# RIDEIT

**~ by Samyak Jain (2021560) and Shrey (2021562)**

## PROJECT SCOPE

Our project deals with the booking system of cabs. Here our main aim is to develop a good database for our cab booking application; this would help the users and people involved in different departments to store and use the data efficiently, conveniently, and securely.

In our project, many stakeholders are involved, mainly admins, customers (users), and drivers. All of them have to log in to their accounts. Every stakeholder has their unique id (which acts as the primary key for their entity), like customers have customer_id, drivers have driver_id, etc. They also have names and contacts, which are the same for all the stakeholders. Specifically for the customer, have their ride_history, feedbacks_receieved, feedbacks_given, etc. For drivers they have vehicle_driven, feedback_given, feedback_received, current_rating, etc. Here we have given the feedback functionality to both the user and the driver so that both of these stakeholders can give feedback to each other and rate it out of 5, and according to that, they will choose whether they want to ride with the other person or not.

A functionality of choosing the type of cab will be given to the user; the types could be SUV, luxury, compact, two-wheeler, etc., and according to that, the options will show up to the user so that options can be filtered between the types of the cab.

Here for a particular ride, we will assign a unique number (primary key of ride entity) to each ride so that when we store it as data, we can easily retrieve it (indexing). The ride entity will consist of attributes like ride_id (primary key), vehicle_id (foreign key), cust_id (foreign key), driver_id (foreign key), starting_location, destination, etc.

Here we will provide every customer a vehicle rental option, where they can rent their required vehicle for a while as needed.

There will be a membership scheme for the customer, which provides discounts on the rides and rents they take. In the customer entity, there will be an attribute that gives information about whether the customer has accepted the membership or not.

And lastly, every internal process (payments, car rental, support, etc.) is assumed to be handled by the admins.

## FUNCTIONAL REQUIREMENTS

For our application, we will provide the following functionalities, and this is how they will work.

**Customer registers for the application:** When using the application for the first time after installing it, the system asks whether he wants to register as a new account or log in as a user. If the user goes for registration, the system asks for his username, phone number, and password. The system then stores the information about the customer in the customer's data.

If the customer, however, goes for login, the system asks for his phone number and password and then checks whether the phone number is registered for a user or not. If the phone number is available, it checks whether the password provided is correct or not. If the username is unavailable or the password is wrong, the system will return the message "wrong credentials."

**Customer books a cab:** The main functionality on which the application is based on is the booking system of cabs by our client. For booking his cab, first, the customer enters its pickup location, and the system takes it in the customer's pickup_location attribute. The system scans around the area of the pickup location and shows the customer the availability of nearby cabs. Then, the system asks the customer for the destination location and takes it in the customer's dest_location attribute. Once this is entered, the system shows the price (Fare) of different types of cabs/services available in its area. The customer then chooses one of the services available, and then the data about the customer (like his destination, pickup location, and phone number) asking for cab availability is given to all the cab drivers with those services in the area. The system waits for a driver to accept the customer. Once a driver accepts it, the information about the driver and his vehicle is given to the customer (now a passenger for that driver).

**Customer rents a vehicle:** Along with cab booking, the customer can also rent a car, and according to the rental period, the charges will be applied.

**Driver rents the vehicle for running the cab:** The driver can also take the car on lease from the company for the cab driving purpose. For that, he needs to give the company his aadhar ID and other documents.

**Booking Flexibility for the customer:** Our application provides the user the time and location flexibility. They have the choice to book a cab for riding immediately or at a later time. At the same time, the booking will be irrespective of the location selected by the user; a driver will accept the cab booked by him only for that date, time, and location. Users who need to schedule the journey for a later date will get a pop-up window where they can select the date and the desired time.

**Flexibility in payment processing:** Our application will provide support to the customer to make payments through various ways, like credits card, net banking, UPI, or Cash-on-delivery.

**Feedback Functionality:** Here, we have given the feedback functionality to both the user and the driver so that both of these stakeholders can provide feedback to each other and rate it out of 5, and according to that, they will choose whether they want to ride with the other person or not.

**Memberships for the customer:** The application allows the customer to make rides cheaper and more convenient to book by providing membership cards to the customer. He has the choice to buy the memberships provided by us, which will be of two types:

- Default: This is the default status of the newly registered customer, who has not applied for any upgradation of his membership status

- Pro: This membership is available for purchase by the customer. The customer would give the monthly payment for having the membership. It will have the following features:

1. 50% off on the day's first ride (for up to 100 km).
2. 30% off on all other rides.

Also, the pro users are given preference over default membership users for cab bookings.

**Support feature for the user:** Here, users will be given a support system for resolving their queries, and the admins will handle this.

**Admin moderation**: The admin of our application will be able to see/get information about all the actions regarding the company, like all the bookings, the drivers and their vehicle information, the customers, etc.

## TECHNICAL REQUIREMENTS:

We will majorly have the following entities in our database (which will be made in MySQL):

- Customer
- Driver
- Admin
- Vehicle
- Ride
- Rental

The following are the access constraints:

- Admins can access all the necessary customer data; some data, like passwords, will not be accessible.
- Admins can access all the necessary driver data; some data, like passwords, will not be accessible.
- Admins can access the data of all the rides and rents taken.
- The customer who has booked the cab can access only the relevant data of the cab driver, such as his phone number and his vehicle info, but cannot access his ID number, and other sensitive information
- In the same way, the cab driver can only access the start location, dest location, and phone number of the customer, but not his personal information, who has booked the cab.
- All the passwords can only be accessed by their respective owners. This is ensured by the database using hashing to encrypt it at the instance they are created.

Additionally, we will also do the following to remove any anomaly during the time of system break-down to ensure customer convenience:

- Making use of logs and triggers, to ensure that all transactions happen in an atomic way and to ensure no hangovers during money transactions.
- If, for some reason, the customer forgets the password, he can retrieve it through his phone number. In all, the customer's phone number will be his prime identity.

## TECH STACK USED:

For the frontend, we will use the following:

- HTML
- CSS
- JavaScript

For the backend, we will use the following:

- Python
- Flask
- MySQL