Week 1: HTML & CSS Basics

Day 1: Introduction to Web Development
- What is web development?
- How the web works: Client-server model, browsers, and dev tools.
- Setting up your environment: VS Code, browser dev tools.
- Basics of HTML: Structure of an HTML document (`<!DOCTYPE>`, `<html>`, `<head>`, `<body>`).
- Common HTML tags: `<h1>` to `<h6>`, `<p>`, `<a>`, `<img>`, `<ul>`, `<ol>`, `<li>`.

Day 2: HTML Forms and Semantic HTML
- HTML forms: `<form>`, `<input>`, `<label>`, `<button>`, `<select>`, `<textarea>`.
- Semantic HTML: `<header>`, `<footer>`, `<nav>`, `<section>`, `<article>`, `<aside>`.

Day 3: Introduction to CSS
- What is CSS? Inline, internal, and external CSS.
- CSS selectors: Element, class, ID, and universal selectors.
- Basic styling: `color`, `background-color`, `font-size`, `font-family`, `margin`, `padding`.
- Box model: Content, padding, border, margin.

Day 4: CSS Layouts
- Display properties: `block`, `inline`, `inline-block`, `none`.
- Positioning: `static`, `relative`, `absolute`, `fixed`, `sticky`.
- Flexbox basics: `display: flex`, `flex-direction`, `justify-content`, `align-items`.

Day 5: Responsive Design
- Media queries: `@media` rule for responsive design.
- Mobile-first design principles.
- Introduction to CSS frameworks

Day 6: Project 1 - Build a Static Website
- Create a multi-page static website using HTML and CSS.
- Include a responsive layout, navigation bar, and contact form.

Week 2: JavaScript Basics

Day 1: Introduction to JavaScript
- What is JavaScript? Role in web development.
- Linking JavaScript to HTML: `<script>` tag.
- Variables: `let`, `const`, `var`.
- Data types: Strings, numbers, booleans, arrays, objects.

Day 2: JavaScript Functions and Control Flow
- Functions: Declaration, parameters, return values.
- Conditional statements: `if`, `else if`, `else`, `switch`.
- Loops: `for`, `while`, `do-while`.

Day 3: DOM Manipulation
- What is the DOM?
- Selecting elements: `document.querySelector`, `document.querySelectorAll`.
- Modifying elements: `innerHTML`, `textContent`, `style`.
- Event listeners: `addEventListener`, `click`, `mouseover`, `keydown`.

Day 4: JavaScript Arrays and Objects
- Array methods: `push`, `pop`, `shift`, `unshift`, `map`, `filter`, `reduce`.
- Objects: Properties, methods, and `this` keyword.
- JSON: Introduction to JSON format.

Day 5: Error Handling and Debugging
- Debugging with `console.log`, browser dev tools.
- Error handling: `try`, `catch`, `finally`.

Day 6: Project 2 - Interactive Web Page
- Build a simple interactive web page (e.g., a to-do list or calculator).
- Use JavaScript to handle user input and update the DOM.

Week 3: Advanced JavaScript and Introduction to React

Day 1: ES6+ Features
- Arrow functions.
- Template literals.
- Destructuring.
- Spread and rest operators.
- Modules: `import` and `export`.

Day 2: Asynchronous JavaScript
- Callbacks.
- Promises: `.then`, `.catch`.
- Async/await.
- Fetch API: Making HTTP requests.

Day 3: Introduction to React
- What is React? Why use React?
- Setting up a React environment: `create-react-app`.
- JSX syntax: Combining HTML and JavaScript.
- Components: Functional components, props.

Day 4: React State and Events
- State: `useState` hook.
- Handling events: `onClick`, `onChange`.
- Conditional rendering.

Day 5: React Components and Props
- Component composition.
- Passing data via props.
- Prop drilling and lifting state up.

Day 6: Project 3 - Simple React App
- Build a simple React app (e.g., a counter or a weather app).
- Use components, state, and props.

Week 4: React Basics and Final Project

Day 1: React Lists and Keys
- Rendering lists: `map` method.
- Importance of keys in React.

Day 2: React Forms
- Controlled components.
- Handling form submissions.

Day 3: React Hooks
- `useEffect` hook: Side effects in React.
- Custom hooks.

Day 4: React Router
- Setting up React Router: `BrowserRouter`, `Route`, `Link`.
- Creating multi-page React apps.

Day 5: Styling in React
- CSS modules.
- Styled-components.

- Using CSS frameworks with React.

Day 6-7: Final Project
- Build a capstone project with api consumption.
- Combine HTML, CSS, JavaScript, and React.
- Deploy the project using platforms like Netlify or Vercel.