4-Month Frontend Web Development Mentorship Program

Month 1: Foundations

Week 1: HTML & CSS Basics

- **Learn**:
    - ✓ What is HTML?
    - ✓ Understanding HTML tags (headings, paragraphs, links, images).
    - ✓ What is CSS?
    - ✓ CSS selectors, properties, and how to apply styles.
    - ✓ The box model: margins, borders, padding.
- **Practice**:
    - Build a static webpage with:
        - ✓ A header section (title and navigation).
        - ✓ A main content area (text and images).
        - ✓ A footer (contact info or copyright).

Week 2: Introduction to Git

- **Learn**:
    - ✓ What is Git and why use it?
    - ✓ Basic commands: clone, commit, push, and pull.
    - ✓ Understanding repositories and version control.
- **Practice**:
    - ✓ Clone an existing GitHub repository.
    - ✓ Make changes to a file, commit those changes, and push them back to GitHub.

Week 3: Responsive Design

- **Learn**:
    - ✓ What is responsive design?
    - ✓ Introduction to media queries.
    - ✓ Mobile-first design principles.

- ➢ **Practice**:
    - ✓ Modify your static webpage to:
        - ▪ Adjust layout for different screen sizes (desktop, tablet, mobile).

Week 4: JavaScript Basics

- ➢ **Learn**:
    - ✓ Introduction to JavaScript: variables, data types, loops, and conditionals.
    - ✓ How to manipulate the DOM (Document Object Model).
- ➢ **Practice**:
    - ✓ Add interactivity to your webpage:
        - ▪ Create a button that toggles content visibility.
        - ▪ Use JavaScript to change text or styles on your page.

Month 2: Intermediate Skills

Week 5: Advanced JavaScript

- ➢ **Learn**:
    - ✓ ES6+ features: arrow functions, template literals, and destructuring.
- ➢ **Practice**:
    - ✓ Build a simple calculator app using these features:
        - ▪ Implement basic operations (add, subtract, multiply, divide).

Week 6: JavaScript Events & Form Validation

- ➢ **Learn**:
    - ✓ How to handle events (clicks, form submissions).
    - ✓ Basics of form validation (checking input data).
- ➢ **Practice**:
    - ✓ Create a registration form:
        - ▪ Validate user input (e.g., email format, required fields).
        - ▪ Display error messages for incorrect inputs.

Week 7: Fetch API

- **Learn**:
  - ✓ What is the Fetch API?
  - ✓ How to make network requests to retrieve data.
- **Practice**:
  - ✓ Build a weather app:
    - Use a public API to fetch weather data.
    - Display the current weather for a specified location.

Week 8: CSS Grid & Flexbox

- **Learn**:
  - ✓ Introduction to CSS Grid and Flexbox for layout design.
  - ✓ Differences between Grid and Flexbox and when to use each.
- **Practice**:
  - ✓ Design a responsive dashboard layout using Grid and Flexbox:
    - Create sections for a sidebar, main content, and footer.

Month 3: Frameworks & Tools

Week 9: Introduction to React

- **Learn**:
  - ✓ What is React?
  - ✓ Understanding components and JSX (JavaScript XML).
- **Practice**:
  - ✓ Build a simple "Hello, World!" app in React:
    - Set up a React environment using Create React App.

Week 10: Props & State in React

- **Learn**:
  - ✓ What are props and state in React?
  - ✓ How to pass data between components.

- **Practice**:
    - ✓ Create a to-do app:
    - ✓ Allow users to add and remove tasks using state.

Week 11: React Router

- **Learn**:
    - ✓ What is React Router?
    - ✓ How to implement navigation in single-page applications (SPAs).
- **Practice**:
    - ✓ Add multi-page functionality to your to-do app:
        - ▪ Create different routes for viewing, adding, and editing tasks.

Week 12: Styling in React

- **Learn**:
    - ✓ How to style React components (CSS-in-JS, Tailwind CSS).
- **Practice**:
    - ✓ Style your to-do app:
        - ▪ Use a design framework or create custom styles.

Month 4: Advanced Topics & Final Project

Week 13: Advanced React Concepts

- **Learn**:
    - ✓ What is the Context API or Redux?
    - ✓ Managing global state in a React application.
- **Practice**:
    - ✓ Implement global state management in your to-do app:
        - ▪ Allow shared state across multiple components.

Week 14: Deployment

- **Learn**:
    - ✓ How to deploy web applications using services like Netlify or Vercel.

- **Practice**:
    - ✓ Deploy your to-do app online:
        - ▪ Share the live link with others.

Week 15: Capstone Project Planning

- **Start**:
    - ✓ Plan your capstone project (e.g., an e-commerce app).
    - ✓ Outline features and functionalities you want to implement.
- **Practice**:
    - ✓ Begin development:
        - ▪ Implement dynamic features and API integrations.

Week 16: Finalize & Present

- **Finish**:
    - ✓ Complete your capstone project.
    - ✓ Prepare for presentation.
- **Present**:
    - ✓ Share your project with your mentor for feedback and review.

Key Deliverables

1. **Week 4**: A simple, responsive webpage with JavaScript interactivity.
2. **Week 7**: A weather app that fetches data from an API.
3. **Week 12**: A styled to-do app.
4. **Week 16**: A completed project shared online.

Tools to Learn Along the Way

1. **Code Editors**: Use Visual Studio Code.
2. **Free Resources**: MDN Web Docs, freeCodeCamp, or YouTube tutorials.
3. **Browser Developer Tools**: Inspect and debug your code.