FCCU Carpool Service

(FORMANRIDE)

A webapp for riders and drivers to share ride to FCCU

Senior Project

Primary Advisor: **Akheem Yousaf**
Secondary Advisor: **Dr. Aasia Khanum**

Presented by:

231485855	AFNAN AHMED
241547206	HATIB ZUBAIR
231485432	MUHAMMAD MUZAMMIL

Department of Computer Science

Forman Christian College (A Chartered University)

FCCU Carpool Service

By

AFNAN AHMED

HATIB ZUBAIR

MUHAMMAD MUZAMMIL

Project submitted to

Department of Computer Science,

Forman Christian College (A Chartered University),

Lahore, Pakistan.

in partial fulfillment of the requirements for the degree of

BACHELOR OF SCIENCE
IN
COMPUTER SCIENCE (Honors)

Primary Project Advisor

Secondary Project Advisor

Senior Project Management
Committee Representative

Abstract

In FCCU, the number of students keeps increasing every semester, increasing the use of parking spaces. The parking space is occupied most of the time, many students cannot park their cars, and the space is so confined that accidents are a usual occurrence. With FCCU not increasing the parking space anytime soon, our application provides a way to utilize the given parking space in a way that it can be used by everyone coming to FCCU. FORMAN RIDE is a unique app designed to help university students carpool with students in their area so that more parking space is available and fuel costs are saved for the students. The application uses visual studio, and Python is the coding language used. The application has a simple-to-use interface with different functionalities for the driver and rider. The driver will post a weekly schedule so riders can see it and decide whether to travel with it. The rider will decide with which driver they want to go and request a ride. It is upon the driver to accept or reject the request. If the ride is accepted, the driver will reach the rider's location, and both will travel to FCCU. It is a web application with the Front-end based on React JS while Back-end is Django. The database used is SQL Lite, as it is a built-in database for Django. Google Maps API is used to get the location coordinates for travelling, while longitude and Latitude are used to know the exact location of the rider. The application is deployed on Heroku to make it easy for students to use.

Acknowledgement

We truly admire the grace of ALLAH that we were able to complete the project within the given time; it’s all because of his virtual blessing. It would never been possible without participation and contribution of so many people. We sincerely appreciate and acknowledge the part they play in the completion of our project. First of all, we thank ALLAH Almighty who keeps us perseverant on the path of hard work. Secondly, we acknowledge our sense of gratitude to our project supervisor **Sir Akheem and Dr. Aasisa Khanum** with their penetrating guidance in every phase of project. No doubt she proved herself as a committed project supervisor and she is a constant source of inspiration.

Next to her with deep reverence our parents who all always supported us morally and showed their generous love and care though out the entire period. We feel highly oblige to thank all the teacher of computer science who shares their knowledge with us and for their generous attitude. Last but not the least we thank all of our friends and family members for their support.

List of Figures

List of Tables

TABLE OF CONTENTS

Revision History

Name	Date	Reason For Changes	Version

Chapter 1. Introduction

1.1 Introduction

Ridesharing is increasing in developed nations due to everyday commute expenditures increasing daily. Rising costs of fuel have made daily commutes expensive to the masses. Students have a burden of expenses. Cars are one of the primary sources of air pollution; burning gases are deteriorating the atmosphere. Considering the commute costs and the air pollution due to transportation, we have devised an efficient idea to decrease the usage of more cars by creating a carpool app for FCCU Students who would share their ride and the fuel prices among them. It will significantly decrease the fuel expense of students. Over time, the Parking space of FCC is also less compared to the number of students, due to which long lines of cars, wrong parking, and lack of space cause students hurdles and wastage their class time. The Unique point of this app is that the problem here is that no two students go at the same time to university and vice versa. Due to liberal arts education at FCCU, all have different timings. FCCU Car Pooling Service will unite two students through a website who will be going to the same destination. The student willing to share his ride will notify; if a student also wants to join, he can reply and contact to ride.

1.2 Objectives

The aims and objectives of the project would be:

- Reducing overall traffic congestion on the roads
- Reduce peak hour congestion
- Reducing single occupancy car trips by implementing a carpooling system.
- Promoting alternative modes of transport.
- Improve parking in FCC
- Save money by sharing the cost of driving one car.
- Reduce the number of cars on the road.
- Reduce pollution and carbon dioxide emissions.
- Reduces driving-related stress for participants
- Provide social connections for university students

1.3 Problem Statement

There are security risks involved in a third-party commute service; keeping that in mind, a dedicated web app, which is university specific, does the job. However, suppose a ridesharing app made only university-specific, granting access to students based on verifying their student card will maximize security, safety and peace of mind. In that case, it is a win-win situation for both students, and both are saving and getting affordable transportation. Parking in FCCU is a significant problem. It causes blockage inside and outside of campus. Too many students bring their vehicles with FCCU, which requires more space.

Another major thing that this carpool app would help with is dealing with traffic and pollution. Too many vehicles on the road contribute to more traffic and pollution because of the smoke. The carpooling app will help reduce the number of vehicles as many students would come together rather than alone.

1.4 Scope

Web Application is designed considering students' current challenges due to rising fuel and expensive transportation costs. It is a purposefully student-centered web app to solve their problem. My motive is to tackle the challenges students face and provide a peer-to-peer ride-sharing platform to facilitate them in the era of rising fuel and maintenance costs. Air pollution is the topic of the day, and an application that encourages sharing of rides and reduces carbon footprints is highly beneficial. Which also provides opportunities for socializing between students on the go. It would be a student-to-student service and would reduce the need for a third person to provide pick-and-drop, eventually saving costs. The web app is intended to automate and modernize how students travel to university.

Chapter 2. Requirements Analysis

2.1 Literature Review

Ridesharing has snowballed in popularity since the introduction of the first ridesharing app in 2009. Research has found that the main factors driving the growth of the ridesharing industry are convenience, cost-effectiveness, and increased access to transportation for underserved communities.

There are many carpooling applications all over the world. Uber, Careem, and Ola are some of their examples. These applications allow the user to share the commute with different users who are going the same way so that the fare can be shared and everyone reaches the destination on time. These applications provide the solution to high rates of single rides and help adjust with others to be affordable. For about the same price as public transit, passengers save time and enjoy a more comfortable ride, while drivers save some money by sharing the commute cost. This option is better than public transport, like the bus, because it picks you up from your location and you don't have to wait at stations for buses or trains. It allows you to have a good experience at a low cost. The disadvantages of these solutions are that it is for everyone. A person headed to work will share with a student headed to school or university. This can have a time constraint on both people, and they could be late because of the other. When headed to different locations, you will meet different people every day, which could not be a good experience as some might come late, making the user late for their work or university. The individuals who carpool together may finish work at different times and have to reschedule their rides, which could affect the other user. Some users might cancel at the last minute, which would mean the fare is high for the other passengers. These are the issues facing the carpool applications, with users not very fond of using these applications.

To counter these issues, our application should be university-specific. This will help the students head to one place to get together and ride. The ending destination for all the users will be the same. Users who come from the same area and use their cars can use this application to benefit the city's traffic. FCCU has minimal parking space, and many times, there have been blockages because of no parking space, and many students must park their cars outside the university. The carpool will solve this problem, and less parking space will be used if students do not bring their cars and use the carpool. This will also allow the students to interact with their peers and get to know each other. They could decide in university if somebody has a problem going back so that the rest can manage accordingly. This will help in reducing petrol consumption and pollution.

Analysis of similar apps and websites:

UBER/Careem:

They all are pioneers of digitizing and making the commutes accessible and fares reasonable.

The central point in these services, which will be a plus point in our web app, would be that a third-person driver would give the services with his revenue in providing service regularly. If the rider and driver both are students, the need for another person and taking his services will be bypassed, saving the extra costs a student would need to get to a destination.

The car, driver, and service-providing software costs will be eliminated, and the commute will become more reasonable. It does not have peak factor charges which makes the drives expensive.

Airlift, SWYL:

Before the Covid times, these startups provided ride-hailing services using public transport for commutes, making travelling cheap. It was more affordable than going even on a bike.

However, the economic crisis in Pakistan after covid made their business unsustainable, and they closed their operations. So, they are not working anymore, and not all areas were covered, and they would not come individually to a place or residence on the map pinned location.

OLA Cabs:

It is a successful carsharing app and a website used in India, but it is not university specific and caters to a general audience. It is in India and has a web app as well in use.

Carpolyn:

It is a Pakistani App with a database of users, their car registration number, and their transportation route. It does not have a map facility to track or find drivers live. It lacks detailed features like tracking, contacting users, sending or receiving ride requests, etc. It is a static app with little instructiveness.

2.2 User Classes and Characteristics

By the scope of this project, the users will be divided into two categories. One will be the Rider, and one will be the Driver.

DRIVER: This class can create a ride listing, which will be seen by others nearby to request for rideshare, setting up the timetable. Only this class would create a listing and will have the authority to accept or decline the requests of rideshare.

RIDER: This class will be able to request a ride from the list of all nearby drivers (Driver Class). This class can chat and negotiate, Set-up a point of pick-up on the maps.

2.3 Design and Implementation Constraints

As our web app is built using ReactJS and Django, there were certain limitations that arose from using these modern technologies. One of the constraints we encountered is the inability to provide users with live routing and directions within the webapp itself. This limitation is primarily due to the restrictions imposed by the Google Maps API, which offers comprehensive services but at a cost, primarily targeting corporate and enterprise customers. To overcome this constraint, we have implemented a workaround by incorporating a functionality that allows users to be redirected to the native Google Maps app on their devices, where they can access and utilize the live directions feature.

It is important to note that another constraint we faced is the usage limitations and costs associated with the Google Maps API. After a certain threshold of usage, the API becomes a paid service, requiring financial resources to sustain its functionality within our web app. We are aware of this constraint and have devised strategies to address it, such as exploring potential monetization options or seeking alternative mapping solutions if necessary.

Despite these constraints, we provide the best user experience possible within the limitations of the technologies and resources available to us.

2.4 Assumptions and Dependencies

Assumptions:

1. User knows basic working of browser and website.
2. Users would have Google Maps downloaded and have cellular or internet connection to connect to server and client.
3. User is a student of FCCU, and if he has valid driving license if he is signed up as Driver.

Dependencies:

We are relying on Google Maps API, Django as a backend with the SQL Lite as database for storage.

Google is an API provider for maps, for the working of the project, it would be dependent on their allocation of resources.

Some of assumptions are:

1. Django with SQL lite would be enough for the working as a backend.
2. Google Maps API would be free for our limited usage, the threshold to be met. If the threshold of routes, services are more utilized we may integrate google ads for revenue of operational charges.

2.5 Functional Requirements

Note:

There are two modes for using this website, with their own functionalities completely different, so to show which Use-Case is for which one, we have added the Mode which would be using it in brackets.

2.5.1 Use-Case 1: Registration as RIDER or Driver

<table>
<tr><td colspan="2"><b>Identifier</b></td><td colspan="2">UC-1 Registration</td></tr>
<tr><td colspan="2"><b>Purpose</b></td><td colspan="2">Allows user to register as Driver or Rider</td></tr>
<tr><td colspan="2"><b>Priority</b></td><td colspan="2">High</td></tr>
<tr><td colspan="2"><b>Pre-conditions</b></td><td colspan="2">Sign up Button Clicked on Web Page</td></tr>
<tr><td colspan="2"><b>Post-conditions</b></td><td colspan="2">Relevant and Accurate Details Entered</td></tr>
<tr><td colspan="4" align="center"><b>Typical Course of Action</b></td></tr>
<tr><td><b>S#</b></td><td colspan="2"><b>Actor Action</b></td><td><b>System Response</b></td></tr>
<tr><td><b>1</b></td><td colspan="2">The user selects the “signup” option</td><td>Displays Sign up form on screen</td></tr>
<tr><td><b>2</b></td><td colspan="2">User opts as a Rider or Passenger</td><td>System shows different forms if it is Rider or Driver</td></tr>
<tr><td><b>3</b></td><td colspan="2">User enters their details and information</td><td>System verifies the details</td></tr>
<tr><td><b>4</b></td><td colspan="2">User submits the details</td><td>System checks the details and checks if details are different to existing record. Then, saves data and registers the user</td></tr>
<tr><td colspan="4" align="center"><b>Alternate Course of Action</b></td></tr>
<tr><td><b>S#</b></td><td colspan="2"><b>Actor Action</b></td><td><b>System Response</b></td></tr>
<tr><td><b>1</b></td><td colspan="2">The user’s account has already been created</td><td>System displays that id they are trying to sign up with is already in use</td></tr>
</table>

Table 1: UC-1

]

2.5.2 Use-Case 2: Login

Identifier	Login	
Purpose	Allows user to Login	
Priority	High	
Pre-conditions	Login option selected after launching webpage	
Post-conditions	Relevant details entered in Login option	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	The user selects the Login option.	System verifies their choice and displays the form on screen.
2	User enters his id and password	Verifies the details entered
3	Login clicked	User is logged into his account after verification of details
Alternate Course of Action		
S#	**Actor Action**	**System Response**
1	User forgot his password and clicks that option	The system will provide them with an option to reset the password

Table 2: UC-2

2.5.3 Use-Case 3: Search Rides (Rider)

Identifier	UC-3 Show Rides	
Purpose	Passenger Student can check available rides for upcoming days	
Priority	High	
Pre-conditions	User has already logged in the system as a Rider	
Post-conditions	All available rides shown	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	The user clicks on Show Rides	System shows all available rides
2	User clicks Check Schedule	More detailed page opens with schedule and address, vehicle info etc
3		
Alternate Course of Action		
S#	**Actor Action**	**System Response**
1	No rides are shown	System generates a message that no rides are available.

Table 3: UC-3

2.5.4 Use-Case 4: Send Request (Rider)

<table>
<tr><td><b>Identifier</b></td><td colspan="2">UC-4 Send Request</td></tr>
<tr><td><b>Purpose</b></td><td colspan="2">Passenger Student can send request to driver with selecting their preferred timings</td></tr>
<tr><td><b>Priority</b></td><td colspan="2">High</td></tr>
<tr><td><b>Pre-conditions</b></td><td colspan="2">User has already clicked Check Schedule</td></tr>
<tr><td><b>Post-conditions</b></td><td colspan="2">Request sent successfully</td></tr>
<tr><td colspan="3"><b>Typical Course of Action</b></td></tr>
<tr><td><b>S#</b></td><td><b>Actor Action</b></td><td><b>System Response</b></td></tr>
<tr><td><b>1</b></td><td>The user clicks on desired day, time and Return trip or not</td><td>System books desired timings and schedule for rider</td></tr>
<tr><td><b>2</b></td><td>User clicks checkmark of Return</td><td>The systems marks for complete Round Trip and not only one sided.</td></tr>
<tr><td><b>3</b></td><td></td><td></td></tr>
<tr><td colspan="3"><b>Alternate Course of Action</b></td></tr>
<tr><td><b>S#</b></td><td><b>Actor Action</b></td><td><b>System Response</b></td></tr>
<tr><td><b>1</b></td><td>No time slots are available</td><td>System generates response no slots available</td></tr>
</table>

Table 4: UC-4

2.5.5 Use-Case 5: Check Requested Rides (Rider)

<table>
<tr><td><b>Identifier</b></td><td colspan="2">UC-5 Requested Rides</td></tr>
<tr><td><b>Purpose</b></td><td colspan="2">User can check the status of his rides</td></tr>
<tr><td><b>Priority</b></td><td colspan="2">Medium</td></tr>
<tr><td><b>Pre-conditions</b></td><td colspan="2">User is logged into the system.
User has sent at least one ride request</td></tr>
<tr><td><b>Post-conditions</b></td><td colspan="2">All information with Status displayed</td></tr>
<tr><td colspan="3"><b>Typical Course of Action</b></td></tr>
<tr><td><b>S#</b></td><td><b>Actor Action</b></td><td><b>System Response</b></td></tr>
<tr><td><b>1</b></td><td>User clicks on Requested Rides</td><td>System shows all Rides with their Status, Pending or Rejected or Accepted</td></tr>
<tr><td colspan="3"><b>Alternate Course of Action</b></td></tr>
<tr><td><b>S#</b></td><td><b>Actor Action</b></td><td><b>System Response</b></td></tr>
<tr><td><b>1</b></td><td>User has not requested any ride previously</td><td>System generates a message that no rides history record is available</td></tr>
</table>

Table 5: UC-5

2.5.6 Use-Case 6: Upload Schedule (Driver)

<table>
<tr><td><b>Identifier</b></td><td colspan="2">UC-6 Upload Carpool Ride Schedule</td></tr>
<tr><td><b>Purpose</b></td><td colspan="2">Driver User Uploads ride schedule weekly</td></tr>
<tr><td><b>Priority</b></td><td colspan="2">High</td></tr>
<tr><td><b>Pre-conditions</b></td><td colspan="2">Driver has clicked Schedule from navbar</td></tr>
<tr><td><b>Post-conditions</b></td><td colspan="2">Schedule is uploaded and can be seen from Schedules page of from Rider Section in Show Rides</td></tr>
<tr><td colspan="3"><b>Typical Course of Action</b></td></tr>
<tr><td>S#</td><td><b>Actor Action</b></td><td><b>System Response</b></td></tr>
<tr><td><b>1</b></td><td>User clicks on Time/Day in Schedule</td><td>System adds the timings respectedly</td></tr>
<tr><td><b>2</b></td><td>User enters his Arrival and Return for respected Days</td><td>System adds it to their schedule</td></tr>
<tr><td><b>3</b></td><td>User clicks on submit Schedule</td><td>System verifies the information and adds the schedule</td></tr>
<tr><td colspan="3"><b>Alternate Course of Action</b></td></tr>
<tr><td><b>S#</b></td><td><b>Actor Action</b></td><td><b>System Response</b></td></tr>
<tr><td><b>1</b></td><td>User enters invalid timings</td><td>System gives error to enter correct/accurate timings</td></tr>
</table>

Table 6: UC-6

2.5.7 Use-Case 7: Update/Remove Schedule (Driver)

<table>
<tr><td colspan="2"><b>Identifier</b></td><td colspan="2">UC-7 Update/Remove Carpool Ride Schedule</td></tr>
<tr><td colspan="2"><b>Purpose</b></td><td colspan="2">Driver User Updates his Schedule</td></tr>
<tr><td colspan="2"><b>Priority</b></td><td colspan="2">High</td></tr>
<tr><td colspan="2"><b>Pre-conditions</b></td><td colspan="2">Driver user is logged in the system
There is an existing schedule</td></tr>
<tr><td colspan="2"><b>Post-conditions</b></td><td colspan="2">Schedule is uploaded</td></tr>
<tr><th colspan="4">Typical Course of Action</th></tr>
<tr><th>S#</th><th colspan="2">Actor Action</th><th>System Response</th></tr>
<tr><td><b>1</b></td><td colspan="2">User clicks on Update Schedule</td><td>System verifies if there is an existing schedule and then display the same schedule to be updated</td></tr>
<tr><td><b>2</b></td><td colspan="2">User enters his new schedule timings or just removes existing schedule</td><td>System verifies the information and checks timings are correct or removes existing schedule</td></tr>
<tr><td><b>3</b></td><td colspan="2">User clicks on submit Schedule</td><td>System verifies the information and updates the ride details</td></tr>
<tr><th colspan="4">Alternate Course of Action</th></tr>
<tr><th>S#</th><th colspan="2">Actor Action</th><th>System Response</th></tr>
<tr><td><b>1</b></td><td colspan="2">User enters invalid timings</td><td>System gives error to enter correct/accurate timings</td></tr>
<tr><td><b>2</b></td><td colspan="2">User enters same timings as before</td><td>System gives warning message that timings are not updated</td></tr>
</table>

Table 7: UC-7

2.5.8 Use-Case 8: Update/Add Information (Rider/Driver)

<table>
<tr><td colspan="2"><b>Identifier</b></td><td colspan="2">UC-8 Add/Update Information</td></tr>
<tr><td colspan="2"><b>Purpose</b></td><td colspan="2">User adds/updates his information (name, place, phone veh#)</td></tr>
<tr><td colspan="2"><b>Priority</b></td><td colspan="2">Medium</td></tr>
<tr><td colspan="2"><b>Pre-conditions</b></td><td colspan="2">User is logged into the system</td></tr>
<tr><td colspan="2"><b>Post-conditions</b></td><td colspan="2">Information added/removed/updated</td></tr>
<tr><th colspan="4">Typical Course of Action</th></tr>
<tr><th>S#</th><th colspan="2">Actor Action</th><th>System Response</th></tr>
<tr><td><b>1</b></td><td colspan="2">User clicks on update information</td><td>System displays a page with all information of user</td></tr>
<tr><td><b>2</b></td><td colspan="2">User selects specific option they want to change</td><td>System displays a new box for information to be updated</td></tr>
<tr><td><b>3</b></td><td colspan="2">User clicks on update information</td><td>System verifies the information and updates the information</td></tr>
<tr><th colspan="4">Alternate Course of Action</th></tr>
<tr><th>S#</th><th colspan="2">Actor Action</th><th>System Response</th></tr>
<tr><td><b>1</b></td><td colspan="2">User enters invalid information</td><td>System gives error to enter correct information</td></tr>
</table>

Table 8: UC-8

2.5.9 Use-Case 9: Accept/Decline Ride Request (Driver)

<table>
<tr><td><b>Identifier</b></td><td colspan="2">UC-9 Accept/Decline Ride Request</td></tr>
<tr><td><b>Purpose</b></td><td colspan="2">The driver accepts or declines request of rider</td></tr>
<tr><td><b>Priority</b></td><td colspan="2">High</td></tr>
<tr><td><b>Pre-conditions</b></td><td colspan="2">Driver receives Ride Request and is in Show Requests Page</td></tr>
<tr><td><b>Post-conditions</b></td><td colspan="2">The status is updated according to the driver’s response</td></tr>
<tr><th colspan="3">Typical Course of Action</th></tr>
<tr><th>S#</th><th>Actor Action</th><th>System Response</th></tr>
<tr><td><b>1</b></td><td>User clicks on Pending Rides when gets notified</td><td>System displays a page with all pending rides</td></tr>
<tr><td><b>2</b></td><td>User selects accept or decline ride button</td><td>System adds or rejects the ride in status</td></tr>
<tr><th colspan="3">Alternate Course of Action</th></tr>
<tr><th>S#</th><th>Actor Action</th><th>System Response</th></tr>
<tr><td><b>1</b></td><td>User clicks on pending rides</td><td>System displays “no rides in waiting list”</td></tr>
</table>

Table 9: UC-9

2.5.10 Use-Case 10: All Accepted Rides (Driver)

<table>
<tr><td><b>Identifier</b></td><td colspan="2">UC-10 All Accepted Rides</td></tr>
<tr><td><b>Purpose</b></td><td colspan="2">Driver can Start or Cancel the Ride which is already accepted</td></tr>
<tr><td><b>Priority</b></td><td colspan="2">Medium</td></tr>
<tr><td><b>Pre-conditions</b></td><td colspan="2">Driver has accepted a Ride</td></tr>
<tr><td><b>Post-conditions</b></td><td colspan="2">Maps Page is opened with option of Redirection to Riders Origin</td></tr>
<tr><th colspan="3">Typical Course of Action</th></tr>
<tr><th>S#</th><th>Actor Action</th><th>System Response</th></tr>
<tr><td><b>1</b></td><td>Driver clicks on Accepted Rides</td><td>System displays a page with buttons to Start Ride or End Ride</td></tr>
<tr><th colspan="3">Alternate Course of Action</th></tr>
<tr><th>S#</th><th>Actor Action</th><th>System Response</th></tr>
<tr><td><b>2</b></td><td>User clicks on Accepted Rides</td><td>Driver has not accepted any ride previously, so the area is blank</td></tr>
</table>

Table 20: UC-10

2.5.11 Use-Case 11: Start Ride (Driver)

<table>
<tr><td><b>Identifier</b></td><td colspan="2">UC-11 Start Ride</td></tr>
<tr><td><b>Purpose</b></td><td colspan="2">Driver can Start or Cancel the Ride which is already accepted</td></tr>
<tr><td><b>Priority</b></td><td colspan="2">Medium</td></tr>
<tr><td><b>Pre-conditions</b></td><td colspan="2">Driver has accepted a Ride</td></tr>
<tr><td><b>Post-conditions</b></td><td colspan="2">Maps Page is opened with option of Redirection to Riders Origin</td></tr>
<tr><td colspan="3"><b>Typical Course of Action</b></td></tr>
<tr><td><b>S#</b></td><td><b>Actor Action</b></td><td><b>System Response</b></td></tr>
<tr><td><b>1</b></td><td>Driver clicks on Start Ride</td><td>System opens a Google Maps, with origin of Rider</td></tr>
<tr><td colspan="3"><b>Alternate Course of Action</b></td></tr>
<tr><td><b>S#</b></td><td><b>Actor Action</b></td><td><b>System Response</b></td></tr>
<tr><td><b>2</b></td><td>User clicks on Cancel Ride</td><td>The Ride is cancelled, and no maps page will open</td></tr>
</table>

Table 31: UC-11

2.6 Use Case Diagram

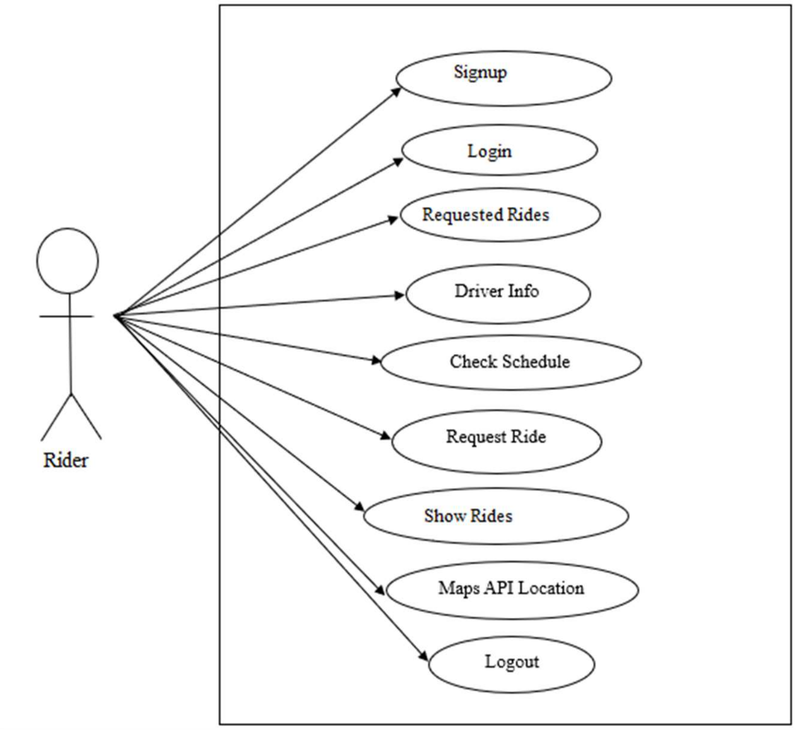


Figure 1: Use Case Diagram 1

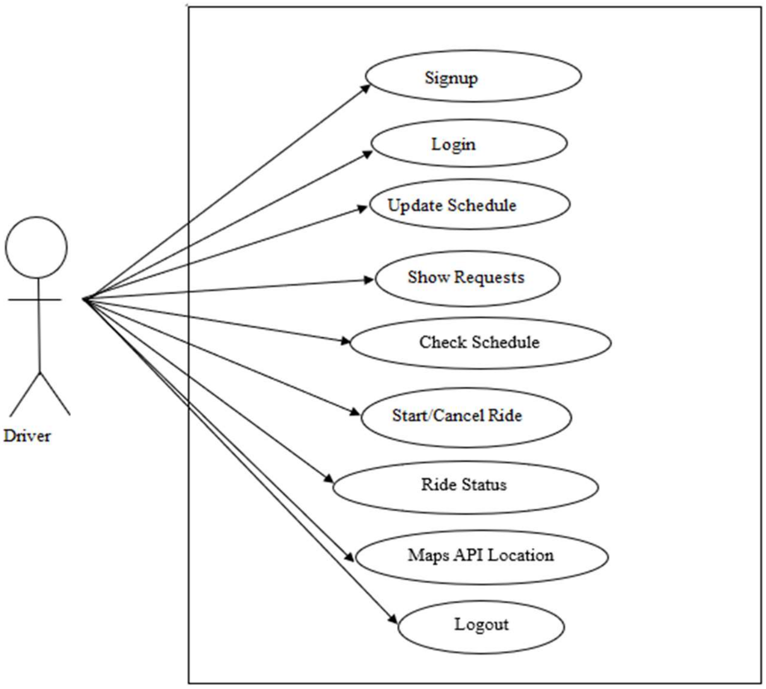


Figure 2: Use Case Diagram 2

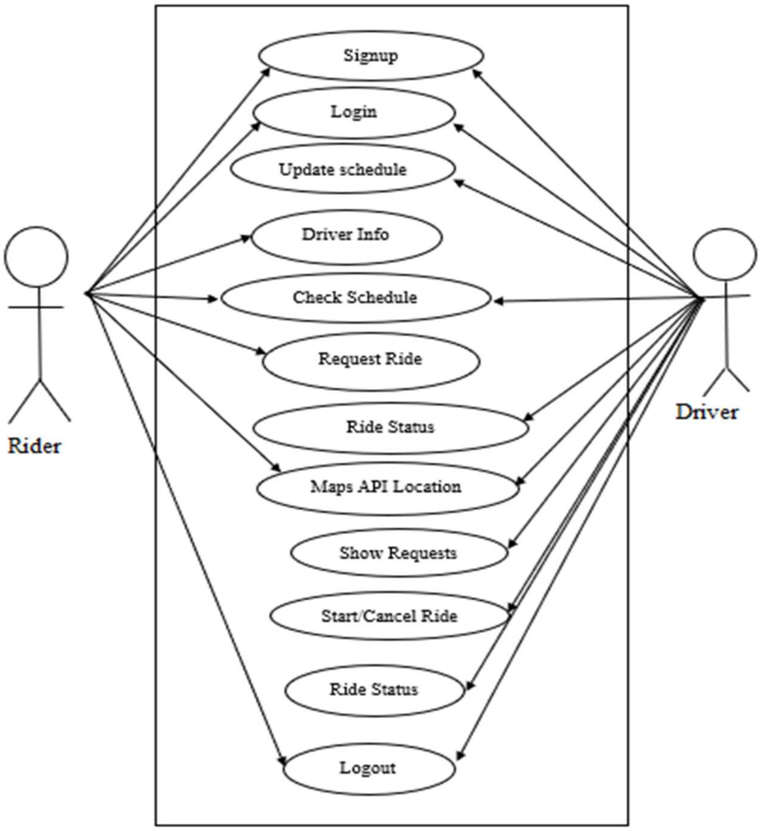


Figure 3: Use Case Diagram 3

2.7 Nonfunctional Requirements

2.7.1 Performance Requirements

The application shall have specifications other than technical requirements. There are performance, protection, safety and efficiency criteria. Complying with the non-functional specifications is necessary to get the project up to the mark. These requirements are to be fulfilled so that the project is well explained and the design feature is suitable according to it. This will help the project elevate and be more descriptive.

2.7.2 Safety Requirements

The application shall have the privacy of the data and anonymity of the driver using it. It is, therefore, not vulnerable to any leakage of information. It allows the user to have personal information. The application will also keep the user data private, and only authorized people can see it saving the integrity of the application. The information will be kept private so that the leakage of it is prevented, as the information is essential.

2.7.3 Security Requirements

The project does not leak any information about the driver or the user using it. It does not store any unnecessary information that is not needed for the use of the application. High-level security methods are used to protect the data being provided by the driver and the user.

2.7.4 Additional Software Quality Attributes

2.7.4.1 Maintainability:

This system can be maintained under low maintenance circumstances. Not much haste is needed to maintain its

2.7.4.2 Portability:

This system does not require major portability as this will be accessed through webpages on any browser.

2.7.4.3 Reusability:

The system can be configured to store data from the user, admin, and the driver so that they can login later for better user experience. It will allow them to stay connected with the app and share experience of using it.

2.8 Other Requirements

2.8.1 Licensing Requirements

- Not Applicable

2.8.2 Legal, Copyright, and Other Notices

- We would be buying our domain and web hosting.
- Our copyright and trademark with logo will be displayed on our webpage.

Chapter 3. System Design

3.1 Application and Data Architecture

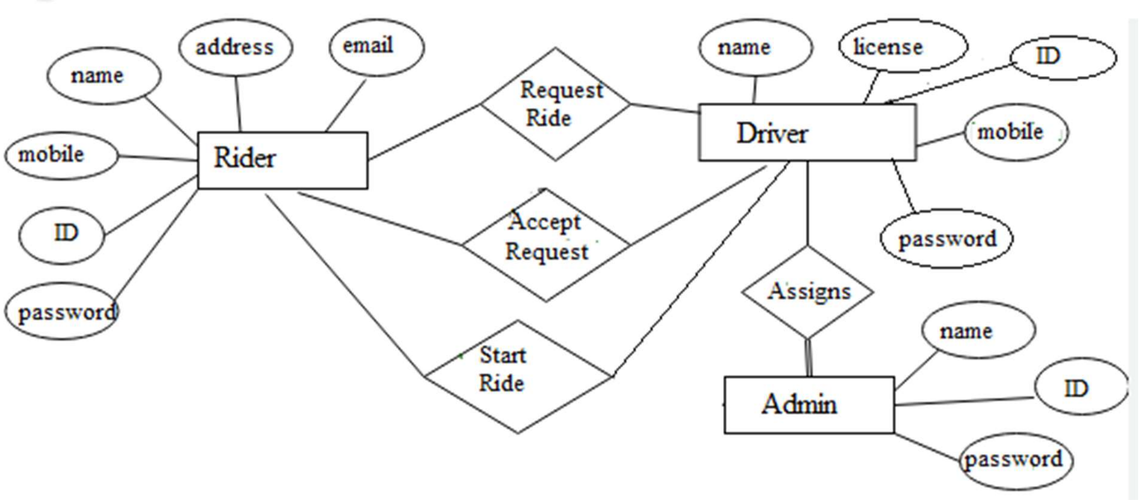
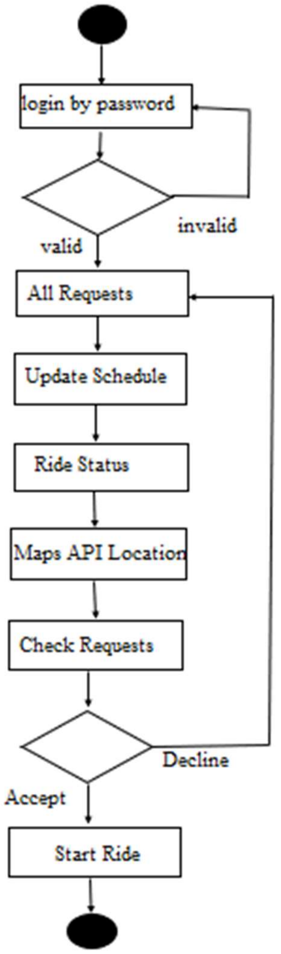


Figure 4: ER Diagram

Figure 1, shown above, shows the ER diagram of our system. This diagram shows how each use case of our system is linked to the user. Let us look at one of these use cases. For example, there will be only one server, but the users viewing it could be many, and they would have access to all the public information of the server, such as the front end and the ability to interact with it.

Initially, the user is made to fill in all mandatory fields in the registration form. Once the user clicks submit, the username is verified. If the username is already present, the user is again taken back to change the username. If the username is absent, it checks for the password and remaining mandatory fields. If any mandatory fields are left empty or filled incorrectly, then the user is informed to enter the correct values. Once all these verifications are succeeded, then the registration is done. User Login to the system by entering username & password, then submit it, verify it & then grant access.

Figure 2 below shows the Activity diagram of our project. The image showcases how our system operates, what is the starting point and what is the endpoint. Activity diagrams graphically represent workflows. They can be used to describe business workflow or the operational workflow of any component in a system. Sometimes activity diagrams are used as an alternative to State machine diagrams. Check out this wiki article to learn about symbols and the usage of activity diagrams.



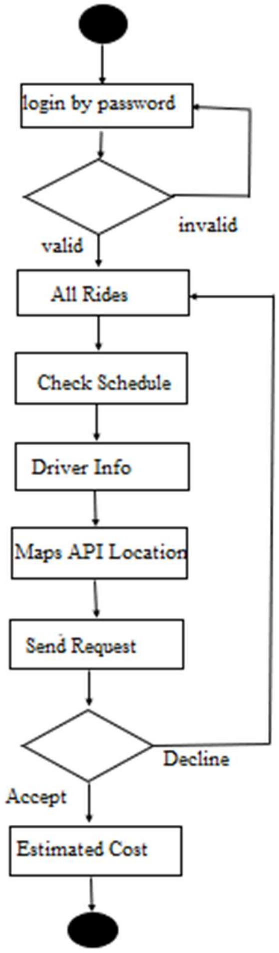


Figure 5: Driver Activity Diagram ‘ Figure 6: Rider Activity Diagram

3.2 Component Interactions and Collaborations

The figure 3,4,5 shown below is that of the DFD or Data Flow Diagram of our project. What this figure shows is that the user will select the data from the server, then the server transfers that data to the database which in turn transfers it to the algorithm for it to process. The algorithm then sends the result of that data back to the database which in turn using the server is displayed on the front end of the website.

DFD Level 0

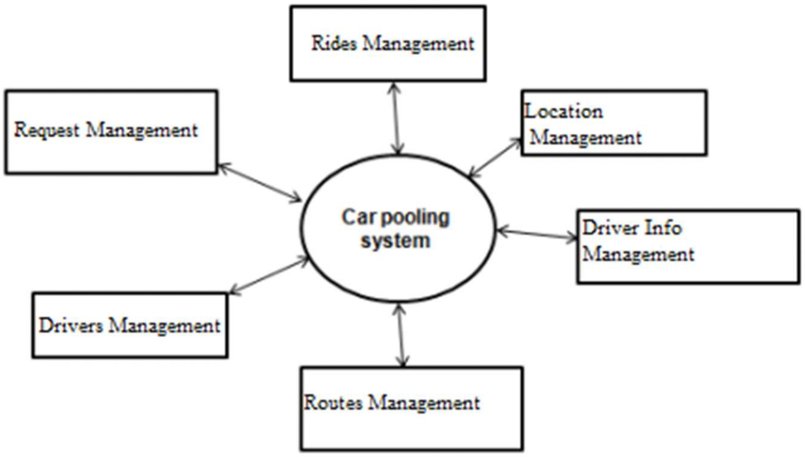


Figure 7: DFD Level 0

DFD Level 1

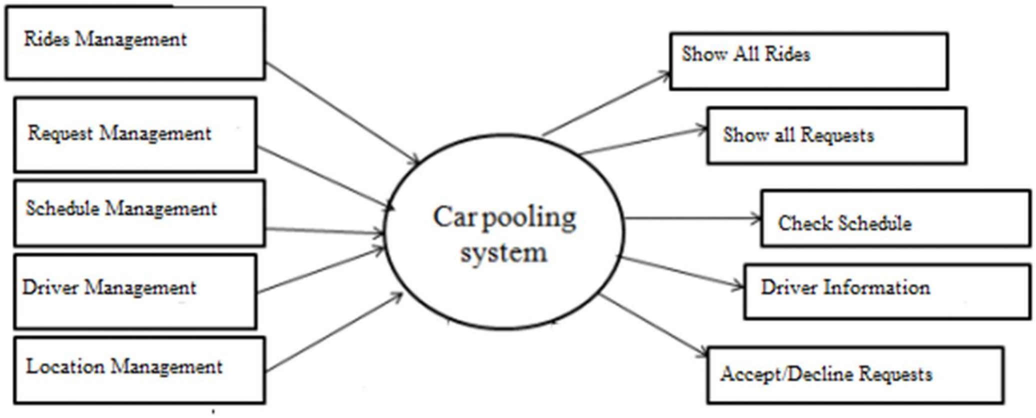


Figure 8: DFD Level 1

DFD Level 2

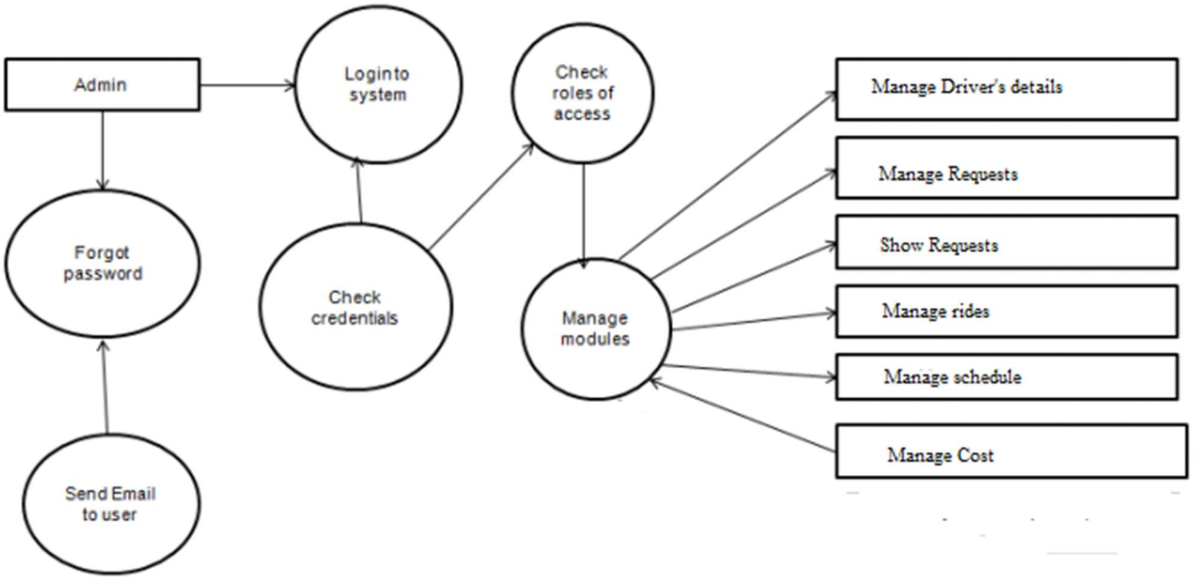


Figure 9: DFD Level 2

The figure 10 shown below is that of the Sequence Diagram of our project.

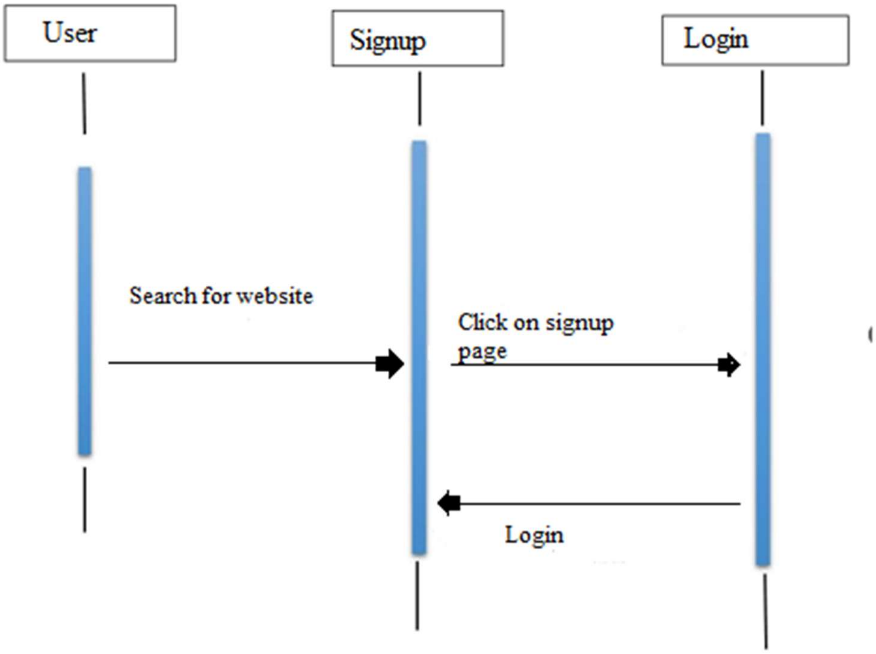


Figure 10: User Sequence Diagram

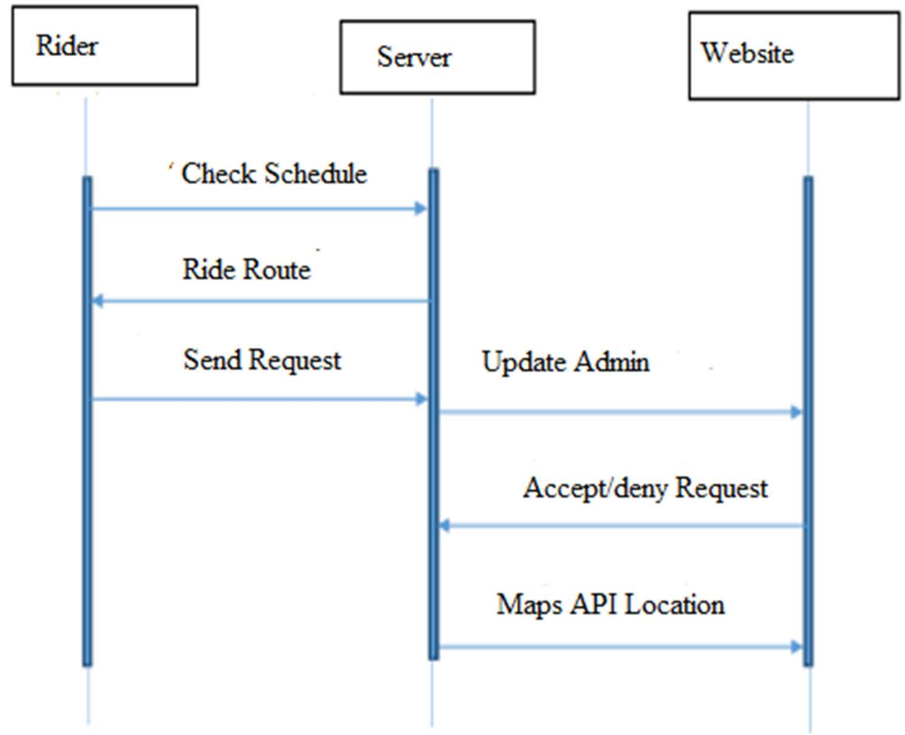


Figure 11: Rider Sequence Diagram

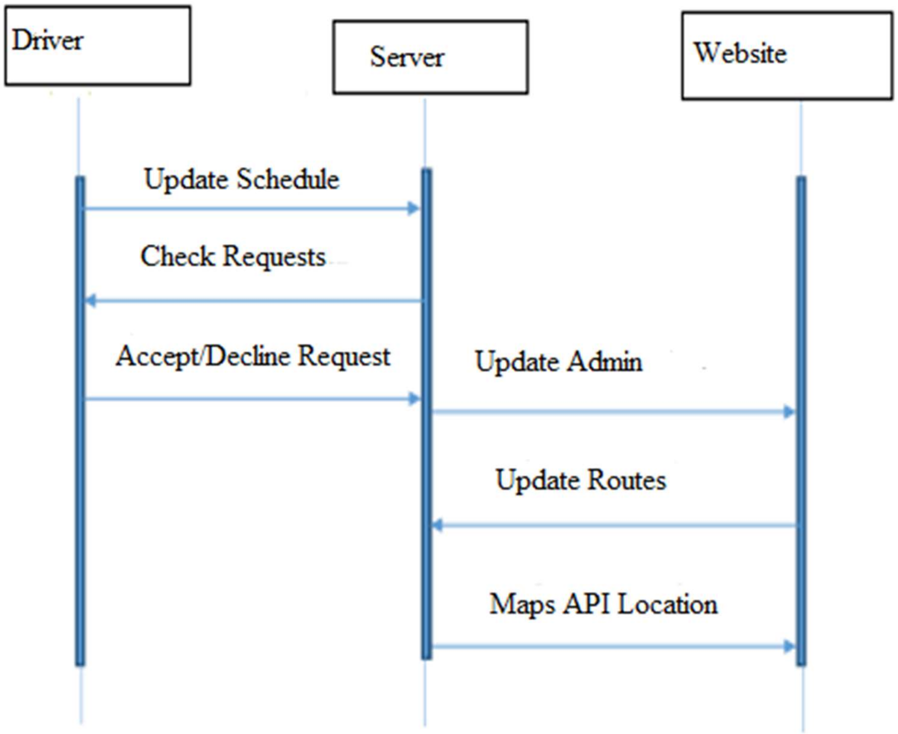


Figure 12: Driver Sequence Diagram

3.3 System Architecture

In this web application, after opening the website, on homepage, user will have following options:

FOR RIDER:

- Home
- NavBar
- Sign-Up
- Log-Out
- Contact Us
- About Us

Home

Home has all the paths for navigation, for drivers as well as riders. It will be the main informational links.

Signup

Rider and driver both can sign up on this website by providing their basic information like name, address, contact info etc. For the driver the signup will have some different requirements like car name, vehicle number, license number. The driver will just provide name, email and would add their origin location.

Login

After signup, the rider and driver can easily be login by username and password. The login will come after signup.

Nav Bar

Keep login to the website user(rider/driver) has access to the all features of this website like information about rides, rides schedule, ride status, date and time and also cost per hour. After getting information rider can send request or his or her suitable ride and driver can accept or decline request as per his ease. If driver accept request, then he will start ride and maps API location shows route of required location.

Contact Us

If the user wishes to convey a message or feedback on our website, they can go to our Contact Us page and send us the queries they have so that we can response back to their queries ASAP.

About Us

The About Us page of our website is an essential source of information for all who want to know more about our carpool service.

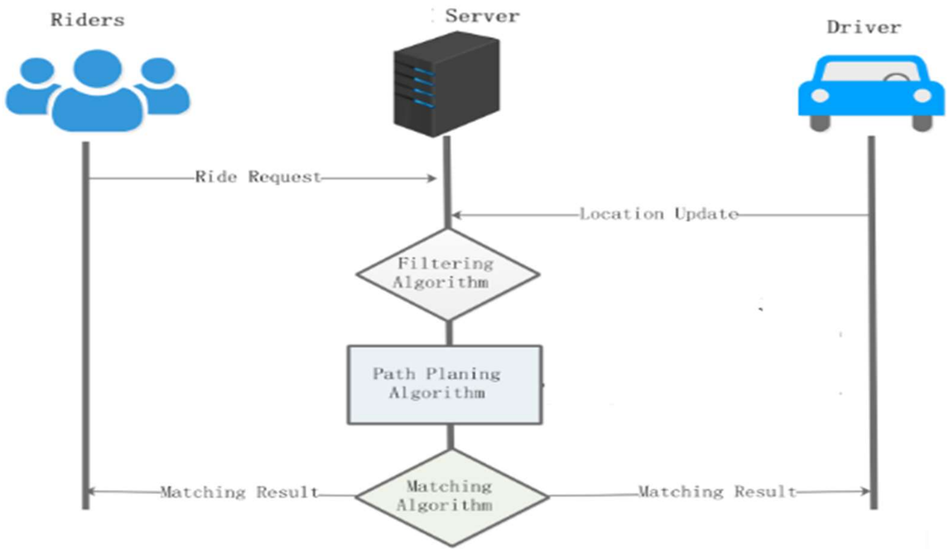


Figure 13: Architecture Diagram

3.4 Architecture Evaluation

Server-Side Technologies:

HTML: The Hypertext Markup Language or HTML is the standard markup language for documents designed to be displayed in a web browser.

CSS: Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML

JavaScript: is a dynamic programming language that is mainly used for client-side scripts and web development. Lately, programmers have been able to implement standalone application servers using this technology.

ReactJS: JavaScript library for UI

Django: Django is a free and open-source, Python-based web framework that follows the model–template–views architectural pattern. It can be used instead of JavaScript.

SQLite: It is a lightweight database which comes already with the Django Setup, and works efficiently.

Google Maps: is mapping application developed by Google for map and directions guidance. It will be integrated in the application to provide possible routes during carpool. It will help in the API process.

Pros:

- SQLite comes already with Django setup, which makes it efficient and lightweight as well as faster response
- ReactJs is a library of JavaScript, which only refreshed according to DOM Manipulation.

Cons:

- Pakistan doesn't has any satellite for maps so we used Google's API.

3.5 Component-External Entities Interface

There is usage of Google Maps API, so we used the service of Google to use it, and used JSON and Axios API for communication. We used bootstrap as well for responsive components.

3.6 Screenshots/Prototype

3.6.1 Workflow

User Registration:

If the user wishes to convey a message or feedback on our website, they can go to our Contact Us page and send us the queries they have so that we can response back to their queries ASAP.

User Registration:

Riders and drivers need to register on the website by providing their basic details such as name, contact information, and email address, location, vehicle information.

Each user should create a unique username and password to securely access their accounts.

User Login:

Users will log in using their registered username and password.

The website should authenticate the login credentials and grant access to the respective user's dashboard.

Rider Requests Ride:

After logging in, riders can navigate to the ride request section.

They can enter their pick-up location, destination, and any additional details or preferences.

Riders will submit their ride request, which will be visible to available drivers.

Driver Accepts Ride Request:

Drivers who are logged in can access the ride request section.

They can view the available ride requests with details of pick-up and drop-off locations.

Drivers have the option to accept or decline a ride request based on their availability and preferences.

When a driver accepts a ride request, the system will notify the rider and provide them with the driver's details.

Driver Adds Ride Schedule:

Drivers can access their schedule management section in their dashboard.

They can add their availability and preferred time slots for offering rides.

Drivers should provide details such as start and end times, days of the week, and any other relevant information.

Rider Accepts Scheduled Ride:

Riders can view the driver's available ride schedules in the scheduling section of their dashboard.

They can select a suitable schedule offered by a driver.

The rider can accept the scheduled ride, and the system will confirm the booking with both the rider and driver.

Log Out:

Users can log out of their accounts to ensure the security of their personal information.

3.6.2 Screens

The display of our web interface is shown below in the following images:

The figure 11 shows the home page of our website, here the user can get an interface for the website

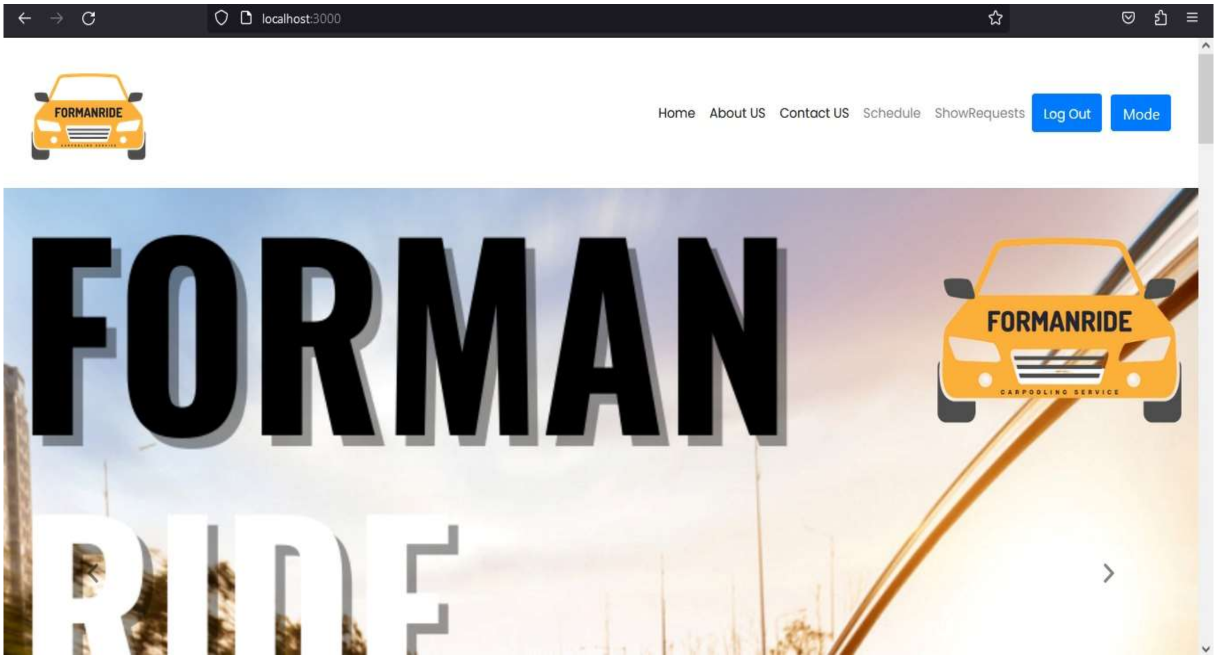
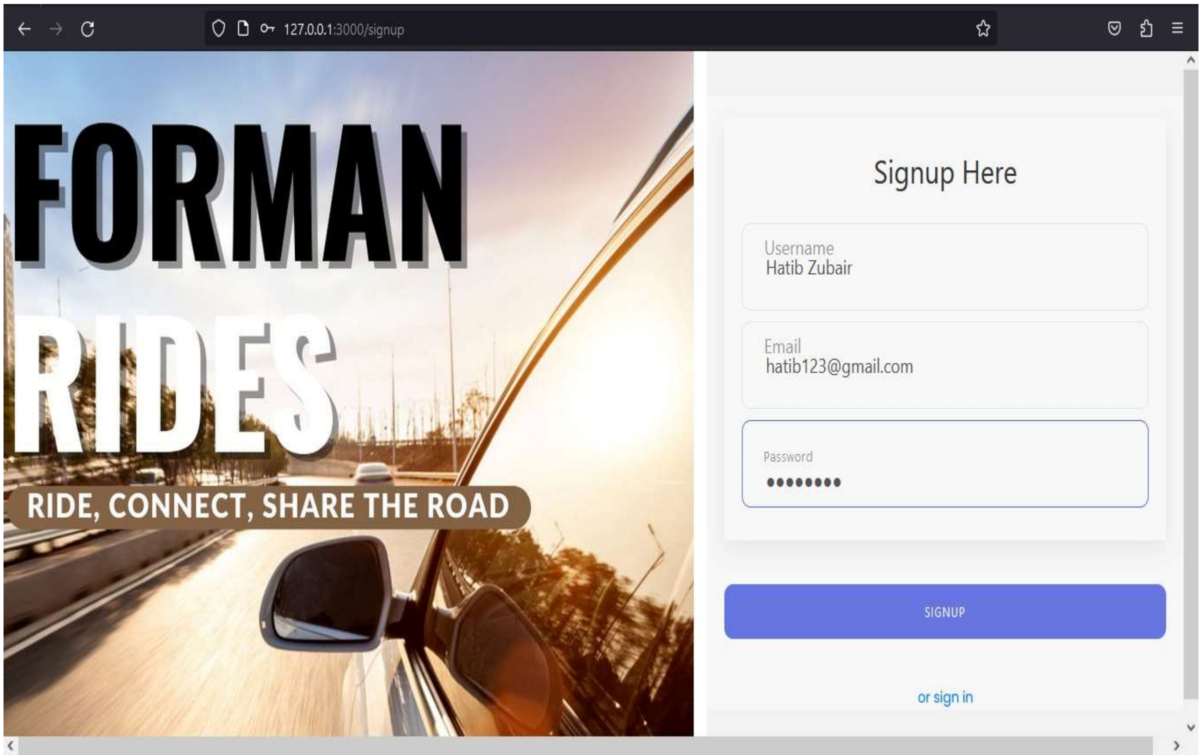


Figure 14: Home Page

The figure 14 shows the signup page of our website, here the user can sign up to their account by creating their username and password.



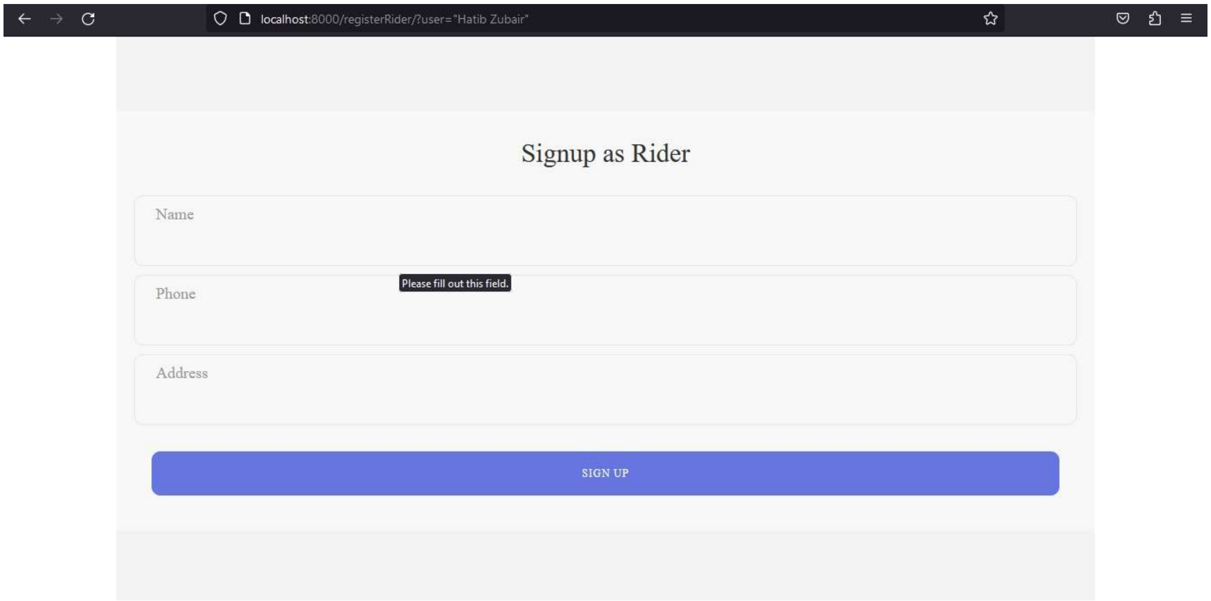
Username: Hatib Zubair

Email: hatib123@gmail.com

Password: ••••••••

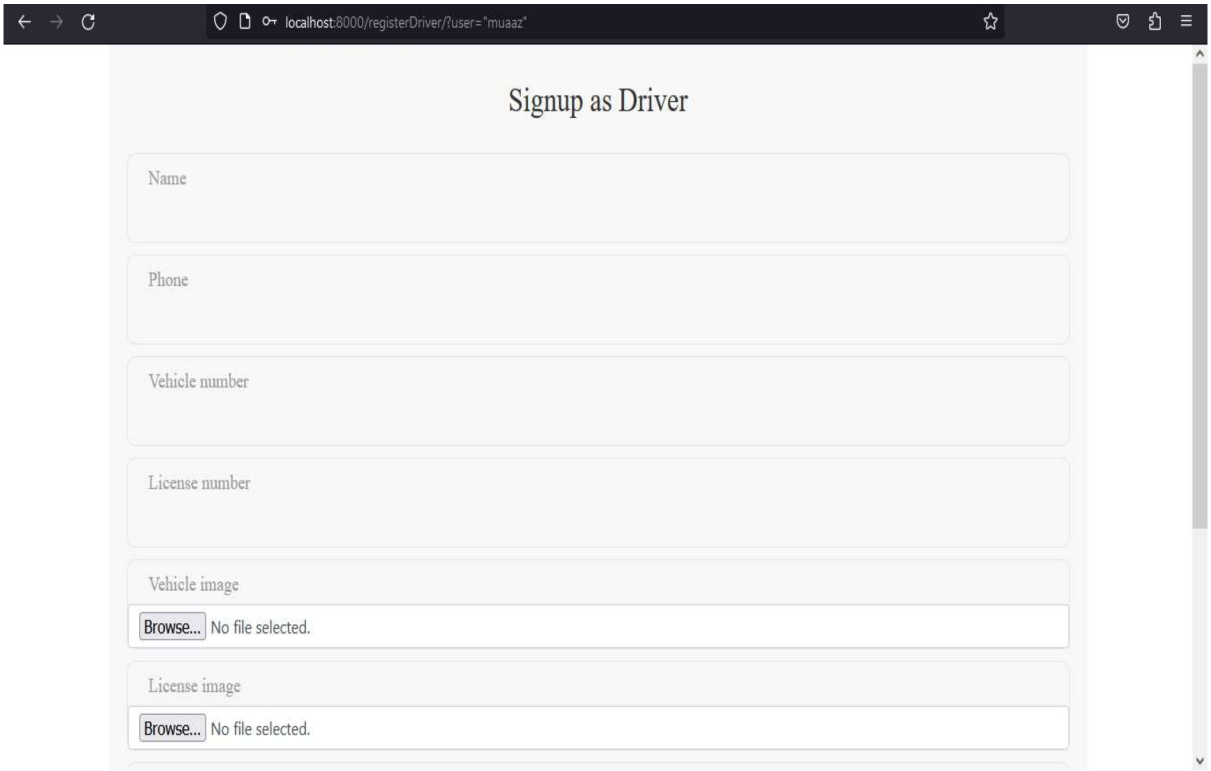
Figure 15: Signup Page

Users have to options here they can sign up as Rider as well as Driver



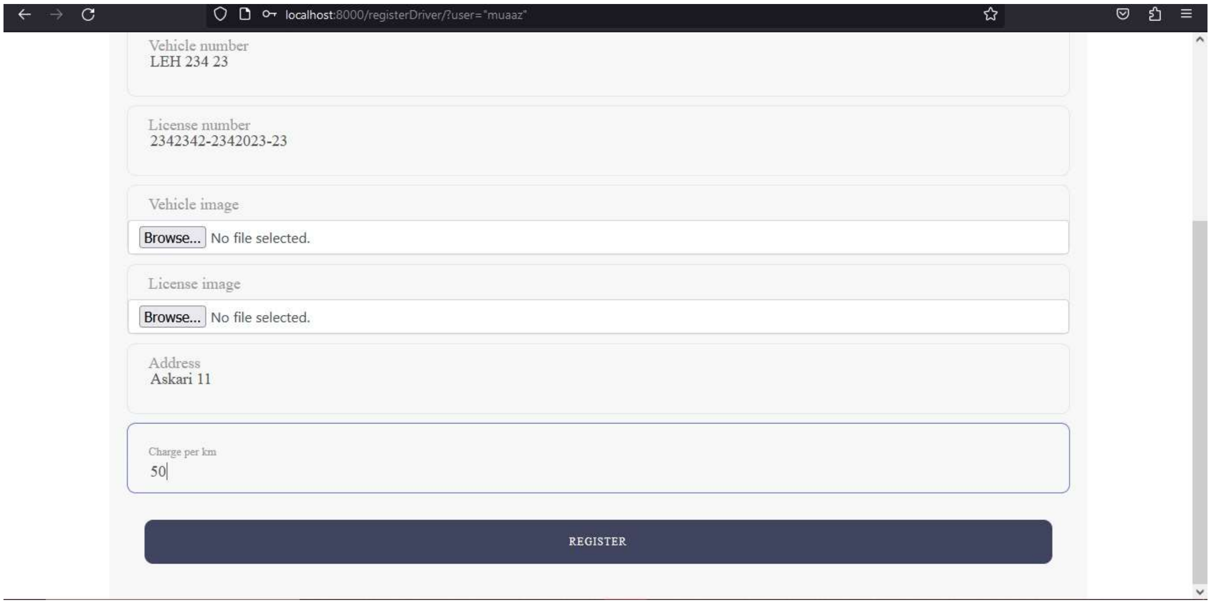
localhost:8000/registerRider/?user="Hatib Zubair"
Signup as Rider
Name
Phone
Please fill out this field.
Address
SIGN UP

Figure 16: Signup Page (Rider)



localhost:8000/registerDriver/?user="muaaz"
Signup as Driver
Name
Phone
Vehicle number
License number
Vehicle image
Browse... No file selected.
License image
Browse... No file selected.

Figure 17: Signup Page (Driver



Vehicle number
LEH 234 23

License number
2342342-2342023-23

Vehicle image
Browse... No file selected.

License image
Browse... No file selected.

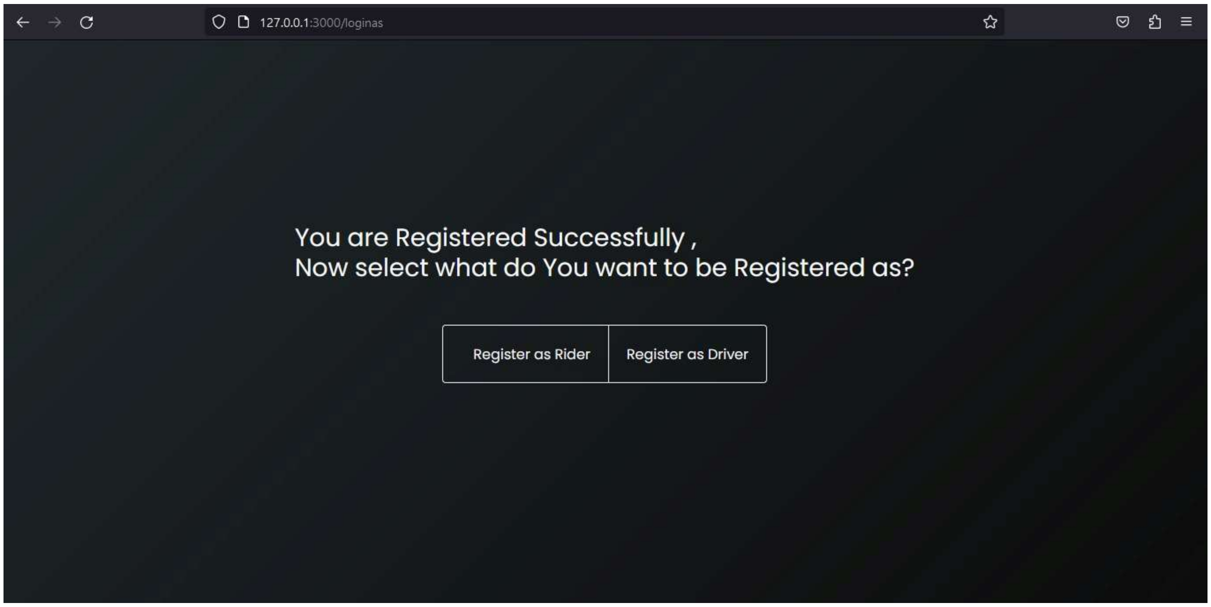
Address
Askari 11

Charge per km
50

Figure 18: Signup Page

If user registered successfully this page will appear

Figure 19: Registered



You are Registered Successfully ,
Now select what do You want to be Registered as?

The figure 17 shows the login page of our website, here the user can sign-in to their account with their username and password.

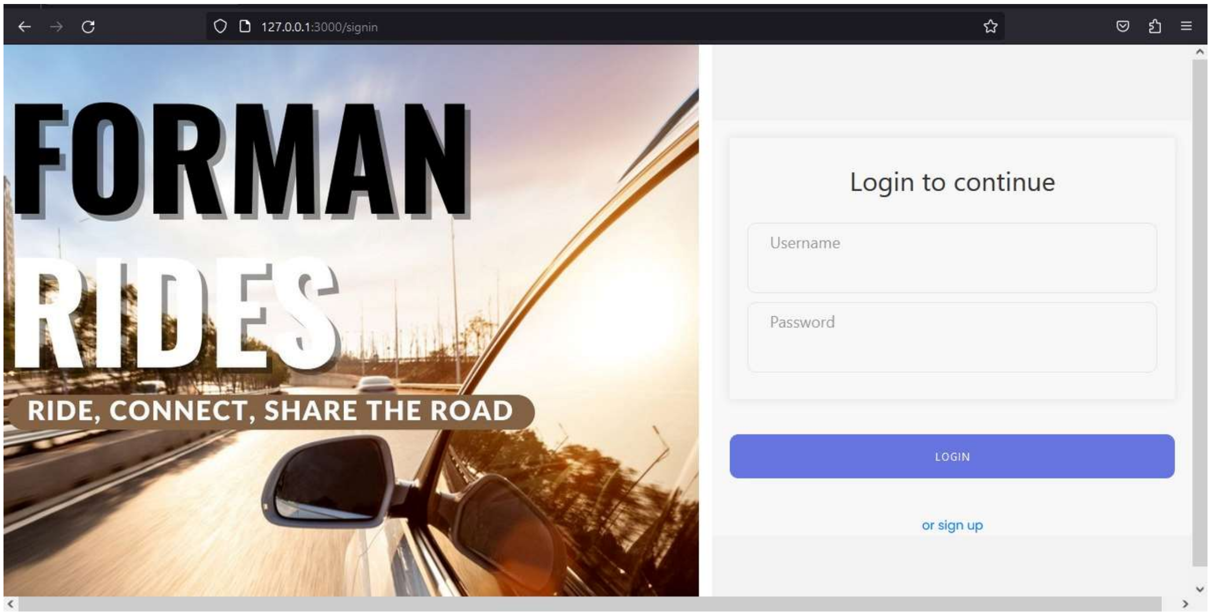
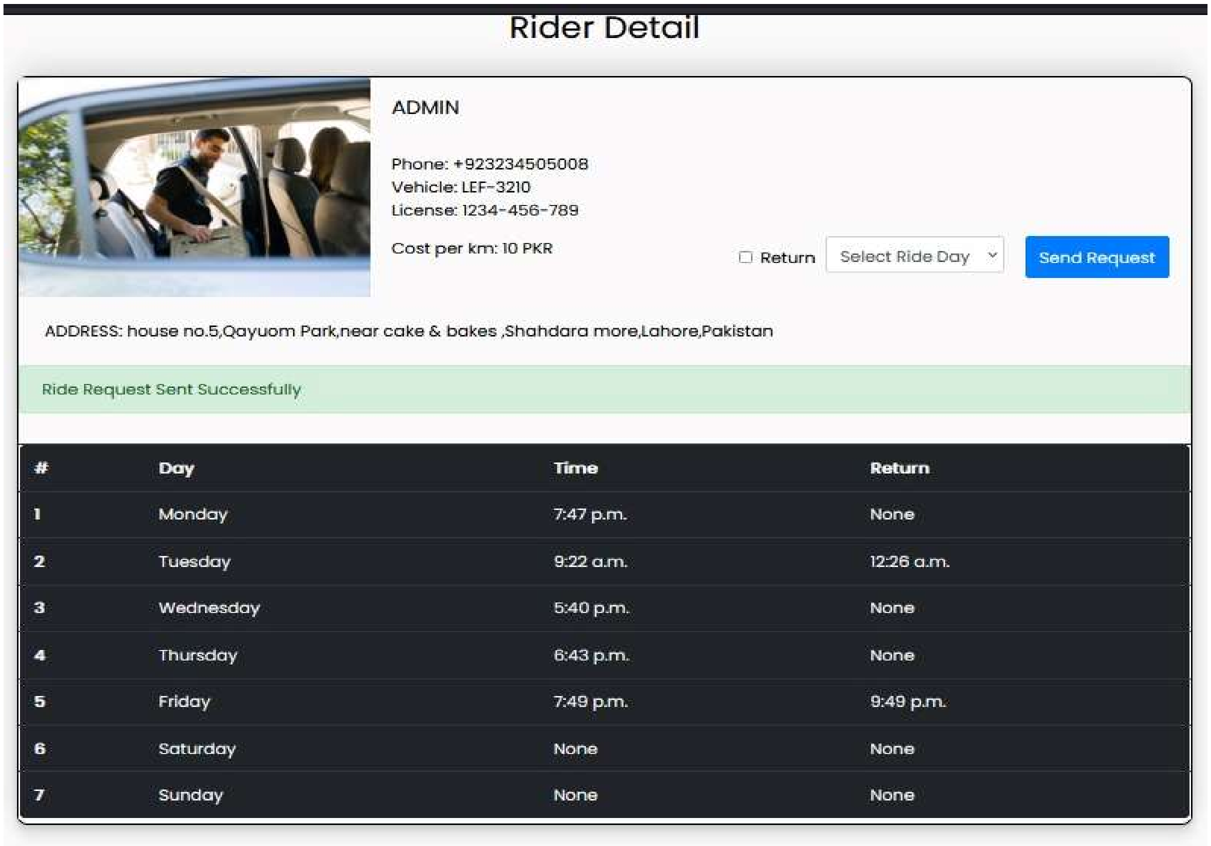


Figure 20: Login Page

Figure 20 shows rider details



Rider Detail

ADMIN

Phone: +923234505008
Vehicle: LEF-3210
License: 1234-456-789

Cost per km: 10 PKR

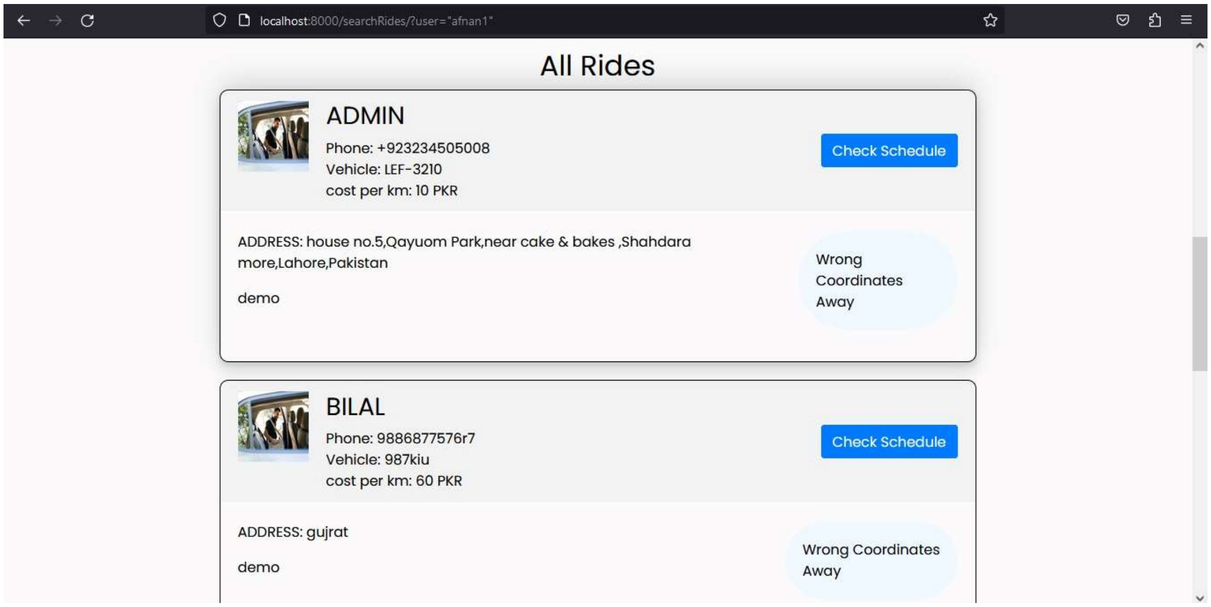
ADDRESS: house no.5,Qayuom Park,near cake & bakes ,Shahdara more,Lahore,Pakistan

Ride Request Sent Successfully

#	Day	Time	Return
1	Monday	7:47 p.m.	None
2	Tuesday	9:22 a.m.	12:26 a.m.
3	Wednesday	5:40 p.m.	None
4	Thursday	6:43 p.m.	None
5	Friday	7:49 p.m.	9:49 p.m.
6	Saturday	None	None
7	Sunday	None	None

Figure 21: Rider Details

Figure 21 illustrate all available rides



All Rides

ADMIN

Phone: +923234505008
Vehicle: LEF-3210
cost per km: 10 PKR

ADDRESS: house no.5,Qayuom Park,near cake & bakes ,Shahdara more,Lahore,Pakistan

demo

Wrong Coordinates Away

BILAL

Phone: 9886877576r7
Vehicle: 987kiu
cost per km: 60 PKR

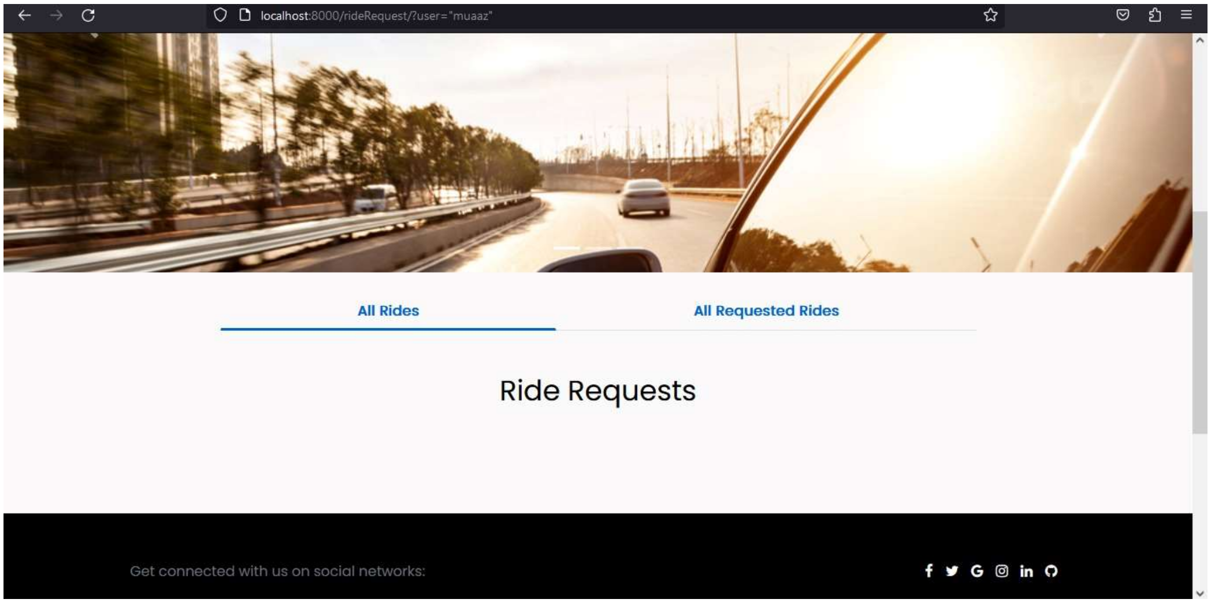
ADDRESS: gujrat

demo

Wrong Coordinates Away

Figure 22: All Rides

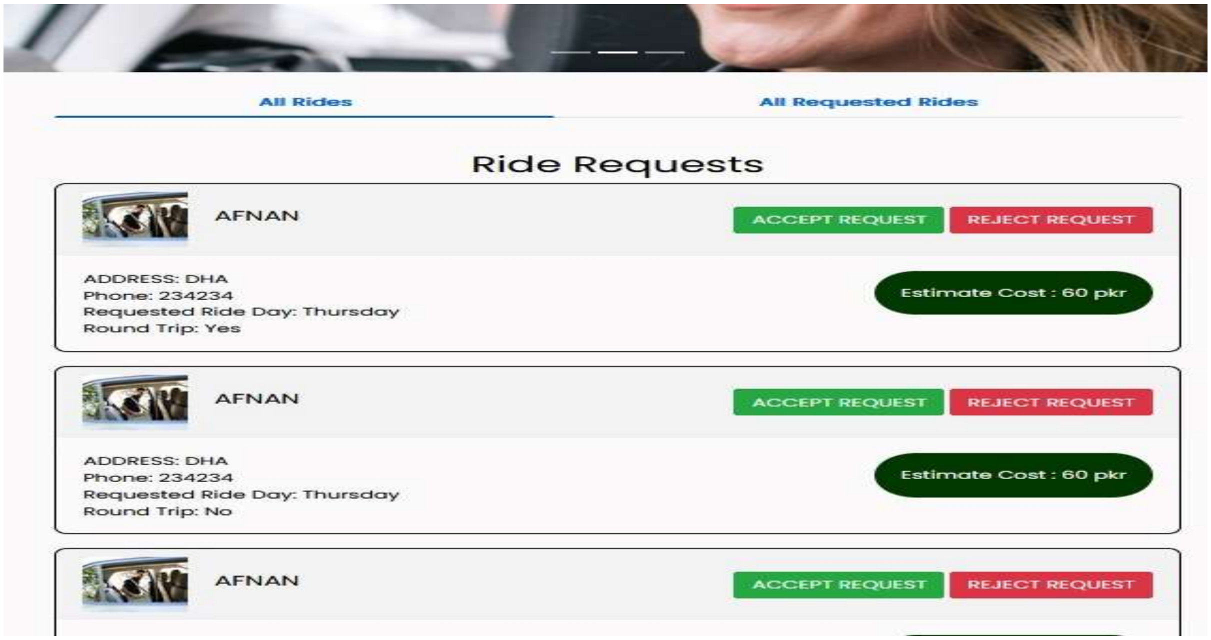
Figure 22 and 23 shows all ride requests that a driver received



Ride Requests

Get connected with us on social networks:

Figure 23: Ride Requests



Ride Requests

AFNAN

ADDRESS: DHA
Phone: 234234
Requested Ride Day: Thursday
Round Trip: Yes

Estimate Cost : 60 pkr

AFNAN

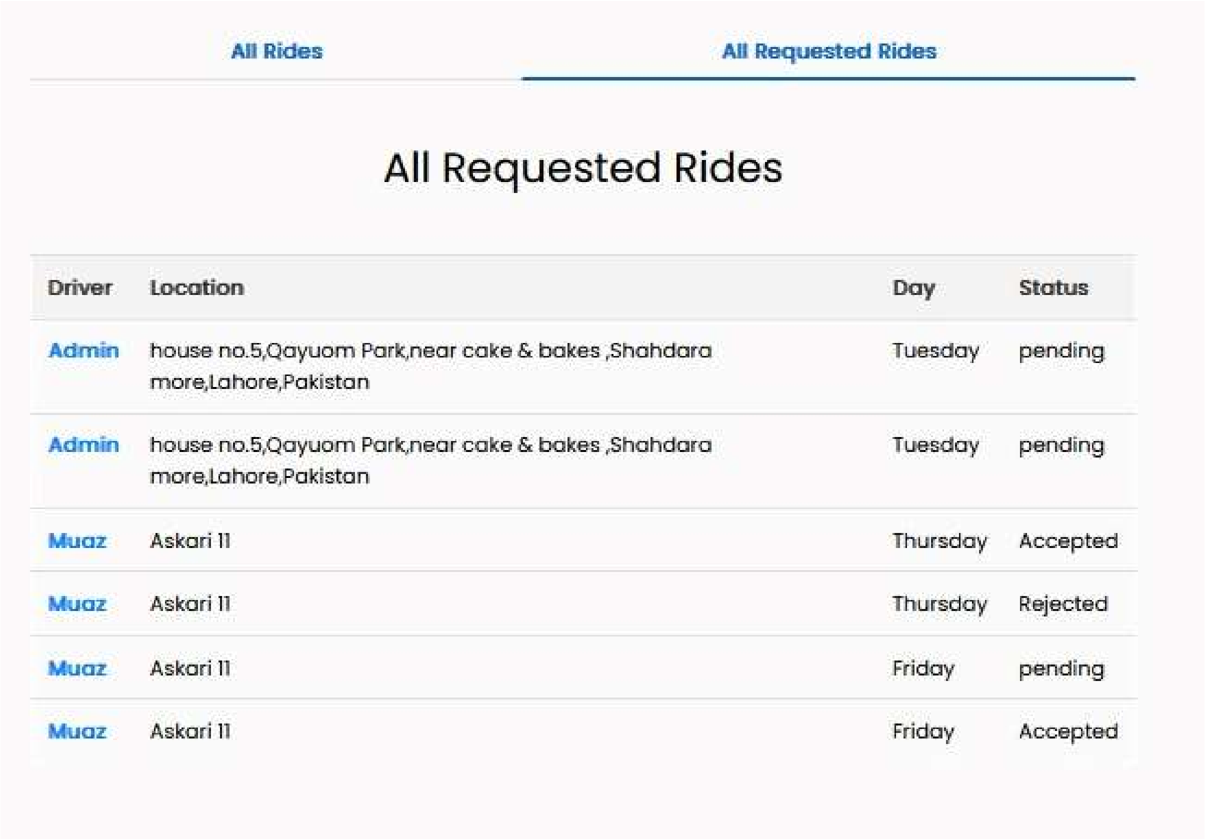
ADDRESS: DHA
Phone: 234234
Requested Ride Day: Thursday
Round Trip: No

Estimate Cost : 60 pkr

AFNAN

Figure 24: Ride Requests

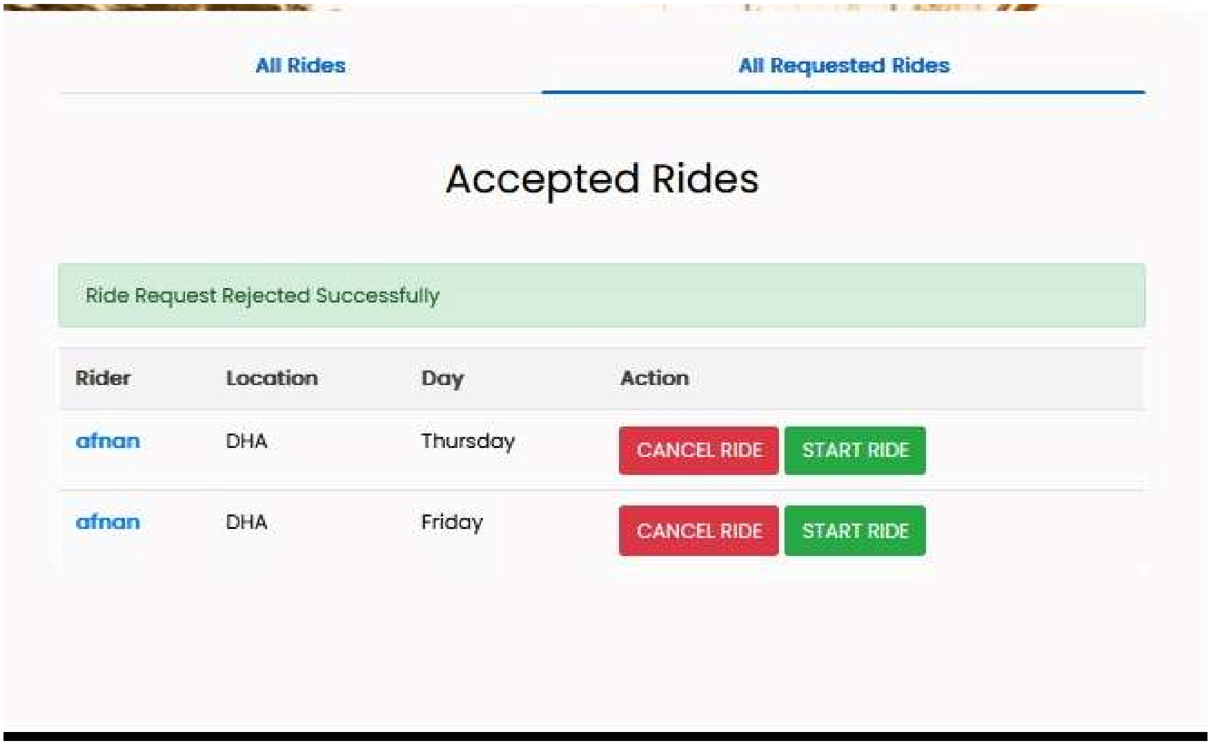
Figure 24 shows all requested rides



All Requested Rides

Driver	Location	Day	Status
Admin	house no.5,Qayuom Park,near cake & bakes ,Shahdara more,Lahore,Pakistan	Tuesday	pending
Admin	house no.5,Qayuom Park,near cake & bakes ,Shahdara more,Lahore,Pakistan	Tuesday	pending
Muaz	Askari 11	Thursday	Accepted
Muaz	Askari 11	Thursday	Rejected
Muaz	Askari 11	Friday	pending
Muaz	Askari 11	Friday	Accepted

Figure 25: All Requested Rides



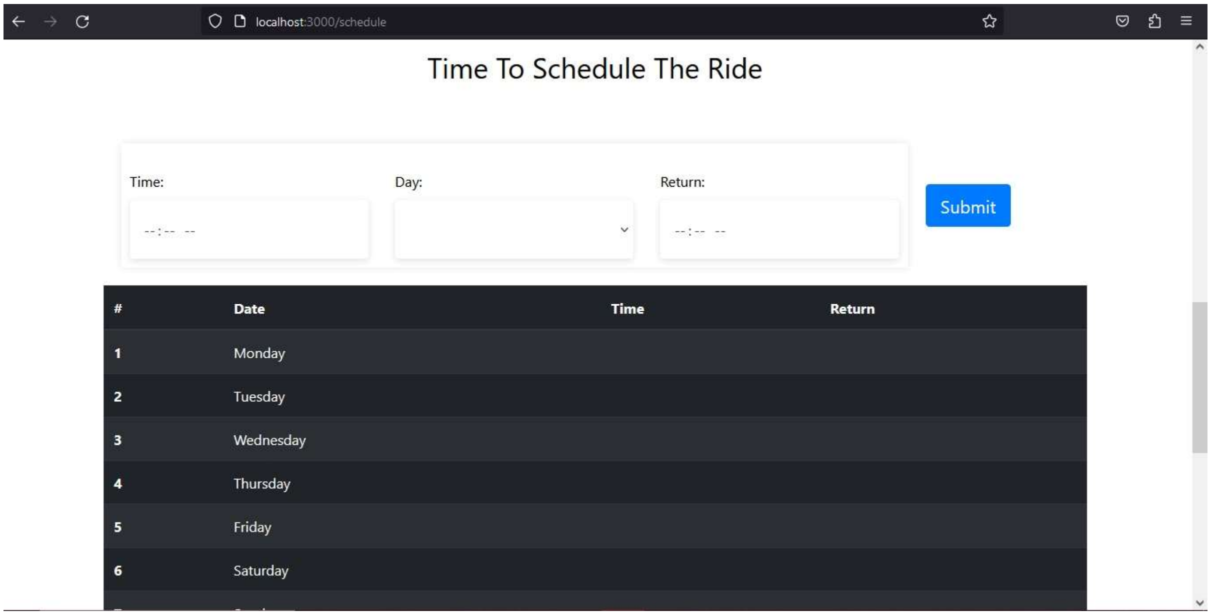
Accepted Rides

Ride Request Rejected Successfully

Rider	Location	Day	Action
afnan	DHA	Thursday	CANCEL RIDE START RIDE
afnan	DHA	Friday	CANCEL RIDE START RIDE

Figure 26: Accepted Rides

Driver scheduled a ride for the users or riders by setting date, time and day etc.



Time To Schedule The Ride

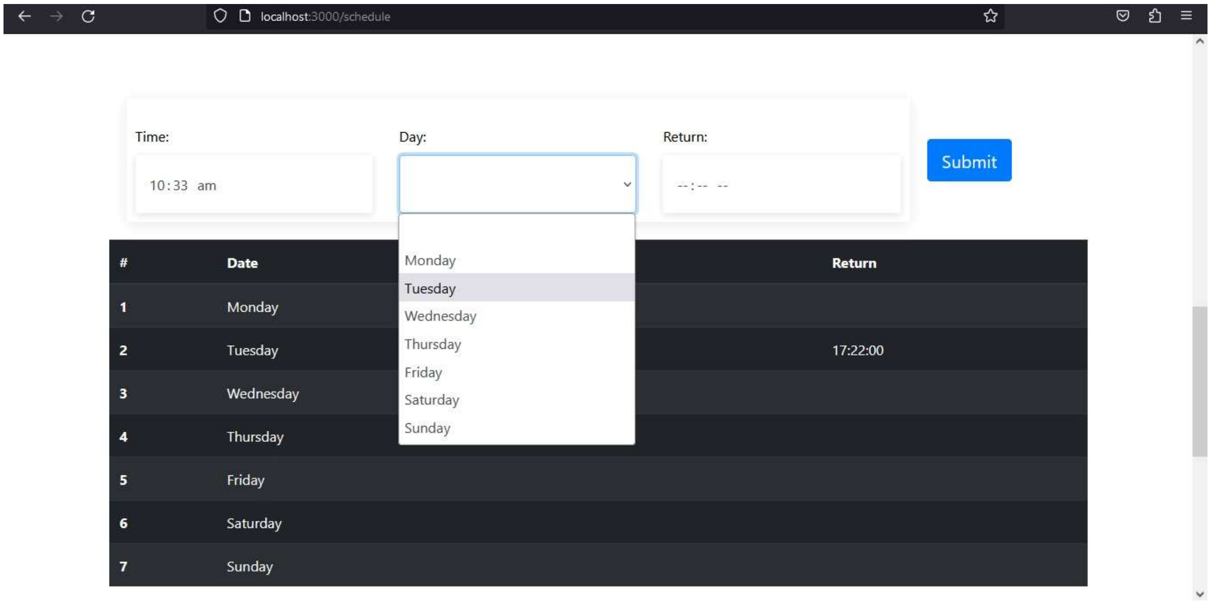
Time: --:-- --

Day:

Return: --:-- --

#	Date	Time	Return
1	Monday		
2	Tuesday		
3	Wednesday		
4	Thursday		
5	Friday		
6	Saturday		

Figure 27: Schedule Ride



Time: 10:33 am

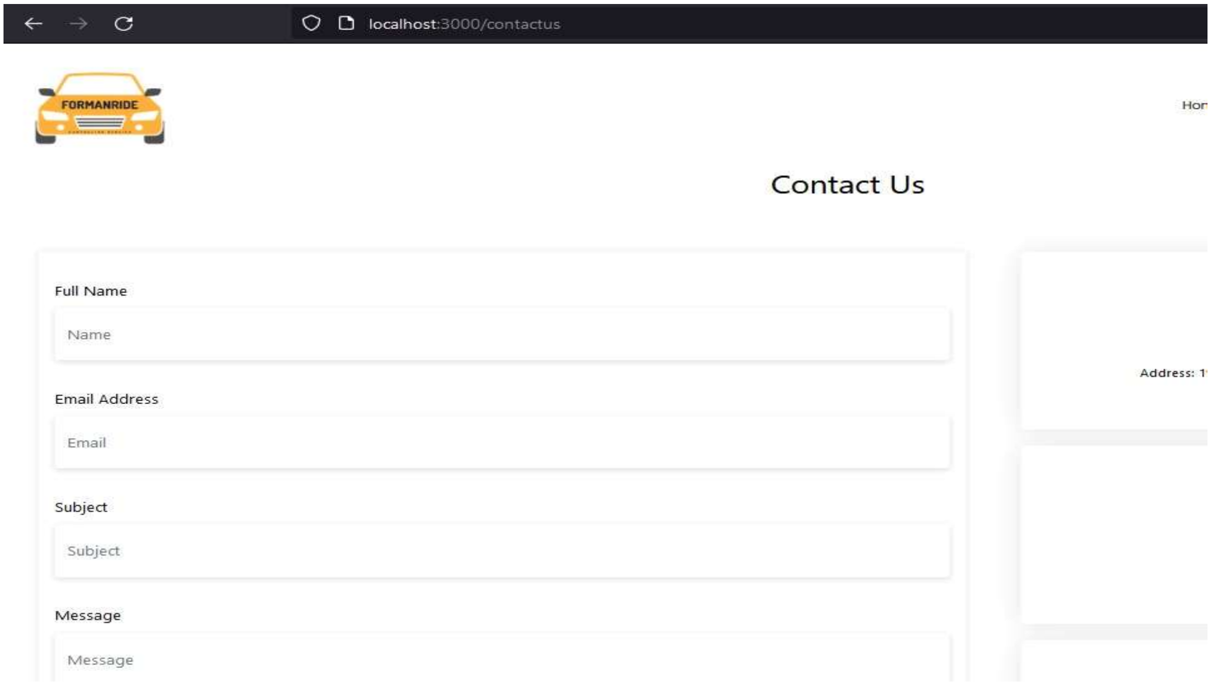
Day:

Return: --:-- --

#	Date	Return
1	Monday	
2	Tuesday	17:22:00
3	Wednesday	
4	Thursday	
5	Friday	
6	Saturday	
7	Sunday	

Figure 28: Schedule Ride

If the user wishes to convey a message or feedback on our website, they can go to our Contact Us page and send us the queries they have so that we can response back to their queries



Contact Us

Full Name: Name

Email Address: Email

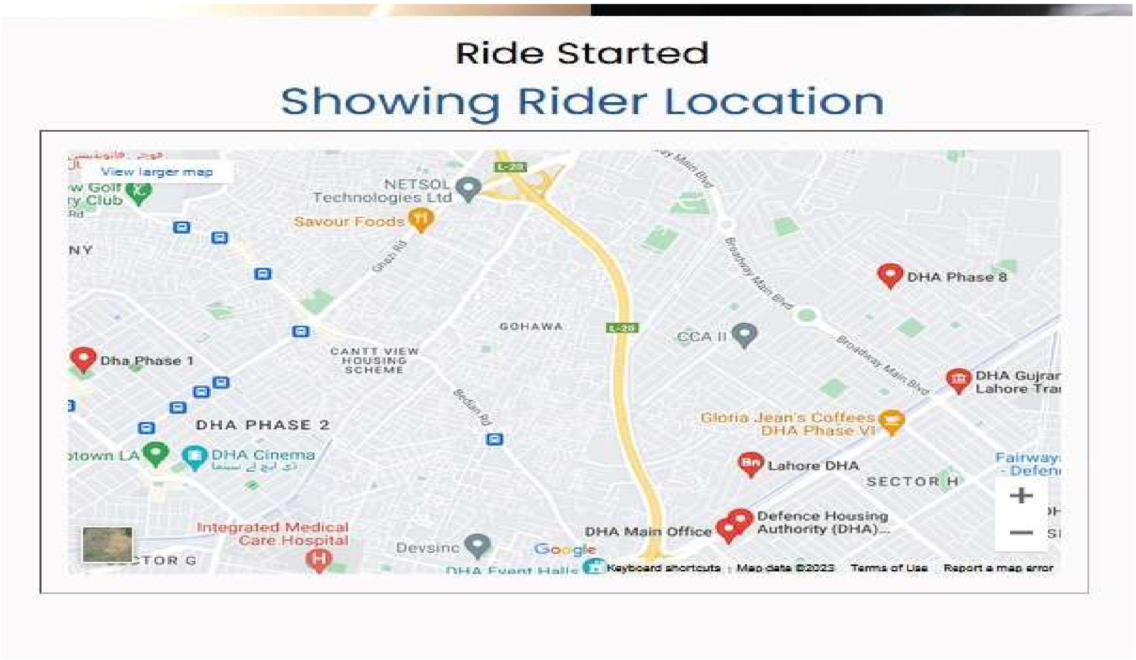
Subject: Subject

Message: Message

Address: 1

Figure 29: Contact Us

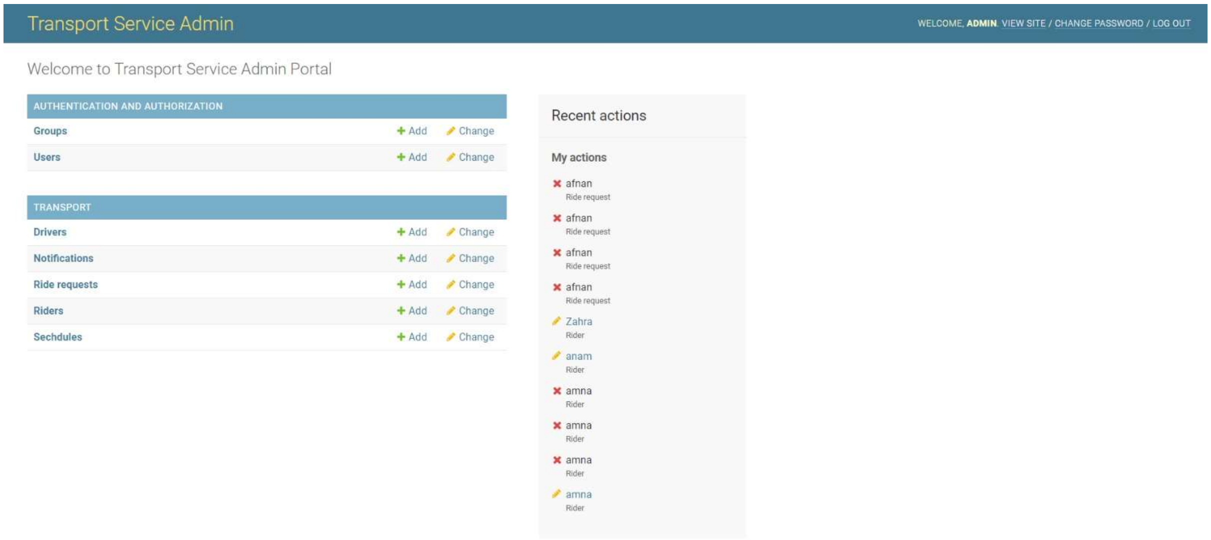
Figure 30 shows the rider location by API maps



Ride Started

Showing Rider Location

Figure 30: API Map



Transport Service Admin

WELCOME, **ADMIN**. VIEW SITE / CHANGE PASSWORD / LOG OUT

Welcome to Transport Service Admin Portal

AUTHENTICATION AND AUTHORIZATION		
Groups	+ Add	Change
Users	+ Add	Change

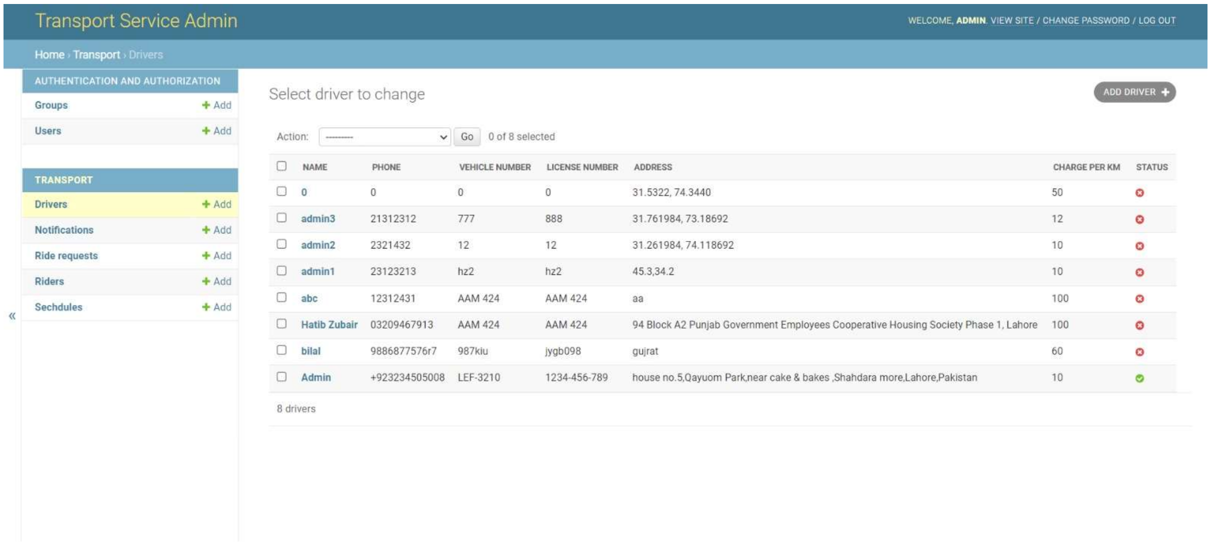
TRANSPORT		
Drivers	+ Add	Change
Notifications	+ Add	Change
Ride requests	+ Add	Change
Riders	+ Add	Change
Sechdules	+ Add	Change

Recent actions

My actions

- ✖ afnan
  Ride request
- ✖ afnan
  Ride request
- ✖ afnan
  Ride request
- ✖ afnan
  Ride request
- ✎ Zahra
  Rider
- ✎ anam
  Rider
- ✖ amna
  Rider
- ✖ amna
  Rider
- ✖ amna
  Rider
- ✎ amna
  Rider

Figure 31: Admin Panel



Select driver to change

0 of 8 selected

□	NAME	PHONE	VEHICLE NUMBER	LICENSE NUMBER	ADDRESS	CHARGE PER KM	STATUS
□	0	0	0	0	31.5322, 74.3440	50	☒
□	admin3	21312312	777	888	31.761984, 73.18692	12	☒
□	admin2	2321432	12	12	31.261984, 74.118692	10	☒
□	admin1	23123213	hz2	hz2	45.3,34.2	10	☒
□	abc	12312431	AAM 424	AAM 424	aa	100	☒
□	Hatib Zubair	03209467913	AAM 424	AAM 424	94 Block A2 Punjab Government Employees Cooperative Housing Society Phase 1, Lahore	100	☒
□	bilal	9886877576r7	987kiu	jygb098	gujrat	60	☒
□	Admin	+923234505008	LEF-3210	1234-456-789	house no.5,Qayuom Park,near cake & bakes ,Shahdara more,Lahore,Pakistan	10	✓

8 drivers

Figure 32: Admin Panel

3.7 Other Design Details

User-Friendly Interface:

The website should have an intuitive and user-friendly interface, with clear navigation menus and well-organized sections.

The design should prioritize simplicity and ease of use to ensure a seamless user experience.

Responsive Design:

The website should be responsive and compatible with different devices such as desktops, laptops, tablets, and mobile phones.

This will allow users to access the website and request/accept rides from any device, enhancing convenience and accessibility.

Registration and Login:

The registration process should have a clean and straightforward design, guiding users through the required fields for registration.

Login should have a prominent and easily accessible area on the homepage, allowing users to log in quickly.

Chapter 4. Test Specification and Results

4.1 Test Case Specification

Identifier	TC-1
Related requirements(s)	username, password
Short description	The input required here will allow the user login and use the website
Pre-condition(s)	All the required fields should be filled.
Input data	Shahmeer,23-145783
Detailed steps	The user must input the required fields according to the data.
Expected result(s)	Login
Post-condition(s)	Inputs should be available in the database.
Actual result(s)	Login
Test Case Result	Pass

Table 12: TC-1 Login

Identifier	TC-2
Related requirements(s)	Username, password, confirm password.
Short description	The input required here will let the user register itself.
Pre-condition(s)	All the required fields should be filled.
Input data	Asfand Rana,23-145783,23-145783
Detailed steps	The user must input the required fields according to the data.
Expected result(s)	Sign up
Post-condition(s)	All inputs will be saved in the database.
Actual result(s)	Sign up successfully
Test Case Result	Pass

Table 13: TC-2 Sign up

Identifier	TC-3
Related requirements(s)	Send Request for the Ride
Short description	The input required here will send request for the ride
Pre-condition(s)	1. User must have access to the web app by entering URL 2. User must have access to the internet.
Input data	User have Requested the Ride
Detailed steps	The user must input the required field.
Expected result(s)	Suitable Ride for Carpool
Post-condition(s)	Driver must Accept the ride request.
Actual result(s)	Ride found.
Test Case Result	Pass

Table 14: TC-3 Rider

Identifier	TC-4
Related requirements(s)	Accept Request for Ride
Short description	The input required here will Accept Request for Ride.
Pre-condition(s)	Rider must be logged In.
Input data	Rider have Send Request for the Ride.
Detailed steps	The user must input the required field.
Expected result(s)	Get suitable rider.
Post-condition(s)	Rider must send request for the Ride.
Actual result(s)	Start Ride
Test Case Result	Pass

Table 15: TC-4 Driver

4.2 Summary of Test Results

Module Name	Test cases run	Number of defects found	Number of defects corrected so far	Number of defects still need to be corrected
Module 1 (sign-up)	TC-2	0	0	0
Module 2 (log-in)	TC-1	0	0	0
Module 3 (Ride Found)	TC-3	0	0	0
Module 4 (Carpool)	TC-4	0	0	0
Complete System	TC-1, TC-2, TC-3, TC-4.	0	0	0

Table 16: Summary of All Test Results

Chapter 5. Conclusion and Future Work

5.1 Project summary

Ride sharing is increasingly popular in developed nations due to the rising costs of daily commutes, especially for students who face financial burdens. Cars, as major contributors to air pollution, exacerbate environmental issues. To tackle these challenges, we have developed a React.js web app. This app serves as a carpooling platform exclusively for FCCU students, enabling them to share rides, split fuel expenses, and significantly reduce their financial burdens. Additionally, the webapp addresses the issue of limited parking space by connecting students with similar destinations, despite their differing schedules resulting from FCCU's liberal arts education. By providing an innovative and user-friendly platform, our solution promotes cost savings, environmental sustainability, and a seamless commuting experience for FCCU students.

Problems faced and lessons learned

The problems we faced were during development. We had different framework for front-end and back-end so integrating both was a big task for us as we had not done it before. It took proper dedication to reach our goal. Another problem was the mapping functionality using the google maps API. It was very difficult to integrate in the application as the cost increases by using certain API's. In the end, deployment of the application took time as no one was experienced enough to do it. We had to ask people to know what to do and were successful in the end.

The lessons we learned during this time were life changing. For one we understood the importance of patience as our patience was tested multiple times during this project. The major lesson we learned was the framework we used. We had not used React and Django before so it was a completely new experience. We understood the value of teamwork and work division as it helped us reach our goal and complete our project. This project taught us many things on what to do as coders and what to do as a team player. This project will help us in future when we set out in professional life's. This is a big stepping-stone for us.

5.2 Future work

- This version of the carpool system only consisted of being deployed on a website.
- However, it lacks the edge of being on android and iOS. That would be the first thing to work on in the future for this project.
- Reach out to other university students.
- To make it commercially available outside of university specific and to public.
- Getting premium API's from Google, just like Uber, Bykea and Careem, for live tracking functionality which was not offered due to low scale nature.

References

[1] Goncalo Correia, Jose Manuel Viegas: Carpooling and carpool clubs: “Clarifying concepts and assessing value enhancement possibilities through a Stated Preference web survey in Lisbon, Portugal”, Transportation Research Part A 45 (2011) 81–90,ScienceDirect.

[2] Seyedehsan Seyedabrishami, Amirreza Mamdoohi, Ali Barzegar, Sajjad Hasanpour:” Impact of Carpooling on Fuel Saving in Urban Transportation: Case Study of Tehran.” Procedia - Social and Behavioral Sciences 54 (2012) 323 – 331, Sciverse ScienceDirect.

[3] Luk Knapena, Daniel Keren, Ansar-Ul-Haque Yasar, Sungjin Cho, Tom Bellemans, Davy Janssens, Geert Wets: “Estimating scalability issues while finding an optimal assignment for carpooling.”, Procedia Computer Science 19 (2013) 372 – 379 Sciverse ScienceDirect

[4] Joao Ferreira, Paulo Trigo and Porfirio Filipe: “Collaborative Carpooling System”, World Academy of Science, Engineering and Technology 54 2009.

[5] Maurizio Bruglieri, Diego Ciccarelli, Alberto Colornia, and Alessandro Luè: “PoliUniPool: a carpooling system for universities.” Procedia Social and Behavioral Sciences 20 (2011) 558–567 ScienceDirect

[6] Shangyao Yan, Chun-Ying Chen, and Sheng-Chieh Chang:”A Car Pooling Model and Solution Method With Stochastic Vehicle Travel Times” IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS 1

[7] Rajesh Kannan Megalingam, Ramesh Nammily Nair, Vineeth Radhakrishnan Amrita Vishwa Vidyapeetham, Amritapuri, Clappana P.O, Kollam-690525, Kerala, India:” Automated Wireless Carpooling System for an Eco Friendly Travel.”

[8] Mario Collotta, Giovanni Pau, Valerio Mario Salerno, Gianfranco Scat`a Kore University of Enna – Italy:” A NOVEL TRUST BASED ALGORITHM FOR

CARPOOLING TRANSPORTATION SYSTEMS” 2nd IEEE ENERGYCON Conference & Exhibition, 2012

[9] C´edric Bonhomme, G´erald Arnould and Djamel Khadraoui Public Research Centre Henri Tudor 29 Avenue John F. Kennedy, L-1855, Luxembourg:” Dynamic Carpooling Mobility Services based on Secure Multi-Agent Platform” GIIS'12 1569690819.

[10] Gérald Arnould, Djamel Khadraoui CRP Henri Tudor 29, Avenue John F. Kennedy L-1855 Luxembourg (Luxembourg) Marcelo Armendáriz, Juan C. Burguillo, Ana Peleteiro Dep. of Telematic Engineering University of Vigo 36310-Vigo (Spain):” A Transport Based Clearing System for Dynamic Carpooling Business Services” 2011 11th International Conference on ITS Telecommunications

Appendix A Glossary

Terminologies	Descriptions
Website	A set of web pages under a single domain name; can be used by anyone with internet access.
UC	Use Case
DFD	Data Flow Diagram
Admin	FCCU Carpool Service's website admin. Someone who manages all the frontend and backend of the server and ensures smooth functioning of the website.
User	The person who uses the FCCU Carpool Service website. Any human being could be a user.
ERD	Entity Relationship Diagram
HTML	Hypertext Markup Language
SQL	Structured Query Language

Appendix B Deployment/Installation Guide

All that a user needs to use our website is internet connection and a device to use the world wide web.

Simply typing in our URL will take the user to our website.

Appendix C User Manual

The user would require an internet connected device, this could mean any computer, laptop or mobile can be used. The user would type in the URL of our website and will go to our main page. Scrolling down would show our Carpool Service. User can Login to the website as a rider or as well as a driver.

Appendix D Student Information Sheet

Roll No	Name	Email Address (FC College)	Frequently Checked Email Address	Personal Cell Phone Number
231485855	Afnan Ahmed	231485855@formanite.fccollege.edu.pk		
241547206	Hatib Zubair	241547206@formanite.fccollege.edu.pk		
231485432	Muhammad Muzammil	231485432@formanite.fccollege.edu.pk		

Appendix E Plagiarism Free Certificate

This is to certify that, I am Afnan Ahmed S/D/o Amir Hassan Qureshi, group leader of FYP under registration no _________________ at Computer Science Department, Forman Christian College (A Chartered University), Lahore. I declare that my Final year project report is checked by my supervisor and the similarity index is ________% that is less than 20%, an acceptable limit by HEC. Report is attached herewith as Appendix F. To the best of my knowledge and belief, the report contains no material previously published or written by another person except where due reference is made in the report itself.

Date: _____________ Name of Group Leader: __________________________ Signature: ______________

Name of Supervisor: _______________________ Co-Supervisor (if any):_____________________
Designation: _______________________ Designation: _______________________
Signature: _______________________ Signature: _______________________

Senior Project Management Committee Representative: _______________________
Signature: _______________________

Appendix F Plagiarism Report