**Ride Sharing Application**

**Name : Aditya Jaiswal**

**Date : 7/8/24**

**GitHub:**

[**https://github.com/Aditya-Jaiswal-534/RideConnect/tree/master**](https://github.com/Aditya-Jaiswal-534/RideConnect/tree/master)

**Table of Contents**

**Summary**

* Overview of the project
* Key objectives and outcomes

**Introduction**

* Background and context
* Purpose of the project
* Objectives

**Technologies Used**

* Frontend
* Backend
* Database
* Tools and Libraries

**Implementation**

* Project Setup
* Frontend Development
* Backend Development
* Database Design
* Key Features and Functionalities

**Challenges and Solutions**

* Challenges faced during development
* Solutions implemented

**Future Enhancements**

* Scope of improvements
* Better, secure, and efficient methods

**Conclusion**

### **Summary**

**RideConnect** is a ride-sharing platform designed to offer a seamless and user-friendly experience for travelers. The project focuses on providing a robust solution for ride-sharing, integrating essential features such as user authentication, ride management, feedback system, and real-time notifications via SMS. The key objectives of this project include:

1. **User Authentication**: Ensuring secure access to the platform by implementing a comprehensive user authentication system. This includes login, registration, and secure session management to protect user data and ensure authorized access to the platform's features.
2. **Ride History Management**: Allowing users to view their ride history, providing them with a detailed list of all the rides they have taken. This feature helps users keep track of their past journeys and manage their travel plans effectively.
3. **Ride Creation**: Enabling users to create new rides effortlessly. Users can enter ride details such as driver name, phone number, cab number, source, and destination. This feature streamlines the process of setting up new rides and makes it convenient for users to manage their travel plans.
4. **Feedback System**: Allowing users to provide feedback on their rides. This feature encourages users to share their experiences and helps improve the overall service quality by gathering valuable user insights.
5. **Real-time Notifications**: Facilitating real-time communication by allowing users to share trip details with their travel companions via SMS. This ensures that travel companions are promptly informed about the trip details, enhancing coordination and safety during travel.

### **Introduction :**

#### **1. Project Overview**

RideConnect is a ride-sharing application designed to provide users with a seamless experience in managing and sharing their travel details. The project employs the MERN stack, consisting of MongoDB, Express.js, React.js, and Node.js. This report outlines the key features of TravelMate, including user authentication, ride management, feedback submission, and SMS notifications for trip sharing.

#### **2. Objectives**

* Implement user authentication to secure access to the application.
* Allow users to view their previous rides.
* Enable users to create new rides.
* Provide functionality for users to submit feedback.
* Facilitate sharing of trip details via SMS.

#### **3. Technologies Used**

* **Frontend:** React.js (with Vite for setup), TailwindCSS for styling
* **Backend:** Node.js, Express.js
* **Database:** MongoDB
* **SMS Service:** Twilio (for sending SMS notifications)
* **HTTP Requests:** Fetch API

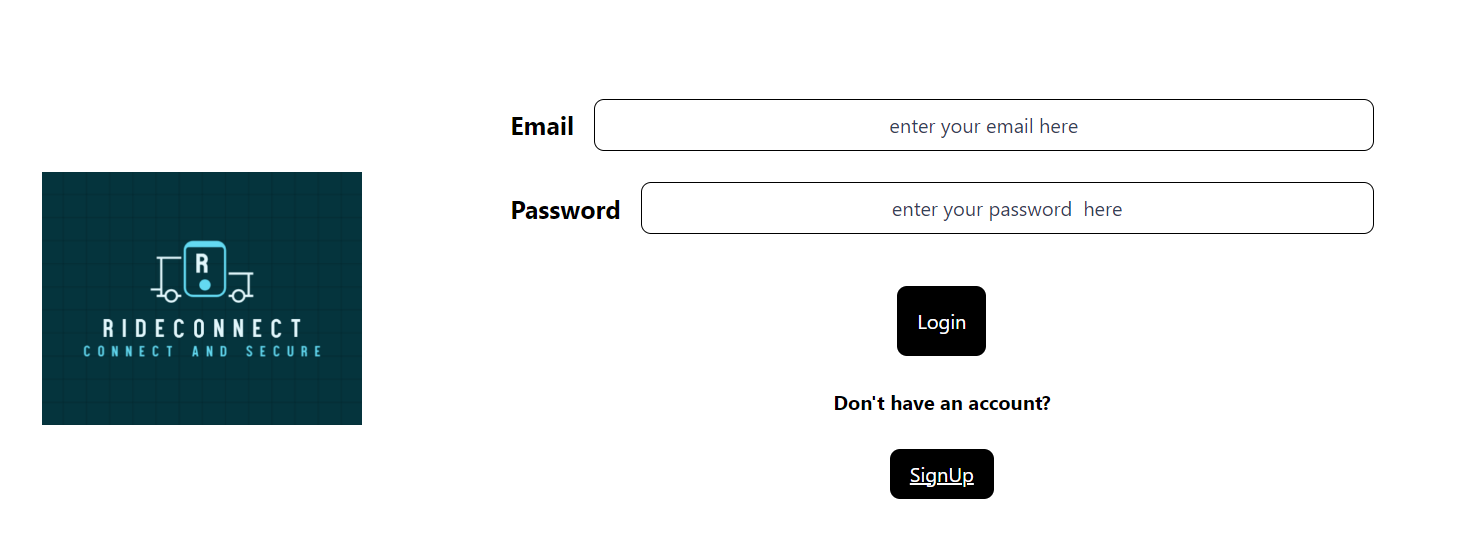
**Implementation :**

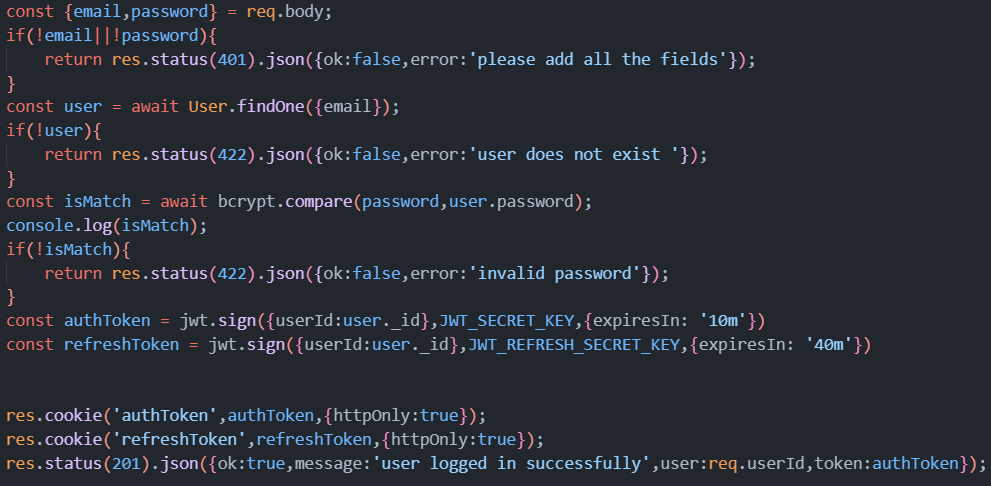
4.1. **User Authentication**

* Description: Users must authenticate themselves to access certain features of the application.

How It Works:

* User credentials are sent to the /auth/login endpoint.
* A JWT (JSON Web Token) is issued upon successful login and is used to authenticate subsequent requests.
* The token is stored in cookies for maintaining session security.

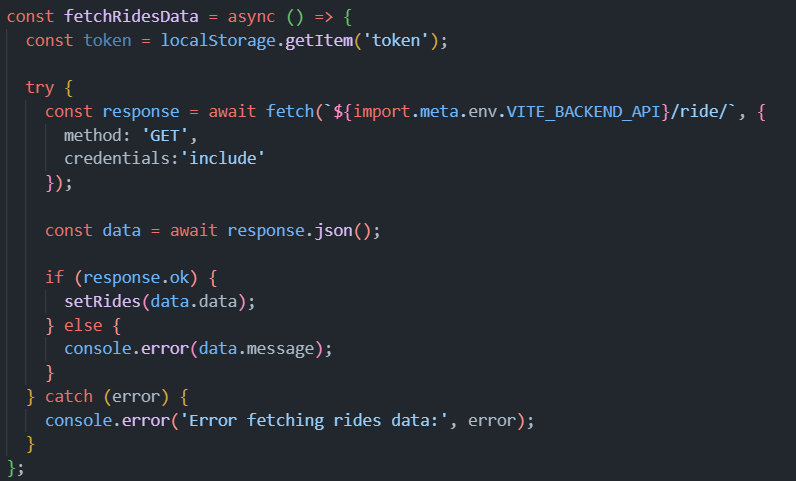
****

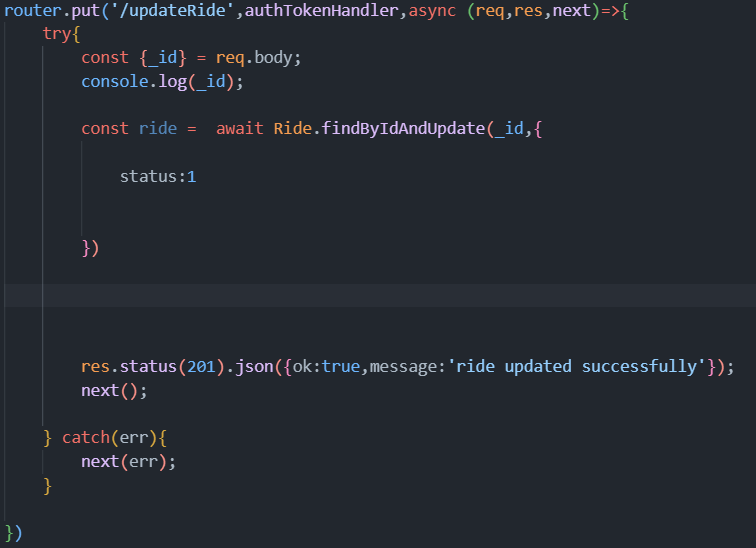
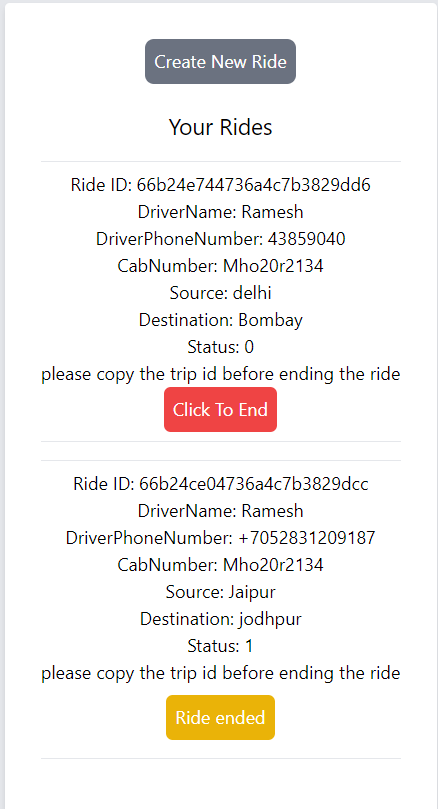
****

**4.2. Viewing Previous Rides**

* Description: Users can view a list of their past rides.

How It Works:

* On accessing the dashboard, a GET request is made to the /ride/ endpoint to fetch the user’s ride data.
* The data is then displayed in a structured format on the dashboard.
* After you reach destination you can manually end the ride
* Status 1 means the ride has been completed

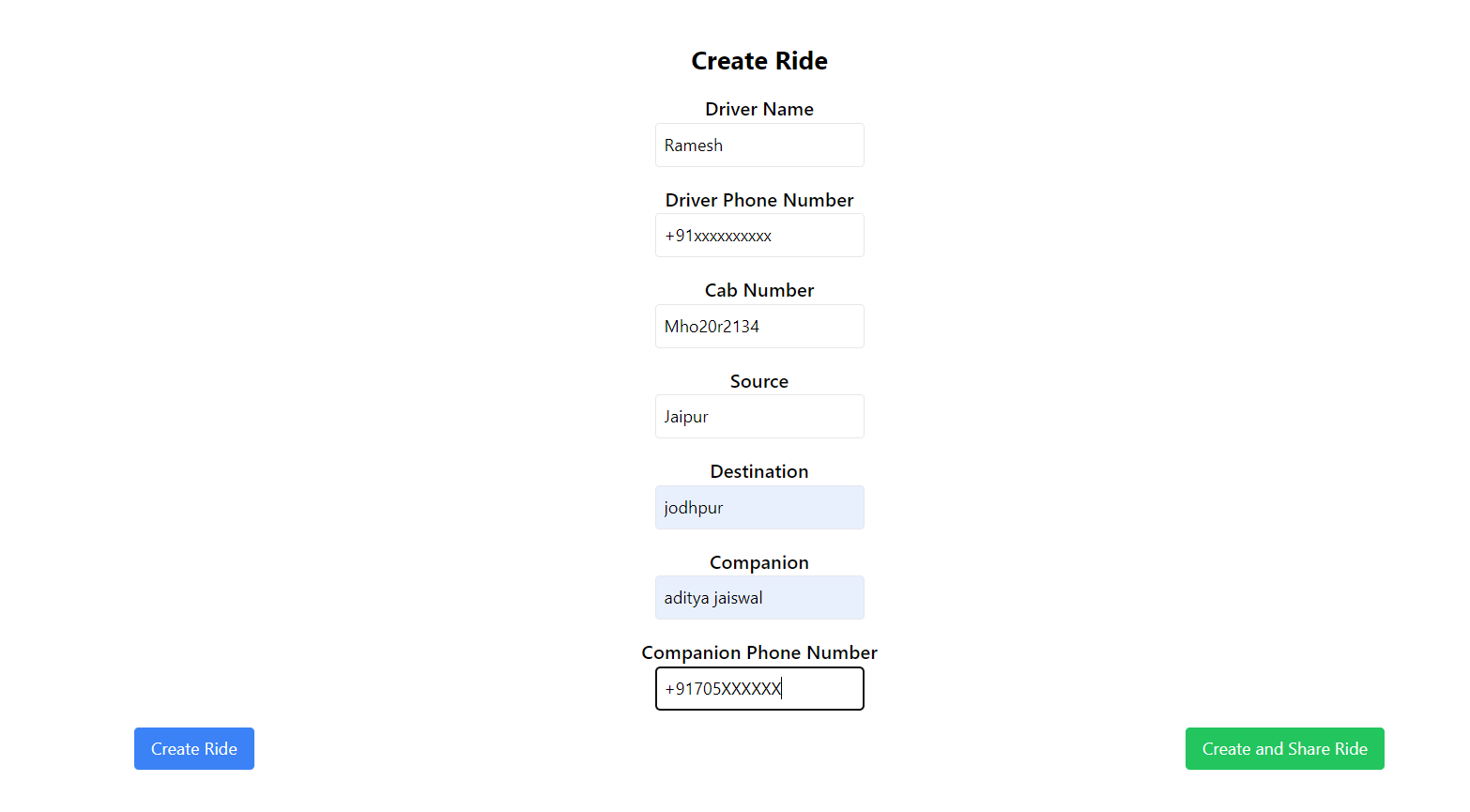


##### **4.3. Creating New Rides**

**Description:** Users can create new ride entries by providing ride details through a form.

**How It Works:**

* + Ride details are sent to the ride/createRide endpoint.
  + The ride is saved in the database, and a confirmation is provided to the user.

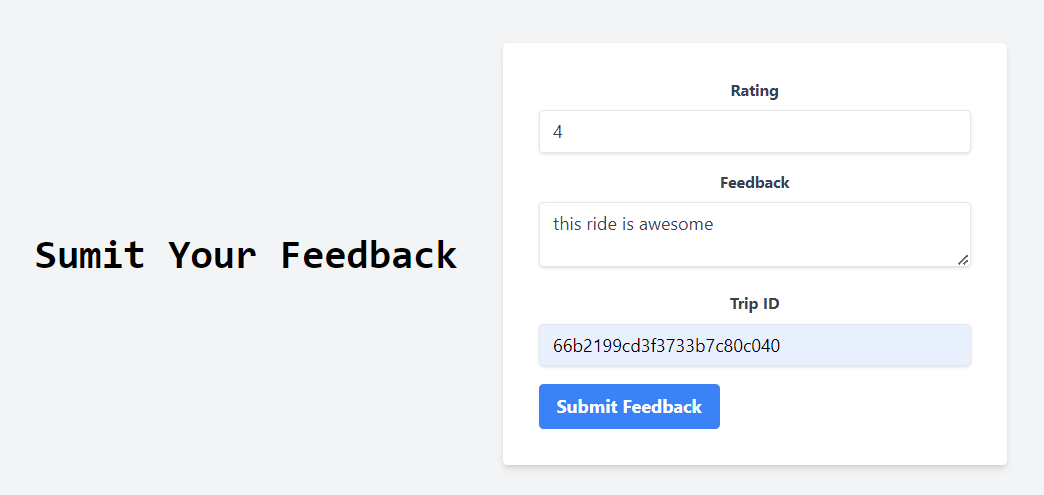


##### **4.4. Submitting Feedback**

**Description:** Users can provide feedback on their rides.

**How It Works:**

* + Feedback data is sent to the feedback/createFeedback endpoint.
  + The feedback is saved in the database and linked to the specific ride.
  + Once you click end ride you will automatically be taken to feedback section for that ride

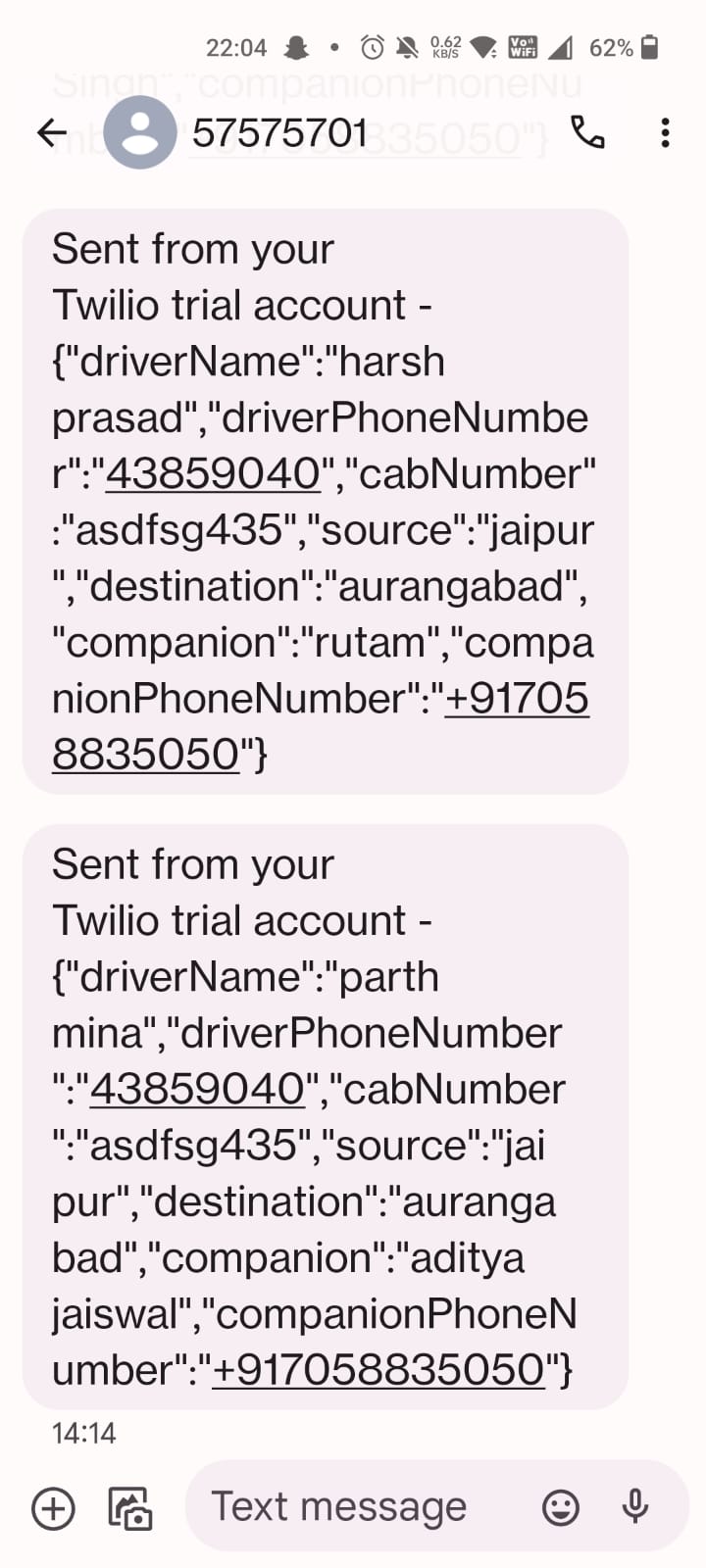


##### **4.5. Sharing Trip Details via SMS**

**Description:** Users can share their trip details with companions via SMS.

**How It Works:**

* + After creating a ride, users can choose to share the ride details.
  + A POST request is made to the /notification/send endpoint to send an SMS to the companion’s phone number using Twilio’s API.



##### **4.6. Admin Backend Functionalities Implemented :**

##### **Description:** most of the remaining Functionalities of the Admin are implemented in the backend

##### **Functionalities:**

* + Admin can see all the ride details shared by the users by making the GET request to the specific EndPoint
  + It can also use filter by inserting the user id in URL so that only rides shared by particular user get displayed
  + All the things mentioned about ride also apply to feedbacks in case of admin

**5. Challenges Faced and Solutions :**

* **Error Handling :**   
    
  **Challenge**: During the development of RideConnect, a key challenge was effectively managing and conveying errors to both developers and users.   
    
   **Solution**: To address this, a consistent JSON response format was implemented, including fields for ‘ok,’’message’, and ’data’, which streamlined error handling and debugging. Additionally, Toast notifications were used to provide clear, real-time feedback to users, enhancing their overall experience. This structured approach improved error clarity, facilitated debugging, and ensured users were promptly informed of issues.

### **Handling Asynchronous Operations**

**Challenge**: Managing asynchronous operations, such as fetching data from APIs or handling user input, lead to issues like race conditions or inconsistent state.

**Solution**: Employing async/await syntax and proper error handling ensures smoother asynchronous operations. Implementing loading states and handling errors gracefully which improved the user experience.

#### **6. Scope of Improvements**

* **Enhance User Interface:** Improve the user interface with better styling and more intuitive navigation for ex- Navbar , sidebar, profile section,etc;
* **Additional Features:** Consider adding features like generating link for the ride and real-time ride tracking and also we can integrate Map for better visualization of the riders location, online Payment system , etc
* **Performance Optimization:** can use better and efficient methods to reduce the time complexity of several processes.

**7. Conclusion**

**RideConnect** serves as a solid foundation for a ride-sharing application, effectively combining essential features with modern technology to address user needs. The project successfully integrates user authentication, ride management, feedback collection, and trip sharing into a cohesive platform. While it provides a robust starting point, there is ample opportunity for growth and refinement.

Future development should focus on enhancing security measures, optimizing performance, and expanding features to create an even more engaging and secure user experience. With these improvements, **RideConnect** has the potential to become a leading solution in the ride-sharing domain, offering both reliability and innovation to its users.