Month 1: Mastering the Foundations of the Web

The goal this month is to build a strong understanding of how web pages are structured and styled.

- **Week 1-2: HTML5**
    - **Topics:** Learn the fundamentals: tags, elements, attributes, forms, semantic HTML, accessibility basics.
    - **Goal:** Be able to structure any static web page, like a blog post, a portfolio page, or a product landing page.
    - **Project:** Build a multi-page personal portfolio website using only HTML.
- **Week 3-4: CSS3 & Responsive Design**
    - **Topics:** Selectors, the box model, Flexbox, CSS Grid, positioning, transitions, and animations. Crucially, learn responsive design using media queries to make your sites look good on all devices (desktops, tablets, and phones).
    - **Goal:** Style the HTML projects you've built and make them fully responsive.
    - **Project:** Take your portfolio website and style it completely with CSS. Try to clone the look of a simple, existing website.
- **Throughout the month: Introduction to Developer Tools**
    - **Topics:** Learn to use Git and GitHub for version control. This is a non-negotiable skill for any developer.
    - **Goal:** Create a GitHub repository for every project you build. Practice basic commands like git add, git commit, git push.

Month 2: Diving Deep into JavaScript

This month is dedicated entirely to JavaScript, the programming language of the web.

- **Week 1-2: JavaScript Fundamentals**
    - **Topics:** Variables, data types, operators, functions, loops, conditionals, objects, and arrays.
    - **Goal:** Solve coding challenges on platforms like HackerRank or LeetCode (easy level) to solidify your understanding of logic.
- **Week 3-4: Advanced JavaScript & DOM Manipulation**
    - **Topics:** Asynchronous JavaScript (Callbacks, Promises, async/await), APIs (fetch), ES6+ features (arrow functions, destructuring), and how to interact with the Document Object Model (DOM) to make your web pages interactive.
    - **Goal:** Make your static portfolio website dynamic.
    - **Project:** Build a weather application that fetches data from a free weather API and displays it on the page. Add a "To-Do List" application to your portfolio.

Month 3: Building Modern Front-Ends with React

Now you'll learn to build complex, fast, and scalable user interfaces using a modern framework.

- **Week 1-2: Introduction to React**
    - **Topics:** What is React? Setting up a development environment (Create React App), JSX, components (functional and class-based), props, and state.
    - **Goal:** Understand the component-based architecture of React.
- **Week 3-4: Advanced React Concepts**
    - **Topics:** React Router for page navigation, hooks (useState, useEffect, useContext), handling forms, and making API calls from a React app.
    - **Goal:** Build single-page applications (SPAs) that are fast and interactive.
    - **Project:** Rebuild your portfolio website as a React application. Then, build a more complex project like a movie search app that uses a public API (e.g., The Movie Database API).

Month 4: Building the Back-End Server

Time to move to the server-side. You'll learn how to build the "brain" of your application.

- **Week 1-2: Node.js & Express.js Fundamentals**
    - **Topics:** Introduction to Node.js, the npm ecosystem, setting up a web server, and understanding the basics of the Express.js framework.
    - **Goal:** Understand how the back-end works and create a simple server.
- **Week 3-4: Building a RESTful API**
    - **Topics:** Learn about REST architecture. Create routes for handling different HTTP requests (GET, POST, PUT, DELETE). Understand middleware and error handling.
    - **Goal:** Design and build a complete API that your front-end can communicate with.
    - **Project:** Build a REST API for a simple blogging platform. It should have endpoints to get all posts, get a single post, create a new post, and delete a post. Use a tool like Postman or Insomnia to test your API endpoints.

Month 5: Databases and Data Persistence

Your application needs to store data. This month, you'll learn how to connect and manage databases.

- **Week 1-2: Databases (SQL vs. NoSQL) & MongoDB**
    - **Topics:** Understand the difference between relational (SQL) and non-relational (NoSQL) databases. Dive deep into MongoDB: schemas, models, and querying data. Use Mongoose, an ODM (Object Data Modeling) library, to connect your Express app to MongoDB.

    - **Goal:** Integrate a database with your back-end server.
    - **Project:** Connect the blog API you built in Month 4 to a MongoDB database. Now, your posts can be saved, retrieved, and deleted from a real database.
- **Week 3-4: User Authentication & Security**
    - **Topics:** Learn how to implement user registration and login. Understand password hashing (using bcrypt) and JSON Web Tokens (JWT) for securing your API.
    - **Goal:** Add secure user authentication to your applications.
    - **Project:** Add user authentication to your blog API. Only logged-in users should be able to create new posts.

Month 6: Tying It All Together & Deployment

In the final month, you'll connect your front-end and back-end, and learn how to deploy your application to the web.

- **Week 1-3: Building a Full-Stack Capstone Project**
    - **Goal:** Create a complete MERN stack application from scratch. This is your chance to build a substantial project for your portfolio.
    - **Project Ideas:**
        - An e-commerce site with products, a shopping cart, and user accounts.
        - A social media application where users can create profiles, make posts, and comment.
        - A project management tool with task boards and user assignments.
    - **Process:** Design the API, build the back-end, then build the React front-end that consumes your API.
- **Week 4: Deployment & Final Touches**
    - **Topics:** Learn how to deploy your application. You can deploy your Node.js back-end to services like **Render** or **Heroku**, and your React front-end to **Netlify** or **Vercel**. Learn about environment variables.
    - **Goal:** Get your capstone project live on the internet!
    - **Final Steps:** Clean up your GitHub profile, update your resume, and start preparing for interviews.