1-2:**
        - Functions (Declaration, Expression, Arrow functions)
    - **Day 3-4:**
        - Understanding Scope: Block, Function, Global
        - Closures: Lexical Environment, Use Cases

    - **Day 5-7:**
        - Function Expressions, Hoisting, IIFE (Immediately Invoked Function Expression)
- **Week 3: Objects and Arrays (Sep 15 - Sep 21)**
    - **Day 1-2:**
        - Object Creation: Properties, Methods
        - Object Destructuring, `this` keyword
    - **Day 3-4:**
        - Arrays: Methods (`map`, `filter`, `reduce`), Iteration (`forEach`, `for-of`)
    - **Day 5-7:**
        - Object-Oriented Programming (OOP) in JavaScript (Classes, Constructors, Inheritance)
- **Week 4: Advanced JavaScript Concepts (Sep 22 - Sep 30)**
    - **Day 1-2:**
        - Prototypes, Prototype Chain, Inheritance Models
    - **Day 3-4:**
        - Asynchronous JavaScript (Callbacks, Promises, Async/Await)
    - **Day 5-7:**
        - Error Handling (`try-catch`), Debugging with console, DevTools

Month 2: DOM Manipulation and Events

- **Week 1: DOM Basics (Oct 1 - Oct 7)**
    - **Day 1-2:**
        - Introduction to DOM, Document Object Model
        - Selecting DOM Elements (`getElementById`, `querySelector`)
    - **Day 3-4:**
        - Modifying DOM Elements (innerHTML, textContent, attributes)
    - **Day 5-7:**
        - Creating and Appending Elements (`createElement`, `appendChild`)
- **Week 2: Events and Event Handling (Oct 8 - Oct 14)**
    - **Day 1-2:**
        - Introduction to Events: Types, Event Object
        - Event Listeners (`addEventListener`)
    - **Day 3-4:**
        - Event Delegation and Bubbling
        - Handling Events in Forms
    - **Day 5-7:**
        - Custom Events, Event Handling Patterns
- **Week 3: Forms and Input Validation (Oct 15 - Oct 21)**
    - **Day 1-2:**
        - Handling Form Submissions
        - Validating User Input (Client-side validation)
    - **Day 3-4:**

        - Regular Expressions for Validation
        - Displaying Error Messages
    - **Day 5-7:**
        - Dynamic Form Elements (Adding/Removing Fields)
- **Week 4: Advanced DOM Manipulation (Oct 22 - Oct 31)**
    - **Day 1-2:**
        - Node Collections and Traversing the DOM
        - Manipulating Styles and Classes (`classList`, `style`)
    - **Day 3-4:**
        - Animations and Transitions with JavaScript
        - Using Browser APIs (Local Storage, Session Storage)
    - **Day 5-7:**
        - Advanced DOM Techniques: Drag and Drop, Canvas API

Month 3: Data Structures and Algorithms (DSA) in JavaScript

- **Week 1: Basics of DSA (Nov 1 - Nov 7)**
    - **Day 1-2:**
        - Introduction to Data Structures
        - Arrays: Operations, Methods
    - **Day 3-4:**
        - Searching Algorithms: Linear Search, Binary Search
    - **Day 5-7:**
        - Sorting Algorithms: Bubble Sort, Selection Sort, Insertion Sort
- **Week 2: Stacks, Queues, and Hashing (Nov 8 - Nov 14)**
    - **Day 1-2:**
        - Stacks: Operations, Use Cases
        - Implementation in JavaScript
    - **Day 3-4:**
        - Queues: Operations, Use Cases
        - Implementation in JavaScript
    - **Day 5-7:**
        - Hashing: Hash Tables, Collision Resolution
- **Week 3: Trees and Graphs (Nov 15 - Nov 21)**
    - **Day 1-2:**
        - Trees: Binary Trees, Tree Traversals (Pre-order, In-order, Post-order)
    - **Day 3-4:**
        - Graphs: Graph Representation (Adjacency Matrix, List)
        - Graph Traversal Algorithms (DFS, BFS)
    - **Day 5-7:**
        - Applications of Trees and Graphs
- **Week 4: Algorithms (Nov 22 - Nov 30)**
    - **Day 1-2:**
        - Advanced Sorting Algorithms (Quick Sort, Merge Sort)
    - **Day 3-4:**

        - Dynamic Programming: Basics and Problems
    - **Day 5-7:**
        - Practice and Implementing Algorithms

Month 4: Advanced Problem-Solving Skills

- **Week 1: Arrays and Strings (Dec 1 - Dec 7)**
    - **Day 1-2:**
        - Array Problems: Rotation, Merging, Subarrays
    - **Day 3-4:**
        - String Problems: Palindromes, Anagrams, Substrings
    - **Day 5-7:**
        - Practice Problems on LeetCode
- **Week 2: Linked Lists, Stacks, Queues (Dec 8 - Dec 14)**
    - **Day 1-2:**
        - Linked Lists: Operations, Problems
    - **Day 3-4:**
        - Stack Problems: Balanced Parentheses, Evaluation
    - **Day 5-7:**
        - Queue Problems: Circular Queue, Priority Queue
- **Week 3: Tree and Graph Problems (Dec 15 - Dec 21)**
    - **Day 1-2:**
        - Tree Problems: Diameter, Lowest Common Ancestor
    - **Day 3-4:**
        - Graph Problems: Shortest Path, Minimum Spanning Tree
    - **Day 5-7:**
        - Practice Problems on LeetCode
- **Week 4: Advanced Problem Solving (Dec 22 - Dec 31)**
    - **Day 1-3:**
        - Complex Problems: Backtracking, Greedy Algorithms
    - **Day 4-7:**
        - Practice and Review of Advanced Problems

Month 5: Front-End Frameworks (React.js)

- **Week 1: React.js Basics (Jan 1 - Jan 7)**
    - **Day 1-2:**
        - Introduction to React.js
        - Setting up React Environment (Create React App)
    - **Day 3-4:**
        - JSX and Components: Function Components, Class Components
    - **Day 5-7:**
        - Props and State: Passing Data, Managing State
- **Week 2: React Essentials (Jan 8 - Jan 14)**
    - **Day 1-2:**
        - Handling Events in React

        - Conditional Rendering
    - **Day 3-4:**
        - Lists and Keys: Rendering Lists, Unique Keys
    - **Day 5-7:**
        - React Forms: Controlled Components, Handling Form Submission
- **Week 3: Advanced React.js Concepts (Jan 15 - Jan 21)**
    - **Day 1-2:**
        - React Router: Setting Up, Creating Routes
    - **Day 3-4:**
        - Context API: Creating Context, Using Context
    - **Day 5-7:**
        - React Hooks: `useState`, `useEffect`, Custom Hooks
- **Week 4: React Performance and Testing (Jan 22 - Jan 31)**
    - **Day 1-2:**
        - Performance Optimization: Memoization, Lazy Loading
    - **Day 3-4:**
        - Testing React Components: Jest, React Testing Library
    - **Day 5-7:**
        - Redux Basics: Actions, Reducers, Store

Month 6: Back-End Frameworks (Node.js and Express.js)

- **Week 1: Node.js Basics (Feb 1 - Feb 7)**
    - **Day 1-2:**
        - Introduction to Node.js
        - Node.js Environment Setup
    - **Day 3-4:**
        - Node.js Modules: Built-in, Custom Modules
    - **Day 5-7:**
        - File System Module: Reading, Writing Files
- **Week 2: Express.js Basics (Feb 8 - Feb 14)**
    - **Day 1-2:**
        - Introduction to Express.js
        - Setting Up Express Server
    - **Day 3-4:**
        - Routing in Express: Route Parameters, Query Strings
    - **Day 5-7:**
        - Middleware: Using Built-in and Custom Middleware
- **Week 3: REST API Development (Feb 15 - Feb 21)**
    - **Day 1-2:**
        - REST API Concepts: Endpoints, HTTP Methods
        - Creating API Routes in Express
    - **Day 3-4:**
        - Implementing CRUD Operations: Create, Read, Update, Delete
    - **Day 5-7:**
        - Authentication: JWT, Passport.js

- **Week 4: Advanced Node.js (Feb 22 - Feb 28)**
  - **Day 1-2:**
    - Real-Time Applications: WebSockets
  - **Day 3-4:**
    - Error Handling and Debugging
  - **Day 5-7:**
    - Deployment: Hosting Node.js Apps on Platforms like Heroku, Vercel

Month 7: Full-Stack Project Development

- **Weeks 1-4: Building Full-Stack Applications (Mar 1 - Mar 31)**
  - **Week 1:**
    - Planning and Designing the Full-Stack Project
    - Setting Up the Project (Front-End with React.js, Back-End with Node.js/Express.js)
  - **Week 2:**
    - Developing Features: User Authentication, Data Handling
    - Connecting Front-End to Back-End
  - **Week 3:**
    - Testing and Debugging
    - Adding Advanced Features: Real-Time Updates, Notifications
  - **Week 4:**
    - Deployment: Deploying the Complete Application
    - Final Review and Documentation

Month 8: Mastery and Advanced Topics

- **Week 1-2: Advanced JavaScript Concepts (Apr 1 - Apr 14)**
  - **Day 1-2:**
    - Advanced Asynchronous JavaScript: Async/Await, Promises
  - **Day 3-4:**
    - Advanced Functional Programming: Currying, Closures
  - **Day 5-7:**
    - Web Performance Optimization: Techniques and Tools
- **Week 3-4: Advanced React and Node.js (Apr 15 - Apr 30)**
  - **Day 1-2:**
    - Advanced React Patterns: Higher-Order Components, Render Props
  - **Day 3-4:**
    - Server-Side Rendering (SSR) with Next.js
  - **Day 5-7:**
    - Advanced Node.js Concepts: Streams, Buffers, Clustering

Month 9: Real-World Application and Projects

- **Weeks 1-4:**
  - **Week 1:**
    - Analyzing and Choosing a Real-World Project
    - Planning and Initial Setup
  - **Week 2-3:**
    - Developing the Project: Implementing Features, Integration
  - **Week 4:**
    - Final Testing, Deployment, and Presentation