**Requirements**

**User Authentication and Profiles:**

* Allow users to create accounts using their university email addresses.
* Build user profiles with information such as car details, commute schedule, and preferences.

**Ride Listings:**

* Enable users to create ride listings specifying details like departure time, pickup points, and available seats.
* Implement a search and filter functionality for users to find rides based on their preferences.

**Ride Requests and Confirmations:**

* Allow users to send ride requests to drivers.
* Implement a notification system to inform users about ride requests and confirmations.

**Route Optimization:**

* Integrate a mapping service to help users optimize their routes and find the most efficient carpooling options.

**Real-Time Location Tracking:**

* Integrate a real-time location tracking feature to allow users to track the location of the vehicle they are carpooling with.

**Messaging System:**

* Include a messaging system for users to communicate with each other to coordinate pickup points, drop-off locations, and other details.

**Rating and Review System:**

* Implement a rating and review system for both drivers and passengers to build trust and ensure a reliable carpooling experience.

**Notification System:**

* Send automatic reminders for upcoming rides, ride requests, and confirmations.

**Pages**

**Homepage:**

* Welcome message and brief introduction to the platform.
* Quick access to login or register.

**User Pages:**

* User Profile Page (once logged in)
* Ratings and Reviews Page (showing feedback received by users) \*Can be incorporated in the user profile page

**Authentication Pages:**

* Login Page
* Registration Page

**Ride Management Pages:**

* Create Ride Page (form for specifying ride details)
* Create a Ride Request(for passengers)
* Ride Listings Page for passengers (displaying available rides with filters)
* My Rides(for Drivers)(Incorporate Remove Passengers in it)
* Notifications for drivers (Add Passengers)
* Ride Details Page (detailed information about a specific ride) \*Can be a modal too

**Messaging Pages:**

\*Both can be a single page

* Inbox Page (listing user messages)
* Individual Chat Page (for messaging between users)

**Rating and Review Pages:**

* Leave a Review Page (for users to leave feedback after a ride) \*Can be a modal too

**Notification Pages:** \*Not doing for now

* Notification Center Page (listing all notifications)

**Real-Time Location Tracking Page:**

* Map Page (displaying real-time location tracking during a ride)

**Settings and User Management Pages:** (Not implementing for now)

* User Settings Page (for managing account settings)
* Manage Rides Page (for drivers to view and manage their posted rides)

**Useful Tools**

**Authentication:**

* Use JSON Web Tokens (JWT) for secure authentication.
* Passport.js is a popular authentication middleware for Node.js.

**Mapping and Route Optimization:**

* Integrate the Google Maps API for mapping, geocoding, and route optimization.

**Real-Time Updates:**

* Use Socket.IO for real-time communication between users.

**Messaging:**

* Implement a messaging system using a real-time database like Firebase Realtime Database.