Web Development Roadmap

This outline is designed to take you from the basics to deploying complex, full-stack applications. You will build three major projects, each one building on the skills of the last.

- **Project 1: Your Personal Portfolio.** A professional, responsive website to showcase your work. This is where you will master the front-end fundamentals.
- **Project 2: "FoodFleet" - E-commerce Food Delivery.** A full-stack application with users, restaurants, menus, and orders. This will be your main project for learning back-end, database, and authentication concepts.
- **Project 3: "ConnectSphere" - Social Media App (Capstone).** A feature-rich platform with user profiles, posts, comments, likes, and followers. This project will introduce advanced topics like real-time communication.

Module 1: The Foundation - HTML & CSS

This module is about creating the structure and style of a website. Think of it as the skeleton and the clothes.

- **Topics to Learn:**
  - **HTML:** Basic document structure, semantic tags (<header>, <nav>, <main>), forms, lists, links, and images.
  - **CSS:** Selectors, the Box Model (margin, border, padding), colors, typography, and layout with CSS Flexbox and CSS Grid.
  - **Responsive Design:** Using media queries to make your site look great on all devices.
- **Project Goal:**
  - **[Project 1] Build the first version of your Portfolio.**
    - Create the HTML structure for four pages: Home, About, Projects, and Contact.
    - Use CSS to style all pages with a clean, professional, and responsive design.
    - **Outcome:** A beautiful, static, multi-page website ready for content.

Module 2: Adding Interactivity - JavaScript

This is where your website comes to life, allowing you to respond to user actions.

- **Topics to Learn:**
  - **JS Fundamentals:** Variables, data types, functions, conditional logic, and loops.
  - **The DOM:** Selecting, creating, and changing HTML elements with JavaScript.
  - **Events:** Handling user actions like clicks, form submissions, and keyboard

presses.
    - **APIs:** Fetching data from external sources using the fetch() API.
- **Project Goal:**
    - **[Project 1] Enhance your Portfolio with JS.**
        - Add a responsive "hamburger" menu for mobile screens.
        - Create a light/dark mode theme switcher.
        - Implement client-side validation for your contact form.
        - Use the GitHub API to fetch and display your public repositories on the "Projects" page.

Module 3: The Server-Side - Node.js & Express

Now we move beyond the browser to build the "behind-the-scenes" engine of a web application.

- **Topics to Learn:**
    - **Back-End Concepts:** Client-server model, HTTP requests (GET, POST, PUT, DELETE), RESTful APIs.
    - **Node.js & NPM:** Running JS on a server and managing packages.
    - **Express.js:** Building web servers and APIs with routing and middleware.
- **Project Goal:**
    - **[Project 2] Create the "FoodFleet" API.**
        - Set up a new Node.js/Express project.
        - Create initial API endpoints (routes) to handle restaurants and users. For now, store the data in simple JavaScript arrays.
        - You'll need routes for: GET /api/restaurants, GET /api/restaurants/:id/menu, and POST /api/users/register.

Module 4: Storing Your Data - Databases

A web application needs a place to permanently store its data.

- **Topics to Learn:**
    - **Database Fundamentals:** SQL vs. NoSQL databases.
    - **NoSQL with Firestore:** Setting up and using a simple, cloud-based NoSQL database.
    - **CRUD Operations:** Implementing Create, Read, Update, and Delete operations.
- **Project Goal:**
    - **[Project 2] Integrate a Database into "FoodFleet".**
        - Connect your Express API to a Firestore database.
        - Create collections for users, restaurants, menuItems, and orders.
        - Modify your API endpoints to save and retrieve data from the database

instead of temporary arrays.

Module 5: Modern Front-Ends - React

It's time to learn the most popular library for building modern, fast, and scalable user interfaces.

- **Topics to Learn:**
  - **Component-Based Architecture:** Thinking in reusable UI pieces.
  - **JSX, State & Props:** Writing UI with JavaScript and managing component data.
  - **Lifecycle & Effects (useEffect):** Fetching data and handling side effects.
  - **Client-Side Routing:** Creating a single-page application (SPA) experience.
- **Project Goal:**
  - **[Project 1] Rebuild your Portfolio with React.**
    - Recreate your entire portfolio as a React SPA, breaking the UI into reusable components (Navbar, ProjectCard, Footer).
  - **[Project 2] Start the "FoodFleet" Front-End.**
    - Create a new React application for the e-commerce site.
    - Build components to display a list of restaurants and a restaurant's menu by fetching data from your FoodFleet API.

Module 6: Authentication & E-commerce Features

Secure your application and add core e-commerce functionality.

- **Topics to Learn:**
  - **Authentication:** Implementing user login/logout with JSON Web Tokens (JWT).
  - **Security:** Hashing passwords, using environment variables for secrets.
  - **Payment Gateways (Intro):** Understanding how to integrate a service like Stripe for payments.
- **Project Goal:**
  - **[Project 2] Secure "FoodFleet" and Build Core Features.**
    - Add authentication to your API. Users must be logged in to place an order.
    - Create the front-end login and registration forms.
    - Build the shopping cart functionality.
    - Implement the "place order" feature, which saves the order to your database.

Module 7: Advanced Concepts & Capstone Project

Apply everything you've learned and tackle advanced topics to build your final capstone project.

- **Topics to Learn:**
  - **Advanced State Management:** Using tools like Context API or Zustand for complex application state.
  - **WebSockets:** Enabling real-time, two-way communication for features like live feeds or chat.
  - **Advanced Database Modeling:** Handling complex relationships (e.g., many-to-many for followers).
  - **File Uploads:** Allowing users to upload images for profiles and posts.
  - **Deployment:** Deploying full-stack applications (front-end, back-end, and database).
- **Project Goal:**
  - **[Project 3] Design and Build "ConnectSphere" (Social Media App).**
    - Plan and build the full-stack application from scratch.
    - Implement user registration, login, and profiles with picture uploads.
    - Create functionality for users to create posts, comment on posts, and "like" posts.
    - Implement a "follower" system.
    - Build a real-time home feed that updates when people you follow post something new.
  - **[All Projects] Deploy Everything!**
    - Deploy your Portfolio, FoodFleet, and ConnectSphere applications so they are live on the web.
    - **Final Outcome:** Three impressive, fully functional web applications that create a powerful portfolio.