

Frontend Development Report

Date	21-07-2024
Team ID	SWTID1720162063
Project Name	Cab Booking App
Maximum Marks	

Project Title: [Cab Booking Application]

Date: [21-07-2024]

Prepared by: [Cab Booking App Team]

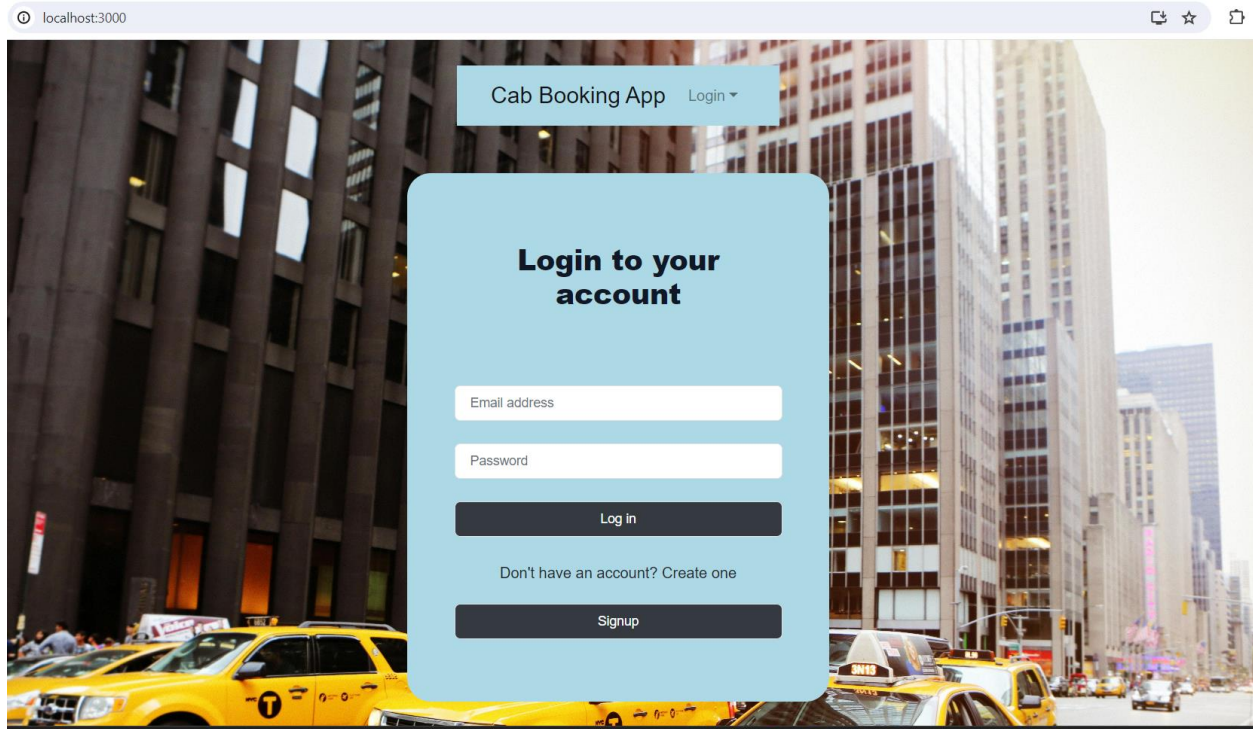
Objective

The objective of this report is to document the frontend development progress and key aspects of the user interface implementation for the Cab Booking App project. This project aims to provide users with a seamless experience for booking rides, managing their bookings, and interacting with the platform. The focus is on creating a responsive, user-friendly interface using modern web development technologies.

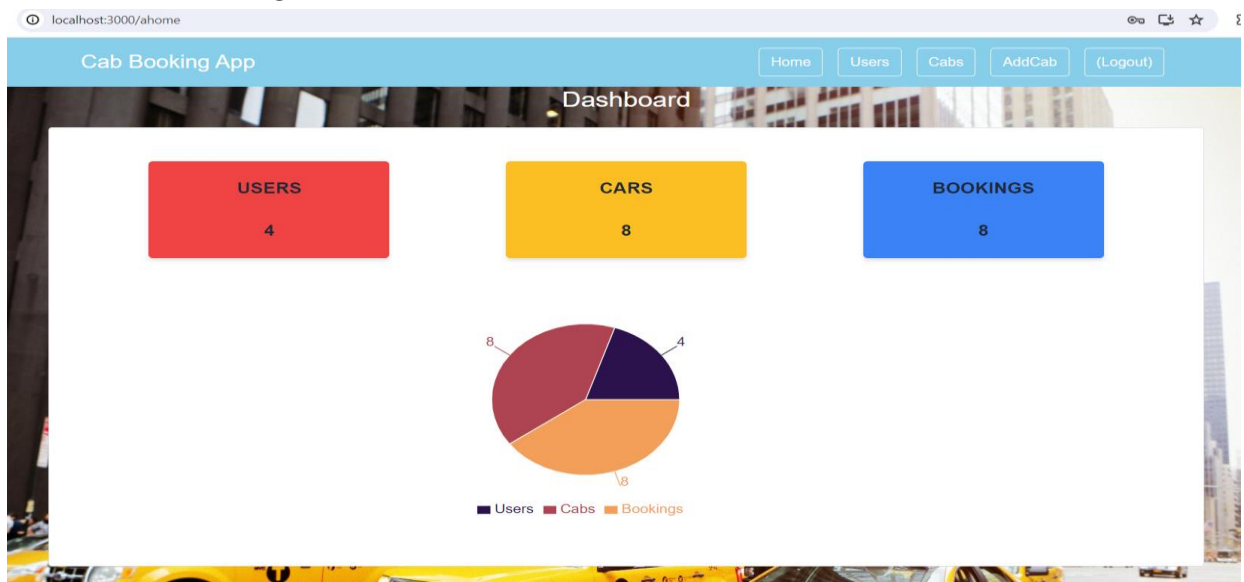
Technologies Used

- **Frontend Framework:** React.js
- **State Management:** Context API
- **UI Framework/Libraries:** Bootstrap
- **API Libraries:** Axios
- **Languages and Tools:** HTML, CSS, JavaScript, Node.js, MongoDB, Express

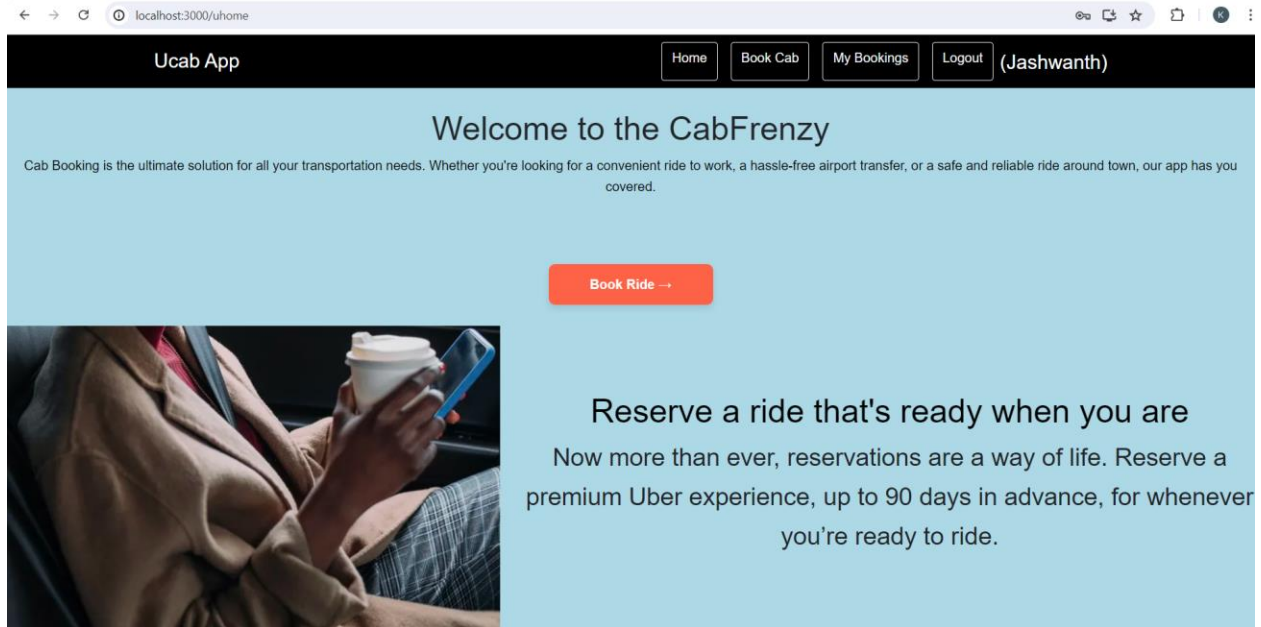
Project Structure



Admin HomePage:



User HomePage:



- **frontend/public:** This directory contains the static files and assets such as the HTML template and images.
- **frontend/src:** The main source directory for the React application.
 - **components:** Contains reusable UI components used across the application.
 - **pages:** Includes different pages for the web app, each representing a different route.
 - **App.js:** Responsible for routing and the main application layout.
 - **index.js:** Entry point for the React application.

Key Components

App.js

App.js is responsible for routing and the main application layout. It includes the setup for React Router and defines the main routes for the application.

/components

The components directory contains reusable UI components used across the application. These components are designed to be modular and can be easily integrated into different parts of the application.

/pages

The pages directory includes different pages for the web app, each corresponding to a specific route. Examples of pages might include the home page, dashboard, and user profile page.

Routing

Routing is managed using React Router. Here are the main routes:

- **/home**: Landing page of the application.
- **/dashboard**: Dashboard displaying user data and statistics.
- **/profile**: User profile management.

State Management

State management is achieved using Context API. This allows for managing global state across the application and provides a way to share data between components without having to pass props manually at every level.

Integration with Backend

The frontend communicates with the backend APIs hosted on <http://backend-url>. Key endpoints include:

- **GET /api/data**: Retrieves data for display.
- **POST /api/user/login**: Handles user authentication.

User Interface (UI) Design

The UI design follows a user-centric approach, prioritizing usability and accessibility. The design principles include:

- **Consistency**: Ensuring a uniform look and feel across the application.
- **Responsive Design**: Making the application accessible on various devices and screen sizes.
- **Simplicity**: Keeping the interface clean and intuitive to enhance user experience.

Implemented using Bootstrap, the UI leverages pre-designed components and styles to streamline development and maintain visual coherence.

Third-Party Integrations (If any)

- **Axios:** Used for making API requests.
- **React Router:** Used for routing within the React application.