WEEKLY PROGRESS REPORT

SUBMITTED BY:

Name: **Mehtabbir Singh**

Roll No: **21103091**

**TABLE OF CONTENT**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **Title** | **Duration** | **Page No.** |
|  | Week 1 | 03/06/2024 – 8/06/2024 | 2 |
|  | Week 2 | 10/06/2024 – 15/06/2024 | 3 |
|  | Week 3 | 17/06/2024 – 22/06/2024 | 4 |
|  | Week 4 | 24/07/2024 – 29/07/2024 | 5 |
|  | Week 5 | 1/07/2024 – 06/07/2024 | 6 |
|  | Week 6 | 08/07/2024 – 13/07/2024 | 7 |
|  | Week 7 | 15/07/2024 - 20/07/2024 | 8 |

### ****Week 1:****

### ****Learning HTML, CSS, JavaScript, and ECMAScript****

(03/06/2024 – 8/06/2024)

#### ****Activities:****

* Studied the fundamentals of **HTML** and **CSS**, focusing on structuring and styling web pages.
* Gained hands-on experience in creating basic web pages, using elements like divs, links, forms, and tables.
* The primary focus was on **JavaScript** and **ECMAScript** (ES6+), including topics like variable declarations (let, const, var), functions, arrow functions, template literals, and modules.
* Implemented small interactive features like form validation, DOM manipulation, and event handling using JavaScript.

#### ****Achievements:****

* Completed hands-on exercises and mini-projects that helped solidify knowledge of web development basics.
* Built simple dynamic web pages with interactive features like image sliders and pop-up windows using JavaScript.
* Gained proficiency in using **JavaScript ES6** features such as destructuring, spread operators, and promises.

#### ****Reflection:****

* This week helped me build a strong foundation in **frontend development**. I realized how important a solid understanding of HTML, CSS, and JavaScript is before diving into complex frameworks.
* Learning **ECMAScript** was an eye-opener, as it allowed me to write cleaner and more maintainable code. I was excited to see how these foundational technologies work together to form a seamless web experience.

### ****Week 2:****

### ****Learning React****

(10/06/2024 – 15/06/2024)

#### ****Activities:****

* Focused on **React.js**, learning about **components**, **props**, **state**, and **JSX**.
* Created several small React applications to practice concepts like functional components, hooks (useState, useEffect), and event handling.
* Explored **React Router** for handling navigation between different components in a single-page application (SPA).
* Integrated external APIs with React and worked on displaying dynamic data using **useEffect** and **useState** hooks.

#### ****Achievements:****

* Built a simple to-do list application that allowed users to add and remove tasks, leveraging React’s state and props for data flow.
* Successfully set up routing in a React app to navigate between different pages without reloading the whole page.
* Gained a deeper understanding of the **React lifecycle** and how state and props work together for data management.

#### ****Reflection:****

* React significantly enhanced my ability to build dynamic and responsive user interfaces. I found the component-based structure and the concept of **state** to be powerful tools for managing data within an application.
* By the end of the week, I felt confident using React to build and manage the flow of data within a UI. It was a crucial step forward in my development as a frontend developer, and I started to see how this framework could help in the development of larger projects.

### ****Week 3:****

### ****Learning Node.js****

#### **(17/06/2024 – 22/06/2024)**

#### ****Activities:****

* Focused on **Node.js**, learning about its role as a server-side runtime for JavaScript.
* Gained experience in setting up a **Node.js server**, creating **RESTful APIs**, and handling **HTTP requests** (GET, POST, PUT, DELETE).
* Worked with **Express.js**, a popular framework for building backend applications, learning to create routes, middlewares, and manage requests.
* Connected Node.js with a local **MongoDB** database to store and retrieve data.

#### ****Achievements:****

* Built a simple API to manage a list of users, allowing for CRUD (Create, Read, Update, Delete) operations.
* Gained hands-on experience with **Express.js routing**, and how to handle asynchronous tasks using **Promises** and **async/await**.
* Successfully set up **MongoDB** for data storage and integrated it with the backend to persist and retrieve data.

#### ****Reflection:****

* This week marked a key milestone in my journey as a full-stack developer. I learned how backend technologies like **Node.js** and **Express.js** power web applications by handling business logic and managing databases.
* While setting up a basic API and connecting to a database was challenging, it gave me a better understanding of how the backend communicates with the frontend. I became more confident in my ability to handle both the server-side and client-side aspects of a web application.

### ****Week 4:****

### ****Setting Up the Project Structure, Backend, and Initial Routes****

(24/07/2024 – 29/07/2024)

#### ****Activities:****

* **Project Initialization**: Set up the folder structure for both frontend and backend. Initiated the **React.js** frontend and **Node.js** backend repositories.
* **Database Setup**: Configured **MongoDB** as the database solution and established initial connection with the backend using **Mongoose**.
* **Node.js Backend**: Set up the basic **Node.js/Express.js** server. Created initial routes for user registration and login functionalities.
* **Backend API Setup**: Designed **API routes** for basic CRUD operations on the database (e.g., creating and managing users and craftsmen profiles).
* **User Authentication**: Implemented **JWT (JSON Web Tokens)** for secure user authentication and created the login and registration system.

#### ****Achievements:****

* Successfully initialized the **backend** and **frontend** project structure.
* Established **MongoDB** as the data store and connected it to the backend using **Mongoose**.
* Implemented **user authentication** using **JWT** and created routes for basic user registration and login.

#### ****Reflection:****

* The first week of the project was about setting up the necessary tools and creating the foundation for both the frontend and backend.
* I learned the importance of organizing the project structure efficiently and the challenges of connecting the backend with the database.
* Setting up **JWT-based authentication** was a valuable experience in securing the application and ensuring that only authorized users can access certain resources.

### ****Week 5:****

### ****Frontend Component Creation, API Integration, and Routing****

(1/07/2024 – 06/07/2024)

#### ****Activities:****

* **React Component Setup**: Started building core **React components** such as **Cart**, **Form**, and **Home, Product, products, sidebar** for displaying relevant data.
* **State Management**: Managed state using **React hooks** like **useState** and **useEffect** to control dynamic content such as displaying craftsman profiles and handling user input.
* **API Integration**: Integrated **API calls** from the backend to the frontend, fetching data for craftsman profiles, services, and user bookings.
* **Routing**: Set up **React Router** for navigating between pages like the Home page, craftsman profile pages, and user dashboards.
* **Form Handling**: Implemented forms for user registration, craftsman profile creation, and service requests, ensuring data was properly validated before submission.

#### ****Achievements:****

* Successfully created key **React components** for user and craftsman profiles and integrated them with the backend.
* Completed the frontend’s dynamic data handling by fetching craftsman profiles and services from the backend.
* Implemented **React Router** for smooth navigation between pages without reloading, providing a better user experience.

#### ****Reflection:****

* This week was exciting because I could finally see the frontend and backend working together. **React components** allowed me to structure the application logically and display real-time data dynamically.
* Setting up **React Router** was crucial in turning the application into a **single-page application (SPA)**, improving navigation and performance.
* **API integration** helped me understand how to connect the frontend to the backend effectively, enabling real-time updates in the app.

### ****Week 6:****

### ****Backend Enhancements, Profile Management, and Connecting Components****

(08/07/2024 – 13/07/2024)

#### ****Activities:****

* **Enhancing Backend Functionality**: Continued adding additional backend routes, including those for managing craftsman profiles, editing user details, and creating service requests.
* **Database Schema Improvements**: Expanded the **MongoDB schema** to include new collections for user bookings, reviews, and service requests.
* **Frontend-Building Components**: Built and styled more **React components**, such as **ReviewSystem**, **BookingForm**, and **SearchBar**, which would allow users to review craftsmen and search for specific services.
* **Connecting Components**: Ensured seamless **data flow** between **React components** and backend APIs, allowing real-time interaction such as booking services, updating profiles, and leaving reviews.
* **User Interactions**: Implemented **user interactions** such as submitting a service request, updating profile information, and leaving reviews for completed services.

#### ****Achievements:****

* Developed key features, including **Search**, **Product**, and **Sorting on Backend**.
* Created an intuitive UI for managing **products** and allowed users to interact with the platform easily.
* Connected **React components** to the backend, ensuring seamless functionality across user and craftsman data.

#### ****Reflection:****

* The focus of this week was on enhancing the platform’s usability and functionality by adding features such as profile management and service bookings.
* I realized how critical it is to ensure smooth data flow between frontend components and backend APIs for a real-time user experience. Managing this interaction seamlessly improved my understanding of full-stack development.
* By the end of this week, the core features of **Shree Balaji Spareparts** were functional, and I could see the application taking shape with active features like bookings.

****Week 7: Authorization, Google Sign-In Integration, Bug Fixes, and Cart Page Completion****

**(05/11/2024 – 10/11/2024)**

#### ****Activities:****

* **Authorization Implementation**: Completed user authentication system by refining the login and registration flow. Integrated Google Sign-In for easier access and enhanced user convenience.
* **Google Sign-In Integration**: Implemented Google Sign-In on the frontend using OAuth 2.0, enabling users to log in with their Google accounts. Configured the backend to authenticate and handle Google Sign-In tokens securely.
* **Bug Fixes**: Resolved critical bugs related to the user authentication flow and UI responsiveness. Debugged issues with product pages not loading correctly and fixed payment gateway integration errors.
* **Cart Page Completion:** Developed the cart page for the e-commerce website, ensuring it displays products, quantities, and prices. Implemented the ability to add/remove items from the cart and calculate total prices dynamically. Integrated the cart with the backend to save the cart data across sessions.

#### ****Achievements:****

* Successfully implemented **Google Sign-In** for user **authenticatio**n, improving user experience and simplifying login.
* Fixed several bugs that were affecting the site's functionality, ensuring smoother **navigation and transactions.**
* Completed the cart page functionality, which is now fully operational with features like a**dding/removing items** and calculating totals.

#### ****Reflection:****

* This week focused on improving the user experience by adding Google Sign-In and resolving key bugs.
* Integrating Google Sign-In was a great learning experience, particularly in working with OAuth 2.0 for secure authentication.
* The cart page completion helped solidify my understanding of dynamic frontend development and backend integration, as I learned how to handle real-time data updates on the frontend while ensuring consistency across sessions.