# “LIFT LINK: SHARING RIDES, REDUCING EMISSIONS (CARPOOLING)”

*(for the partial fulfillment of* Bachelor of Technology Degree in Computer Science & Engineering / Information Technology)

*Submitted by*

**VAIBHAV RAUTELA**

**Under the guidance of**

***Mr. Chandradeep Bhatt***

*DESIGNATION*



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING / INFORMATION TECHNOLOGY**

**GRAPHIC ERA HILL UNIVERSITY**

**JUNE, 2024**

**CERTIFICATE**

This is to certify that the thesis titled **“car-pooling system” submitted** by Vaibhav Rautela, to Graphic Era Hill University for the award of the degree of **Bachelor** **of Technology**, is a bona fide record of the research work done by him/her under our supervision. The contents of this project in full or in parts have not been submitted to any other Institute or University for the award of any degree or diploma.

Chandradeep Bhatt

project Guide

Place: Dehradun (Designation)

Date: 29-06-2024

GEHU, Dehradun

**ACKNOWLEDGMENT**

I would like to express our gratitude to the almighty, the most beneficent and the most merciful, for successful completion of major project. I am pleased to acknowledge my sincere thanks to GRAPHIC ERA HILL UNIVERSITY for their kind encouragement in doing this project and for completing it successfully. I am grateful to them. I wish to thank my parents for their continuing support and encouragement. I also wish to thank them for providing me. Thank you for providing me with the opportunity to reach this far in our studies.

I would also like thank external supervisor for his patience, support and encouragement. I also acknowledge to my project supervisor “**Mr *Chandradeep Bhatt****” sir*for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

At last, but not the least, I greatly indebted to others who directly or indirectly helped me during this course.

VAIBHAV RAUTELA

B. TECH,

SEMISTER 8th SECTION M

SESSION 2021-2024

GEHU, DEHRADUN

**ABSTRACT**

India, with nearly the largest population of more than 1 billion people, can be used as a case study. 4 billion, and like many other developing countries, it is experiencing a great deal of pressure in relation to urbanization of the population and the pressure it puts on transportation systems. In line with this, as more urban developments are being put up and as more people fill the metropolis, the need for mobility increases which results to more and more consumption of nonrenewable energy resources like the oil, uranium, natural gas and coal. This increase in the resource use even worsens environmental issues which accounts for most of the pollution /greenhouse gas emissions. As environmental concerns rise and roads in cities filled with more cars than ever, the issue of efficient and environmentally friendly transport have gained paramount significance. Of particular interest in the situation, is carpooling which refers to the process whereby several people travel together in a single vehicle for a mutual trip. To some degree, it assists in the reduction of traffic jams, emission of carbon, and improvements in connectedness.

Carpooling not only reduce the number of vehicles on road leading to free movers on the road but also reduce the general pollution, thus contributing to free air, eradicating pollution agents that are us causing climate change. Altogether, it also saves on the utilization of more resources by eliminating the need for manufacturing and maintaining cars and has implications for the economy because of its potential to save on fuel expenses, car maintenance and space for parking. Communally, carpooling encourages interaction between commuters as well as facilitates the creation of bonds in the society. This paper elaborates on the design and implementation of a sophisticated carpooling system that is designed to optimize available transport links and, at the same time, reduce the impact of the transportation sector on the environment.

A noteworthy aspect of the system is that it is easy to use in order to join a group for carpooling, the smart algorithm used to match individuals for the purpose of ride-sharing, the real-time tracking that is used to ensure that individuals using the system are as reliable as possible, the use of incentives to encourage people to sign up for the program and last but not least it is equipped with security measures to ensure that the consumers are safe when Thus, integrating these components, the proposed carpooling system aims at shaping a more sustainable transportation system, which corresponds to the current needs and complements existing solutions to tackle the problem of traffic congestion and environmental pollution in cities and provide more tangible incentives to both the client and society as a whole.

As a form of transportation, carpooling, which entail sharing rides in a vehicle, remains an effective solution to the transport complications in cities. It explores the potential advantages of carpooling, the main issues that might arise in implementing carpooling scales and methods of addressing these issues. Many experts have tried to cover up advantages of carpooling through various literatures, case studies, and statistical evidences in an effort to seek an approval to its economic, environmental and traffic virtues.

***Table of Contents.***

**CERTIFICATE2**

**ACKNOWLEDGEMENT3**

ABSTRACT4

CHAPTER 1

1.1 Problem Statements17

1.2 Objective18

1.3 Applying Software Engineering Approach19

1.4 Communication 20

1.5 Methodology 22

1.6 Working of API 24

1.7 Ide used in project 25

CHAPTER 2

2.1 Literature Survey26

2.2 Rational of Study27

2.3 Objective of Study28

2.4 Review of Literature28

2.5 Optimization Problems30

CHAPTER 3

3.1 Functional Requirement33

3.2 Software Requirement33

3.3 Hardware Requirement34

3.4 Non- Functional Requirement 34

CHAPTER 4

4.1 Existing System35

4.2 Proposed System36

4.3 System Requirement Specification36

4.4 Objective38

CHAPTER 5

5.1 System Flow diagram39

5.2 ER Diagram40

5.3 Data Flow Diagram41

CHAPTER 6

6.1 IoT Parking44

6.2 Motivation for Project45

6.3 Challenges Faced46

6.4 Methodology of Car parking System47

CHAPTER 7

7.1 Scope of Project49

CHAPTER 8

8.1 Principles of Testing50

8.2 Steps for Software Testing51

8.3 Types of testing51

CHAPTER 9

9.1 Result and Applications55

9.2 Costing of Project66

CHAPTER 10

10.1 Conclusions & Future Scope71

CHAPTER 11

11.1 References 74

CHAPTER 12

12.1 Appendix 76

***List of Tables.***

9.1 Tables showing values for COCOMO67

9.2 Table showing value for Working Hours69

9.3 Human Costing69

9.4 Total Cost70

***List of Figures.***

1.1 Incremental Model19

5.1 System Flow Diagram38

5.2 Entity Relationship Diagram39

5.3 Zero Level Data Flow Diagram 42

5.3 One Level Data Flow Diagram 42

5.3 Two Level Data Flow Diagram 42

6.1 Wireless Connection of Arduino Uno48

9.1 Registration Form56

9.2 Log-in Form57

9.3 Client Log-in58

9.4 User’s Dashboard59

9.5 Showing Bookings59

9.6 Search Ride 60

9.7 Rider Confirmation61

9.8 Confirm Booking61

9.9 Reset Password62

9.10 Showing Confirm Form63

9.11 Book Ride pages 64

9.12 Track user location’s 64

9.13 IoT Sensors65

9.14 Parking Working Model66

## 

***ABBREVIATIONS.***

COCOMO Constructive Cost Model

KLOC Kilo Lines of Code

SLOC Source Lines of Code

E Effort Applied

DT Development Time

PR People Required

ER Entity-Relationship

SF System Flow

DFD Data Flow Diagram

DF Data Flow

SDLC Software Development Life Cycle

VS Visual Studio

HTML HyperText Markup Language

Css Cascading Style Sheets.

***NOTATIONS.***

^ on power

+ addition

* Subtraction

\* Multiplication

() Bracket Of

# CHAPTER 1

# INTRODUCTION

A carpooling system is very advantageous because it helps to minimize pollution and traffic problems in cities, in addition to providing an environmentally friendly means of transportation. It also promotes interaction with other people since one can come across new people. Due to the delay and discomfort associated with public transportation and with a large number of people preferring their private vehicles, carpooling is a secure way as it involves pre-registration of the riders, thereby avoiding contact with strangers. It can also choose particular days for the registered users to use the vehicles to reduce embarrassment to individuals who commute daily. With the earth’s population rising, transportation also rises, thus contributing to pollution. A carpooling system can greatly reduce the impact of pollution and traffic congestion, enhance user interaction, and preserve natural resources by solving the problem of fuel shortages. Implementing such a system on a global scale might help to save natural resources and be a boon to the environment. The application is a carpooling platform designed to connect drivers and passengers, facilitating ride-sharing through two main features: to intercity travel and check-in. It allows the users to post and register for intercity trips and the check-in system alerts drivers or passengers when they arrive at the meeting spot. Users can only access the app by first downloading it, creating an account, and filling in trip information, which will be saved in a Firebase. Regarding passengers, they can select their favorite drivers, while the drivers who want to work have to provide a valid license number and correct information. The app supports Android 3. 0 and its subsequent versions, makes for easy access in booking and managing of trips.

## The use of the application helps in increasing safety during long distance travelling since drivers can swap and avoid fatigue. It is developed using the Android Studio and contains two modules. The first module is intended for drivers, where they should enter personal and car information. The second module is for passengers who also must provide their details to the driver and inform him who will be with them. In case one has the other person’s contact information, it is easy to arrange for a meeting point.

## Transportation is one of the most significant challenges that people face, especially when it comes to road transport. Most road transport is done by single passenger cars resulting to congestion, pollution, wastage of time and more parking space to be provided. The current transportation solutions are known to be area-specific. To counter these problems, a new system has been created and it is designed like a social network and available to people through smartphones and fast internet. Safety and security are the major concerns hence extra additions like an internet-based GPS tracking system through which the users and a central controller of the car get the information of the car’s exact location and this particular feature is deactivated once the user has reached his/her final destination. This carpooling system has several benefits which include: Traffic congestion and other related problems are avoided. Our system has been designed in a way that it helps solve several problems related to transportation by encouraging people to share rides and optimizing the number of people per car. In relation with sharing of vehicles, we expect there to be a decline in the number of vehicles on the roads hence less traffic jam. Moreover, carpooling can considerably bring down the overall number of Kilometers on the road by the individual vehicles and the fuel consumed, thereby help in conserving the environment.

The main goal of our carpooling system includes the optimization of the occupancy of vehicles, with an aim of reducing empty seats. This not only enhances the utilization of the existing resources but also the parking space, as fewer cars are needed in the end. This has made our system easy to use where a user does not have to do much to look for or share a ride. The strength of our system is manifested in the possibility to select the most optimal route out of many passengers who are ready to share the ride. Unlike previous systems, the users are pulled a list of available cars or drivers, the system calculates the longest common route, which ensures that each of the parties using the service benefits.

To safeguard users, our system includes privacy measures and utilizes location tracking through GPS to show the map of passenger and driver locations in real-time time. This means that drivers are able to navigate to areas where passengers have booked and pick them up safely reducing insecurity among users.Signing up is compulsory for ride hailing, which adds layers of security measures to the process. Also, our system provides the interface to show other users’ recommendations of the most popular drop-off points, which constantly updates the information and improves the general result of the interaction. These issues have been outlined, and it is believed that with the strong security features incorporated in the system and a friendly user-interface, the carpooling system will go a long way in changing the current face of transportation.

# Problem Statement

## Our modern society has further worse the situation with regards to traffic congestion since people are using their personal cars to even go for small runs. Through this, it has triggered some problems such as traffic jam and increased pollution in urban areas of the affected cities. There has also been an increase in numbers of vehicles and consequently an increase in the consumption of fuel energy which is classified under our non-renewable energy sources leading to high costs of fuels. To address such challenges, the development of a mobile-based carpooling system seems to be a solution worth implementing. This would be so since an efficiently designed such as a system of sharing of rides would ensure that the population density of cars on the roads was significantly reduced, thereby cutting down on traffic, and pollution. Nevertheless, the establishment of carpooling systems has challenges, more so with regard to safety and reliability within the user community. To meet these requirements, a carpooling system should assign high importance to the safety and security of the users. Incorporating sound security aspects and measures in its development to ensure confidence from users is a crucial factor for its success. In addition, the carpooling system should be able to accommodate both short and long-distance trips which covers the needs of users needing transport within urban areas and those that require transport to other regions or provinces. In conclusion, the above said mobile architecture for carpooling system implementation shows a possible solution to the traffic congestion problem, air pollution, and associated environmental issues.

## **Objective**

Our project aims at tackling cases of traffic congestions by encouraging the use of shared transportation means in order to reduce the number of transport vehicles on the road. Our carpooling system is an online solution allowing users to reach their destinations safely and securely. News updates can be both for the registered and non-registered user and offer up to date information. In this system, users can search for available vehicles in the pooling system and request for bookings by signing up through a login form.It handles customer information, driver information, vehicle information and details, ride payment, and booking managing the work that would otherwise require lots of paperwork. Our system also improves the transactions of businesses through automation of processes to save time and money that would have been used in collecting information. This can in turn be used to ensure that resource utilization is enhanced in the right manner. Unlike our competitors who provide carpooling solutions during limited hours, our system is web-based and functional round the clock provided there are no server problems.

1. It functions well and can be easily modified.2. A product that is easy to comprehend and navigate while reducing the time and amount of capital needed for completion.3. Environmentally conscious system.4. Available system always, enhancing service delivery time.

## **Applying Software Engineering Approach**The purpose of systems design is to design an effective model of representing the usefulness and satisfaction. It provides information regarding the domain of application of the software being developed. It accurately describes the internal workings of each of the software components. undefined1. The incremental model involves iterative cycles: needs, architecture, implementation, and evaluation.2. It focuses on creating and delivering the software fast within the entire software development life cycle.3. Provides the versatility and affordability to accommodate fluctuating demands and the size of the project.4. Enables the integration of feedback from the client into the work being done, and also allows for the identification of errors with each turn.5. This makes testing and debugging processes within small segments much easier to conduct.

## 

## **Communication**The first one involves the user initiating contact with the service provider to seek a particular software product. During this phase, the user convinces the organization by explaining the terms and formally writes a request to the service provider organization.

* **Planning and Requirement Analysis**This phase is very important in the Software Development Life Cycle (SDLC) because it sets the context for the overall project. Requirement analysis is performed by the senior team members where detailed input is taken from customers, the sales team, market research, and domain specialists. It is to gain as much insight into the customer as possible and to record it in order to proceed to the following steps.
* **Designing and Modeling**This involves employing the requirements documented in the Software Requirements Specification (SRS) to create various design solutions for the product structure. These design alternatives are analyzed in terms of their feasibility and efficiency, and the optimal solutions are described in the Design Document Specification (DDS). This stage makes sure that the software system design is properly done in order to achieve the intended objectives**.**
* **Construction**This stage is also referred to as the programming phase during which actual coding of the software takes place. It is often said that testing should account for half of the total software development lifecycle. During coding, developers do some testing to ensure that bugs are detected and removed from the system. Further, efficiency testing specialists perform tests of multiple types, including module tests, program tests, product tests, internal tests, and user tests. This rigorous testing helps make the software dependable, and work as intended.
* **Deployment**Once the product has gone through all the testing procedures then it can be deployed. The implementation process entails officially introducing the product into the right market segment. Sometimes, a product is initially introduced to only a certain group of users to go through User Acceptance Testing (UAT) in an actual business setting. This means that final tweaking can be made in preparation for a mass launch after receiving feedback from the users**.**

## **Methodology**

* Html (Hyper Text Markup Language):

Both HTML and CSS are core web development languages. HTML is used to format the content of web pages, such as headings, symbols, lists, images, paragraph, forms, spreadsheets, and videos. It also provides the ability of using hypertext links for information retrieval. This structure is further complemented by CSS (Cascading Style Sheets) to give the appearance and design of a well-presented web page. HTML and CSS in combination makes it possible to develop dynamic web pages with scripting and graphic features.

* CSS (Cascading Style Sheets):

CSS (Cascading Style Sheets) is a key factor in the presentation of a web page. It determines the arrangement and structure of web pages as well as their appearance so that they can be viewed on a range of devices including large monitors, small monitors, tablets, and handsets. CSS makes websites look better and easier to navigate, and is equally usable on any display size and DPI.

* Javascript:

JavaScript remains one of the most essential frameworks when it comes to web development due to its popularity and easy usage. As a powerful extension of HTML, it stands as an enabler to help developers create engaging web documents. This language does not limit developers to a specific platform and is compatible with all programming systems which makes it easily accessible to all programmers. It is used in almost all tiers of web domain, ranging from frontend to backend where it can be used in creation of strong and responsive web apps.

* JQuery:jQuery is a lightweight JavaScript framework that is used to perform some common operations with webpage elements. It offers a series of functions that help the developers to interact with webpages, handle events, fetch data from the servers, generate effects and animations, and many things more. jQuery adds to the capabilities of browsers through the JavaScript DOM API thus making web development easy.
* Mysql:MySQL is an open-source relational database management system that uses SQL. It is used for the storage, retrieval, and manipulation of data within databases. MySQL enables activities like updating the database or deleting some information from it. It groups data into relations within relational databases, placing it in different tables. The features such as speed, reliability, and ease of use have made MySQL a popular choice. Since MySQL utilizes SQL, it can be integrated with different programming languages, such as PHP, Node.js, which makes Flask a versatile advantage for developers.
* Node.js:

Node. node.js is a JavaScript runtime that executes JavaScript scripts outside of the browser environment and is ideal for creating server-side applications for robust and efficient applications. It employs an event-driven, non-blocking I/O pattern, which makes the platform ideal for real-time services, such as chat applications or streaming services. Node.js is light and fast – using the V8 JavaScript engine from Google Chrome.

## WORKING OF APIIn as much as we are developers, we most of the time use APIs to get data or even use the services from other applications. APIs simply relay our requests to the corresponding applications for processing. It may also have restrictions, for example, the number of requests in an hour or a day.

## Using these APIs, a map can be also enhanced by adding features such as markers, locations, and directions between two places. Furthermore, by evaluating driving distances, formatting addresses, estimating postal codes, and searching for nearby locations, Google APIs enrich the capability and usability of the applications we develop.

## IDLE USED IN PROJECT

* VS CODE

VS Code is a fast-growing and advanced IDE that can be used for modern software development. It supports a number of programming languages and frameworks, which also makes it a versatile tool for web development, mobile applications development, and cloud applications development. They include syntax highlighting, intelligent code completions, debugger, version control, and extension for introducing new features.

**CHAPTER 2**

**LITERATURE SURVEY**

# LITERATURE SURVEY:

# There have been various carpool and ride-sharing concepts and systems in existence for the past few decades, yet none of them have been able to achieve global popularity, at least not until recently. This is mainly because many of them lacked scalability in their initial designs and hence the UX declined greatly with the growing number of users. Also, their compatibility with the most used mobile and desktop environments was significantly restricted, being available only for a tiny number of devices and browsers. Some of the options for future worldwide carpool and ride-sharing solutions in terms of design, distribution, and cloud computing are further presented by the authors in the paper. These strategies help to increase scalability and accessibility and should be further adapted to reach the global audience.Since current public transportation systems have failed to solve the increasing traffic and parking problems in urban centers, scholars are seeking out social solutions like ride sharing. This refers to an arrangement where a driver hires out his/her car to move from one area to another with one or more persons of similar destination. There has been proposed many carpooling systems, but most of them are not effective, do not have automated control, are not functional, and unpractical. Even though GAs have been used to solve various application problems, there is limited use of GAs in carpooling issues. In order to solve problems of many-to-many carpooling scenarios this paper presents a GA with an adapted fitness function.

# This function is to reduce travel distance, to match riders and drivers, to arrive on time, and to be as fair as possible to carpoolers if their preference is taken into account. Performance evaluations and simulations using real user data also show the efficacy of the presented approach and stress the importance of continued research in this field.

## RATIONAL OF STUDY:

## Carpooling and ride-sharing became popular in recent years, but it started way back in the mid-1970s in the US due to the 1973 oil crisis. As the cost of transportation increased, people started to car pool, especially employees who work in the same company. Nevertheless, carpooling reduced its usage in the 1980s because of low oil prices and the workday flexibility in the 1990s.In order to reverse this trend, the US government put in place measures of encouraging carpooling, which involves the construction of High Occupancy Vehicle (HOV) lanes on highways. These lanes were envisaged to decongest traffic and encourage carpooling, as the number of vehicles globally reached over 740 million in the year 2000, projected to rise beyond 2 billion in the year 2030. Some of the areas affected by the increase in the number of vehicles are; traffic, pollution, and parking. In order to tackle these issues and minimize transportation costs, the concept of a global real-time carpooling and ride-sharing system has been suggested. Still, though there may be a positive outcome, this work was done individually without the support of an organization.

## OBJECTIVE OF STUDY:1. To identify changes in carpooling systems over time.2. To establish a conceptual framework for understanding carpooling in the Indian context.3. To review the existing trends of carpooling in India.4. To examine the factors that restrain carpooling in India.5. In order to predict the likely future developments of carpooling systems.

## **REVIEW OF LITERATURE:**

## This has been as a result of increasing costs of environment and fiscally incurred from the use of single occupancy vehicles. Out of these, carpooling or ride sharing turn out to be potential opportunities, planning to exploit the unused space of personal transport vehicles. These practices not only make individual travelling cheaper by filling up otherwise vacated car seats but also help to decongest the road and decrease pollution. In addition, they provide social interaction hence enhancing social relations among the commuters. However, traditional carpool scheduling systems have always had problems regarding their usability. Elimination of schedule flexibility and set rider groups has been a strong turn-off to would-be participants.A survey conducted in 2009 regarding carpooling reluctance found out that issues of organizing carpools and inconvenience that comes with organizing carpools were among the main cause of reluctance among the respondents. Tackling these challenges calls for new ideas that take advantage of modern web technologies and data repositories that contain social and location information about users.In addition, for carpooling and ride-sharing solutions to work globally, they have to be available in different operating systems. This means not only the current mobile and desktop contexts but also the future technological contexts as well. Achieving platform independence is a major concern, however, it is crucial for making the systems widely available and accessible. In order to overcome these challenges and provide foundations for the future advancements in global carpooling and ride-sharing systems, this paper presents a conceptual framework.It comprises new design paradigms, logistic approaches, and cloud-based solutions that have been designed to improve the efficiency and utilisation of such systems. In this regard, the paper aims to provide knowledge on how to come up with the future transportation systems that address the dynamic demands of commuters across the globe.The subsequent sections of the paper proceed with a more detailed description of the related work, including the general design ideas and goals. More specific elaborations on the technical parameters of the implementation choices of a proof-of-concept prototype system are also presented as sub-sections. Moreover, the paper describes the proposed strategies for future studies and developments, which stress the further advancement of carpooling and ride-sharing.

## **Optimization Problems**Over the last few years, shared mobile vehicle services have rapidly grown; the top five car-sharing cities like Tokyo, Moscow, Beijing, Shanghai, and Guangzhou (China) boasting of more than 70,000 car fleets. Research in this field focuses on three primary areas: the location of stations, the size of the fleet needed and how to properly redistribute the vehicles to areas of need.2. 5. 1 Station LocationHence, it brings out the fact that the location of a station greatly influences the performance of car-sharing systems. Analytical models are constructed to estimate optimal locations for car-sharing stations, taking into account such factors as station attractiveness and demand. These models help the user to find nearby stations and also helps in preparing suggestions for the destination, thus improving the usability of the application.2. 5.2 Fleet DimensionA key decision in the operation of a car-sharing system is how many cars should be placed at each station in order to satisfy demand and sustain the system. In order to meet this challenge, a stochastic network flow model has been constructed, which defines a linear programming problem under certain conditions. This model also employs numerical data on past travel patterns to properly distribute vehicles, thus optimizing the system’s capacity and allowing vehicles to flow through the system smoothly.

2.5.3 Station Inventory:

Station inventory management is essential for determining the optimal number of vehicles needed at specific locations to meet demand in a car-sharing system. This involves analyzing the frequency of vehicle reallocation throughout the day and associated costs. The problem can be addressed through chance-constrained stochastic optimization or dynamic inventory system modeling. Simulation and local search methods can then be employed to mitigate imbalances across different stations, ensuring efficient vehicle distribution.

2.5.4 Vehicle Reallocation:

This has made car-sharing systems to become more popular in the last few years because of the flexibility that they offer. Passengers choose a certain station and cars are sent to the chosen station; it is also successful for both passengers and drivers if there is a high passengers’ flow through that route. However, the use of vehicles to meet this demand creates management challenges and increases the cost of managing the system. It is especially important in the carpooling systems as the even distribution of vehicles between stations proves to be vital; thus, the question of reallocation arises. Two major subtopics are of special concern with respect to system management problems and the ease of use of the system; the operator-based approach and the user-based approach both stress the need for appropriate reallocation of the vehicles for better operation of the system.

**CHAPTER 3**

**REQUIREMENT ANALYSIS**

# FUNCTIONAL REQUIREMENTS:

## Software Requirements:1) Platform: Windows 8/9/10/11Platform is a term referring to any hardware that is used to support an application or service.2) Language: Machine learning, Python, Html/CSS, Javascript, Node.js., SQL.A programming language is a formal language used to communicate instructions to a machine with a specific purpose of computing. It can be used to design programs to program the behavior of a machine or to describe algorithms.

## Hardware Requirements:

1) Processor: Ryzen 5 2400g DeutschA processor is the central part of a computer that executes basic instructions which control the operation of the computer.2) RAM: 8 GBIt is a form of computer data storage where the most needed program instructions are stored to make a system generally faster.3) Storage: 1 TBComputer data storage, or storage or memory, is computer components and recording media used to store data. It is a major function and one of the building blocks of computers.4) Monitor: 22”A computer monitors or a computer display is an output device that displays information to the user.

## Non-Functional Requirements:1) Person: Two persons first the driver and second the one who is sharing the car with the other person.

## **CHAPTER 4**

**SYSTEM ANALYSIS**

System analysis involves collecting and processing information; noting problems; and recommending improvements to the system. It requires the exchange of information between the users and the developers and is crucial in the system development cycle. This process involves evaluating the system, identifying inputs, identifying issues, and providing solutions. The proposed solutions are changed until they meet the needs of the users.

# Existing System:

The system developed by the user carpooling system is currently the only less selective option currently we cannot rely on the system we still need to enhance it.1. more distant from the user.2. user Description is limited.3. time-consuming process.4. This does not reach of users that is distant.

## Proposed SystemIn this model the rider/client has to register on the website with some personal information about themselves the app features two sides: one for those in search of a ride and another for those willing to share their vehicle with a traveler.

## System Requirement Specification4.3.1 Problem Statement:Thus, population growth is worsening environmental and resource problems. To address this, a carpooling system is required in order to minimize pollution and traffic within the community.4.3.2 Efficiency Requirement:The online carpooling system will help to minimize traffic congestion and emission of pollutants into the atmosphere.4.3.3 Reliability RequirementIt should ensure a secure operating space for customers and owners while performing all calculations without mistakes.4.3.4 Usability RequirementOne of the key aspects of the Android application is its ease of use and simplicity.4.3.5 Implementation RequirementThe system will be developed using Android Studio for the development of the user interface and XML and Firebase for the development of the back end of the system.4.3.6 Delivery RequirementThe whole system is expected to be delivered in four months with weekly check points from the project guide.

## **Objectives:**

Following are objectives: -

* Reduce overall vehicle usage on roads with low pedestrian populace density.
* This facility is likely to help to reduce concentration of traffic during rush hours.
* Decrease the one-passenger car usage by increasing carpooling.
* Campaign for non-conventional and environmentally friendly means of transport.
* Improve existing parking in densely populated areas.
* Reducing expenses where as some drivers were given responsibility of either reducing their expenses on fuel or splitting the expenses.
* Reduce the occurrence rate of different types of motor vehicles on the roads.
* Reduce pollution and carbon dioxide emissions to the atmosphere.
* Reduce stress that is most likely to be caused by driving a car.
* Build relationships and connections in the community.

**CHAPTER 5**

**MODELLING AND DESIGNING**

# System Flow Diagram: -

A flow diagram, which can be used interchangeably with a flowchart, is a form of graphic that displays a series of related steps. One of them is the System Flow Diagram, which depicts the major activity steps within a systematic flow.

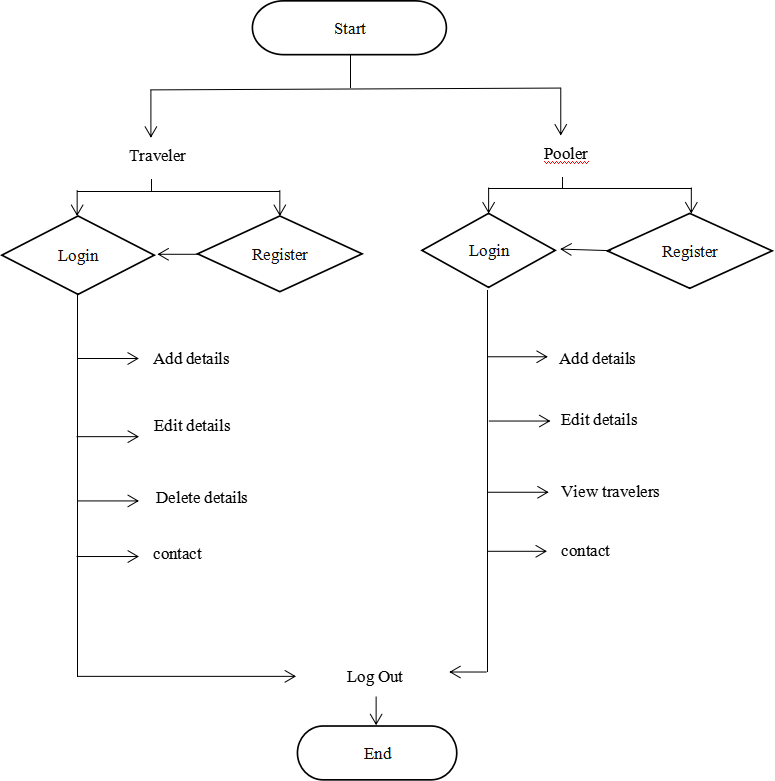
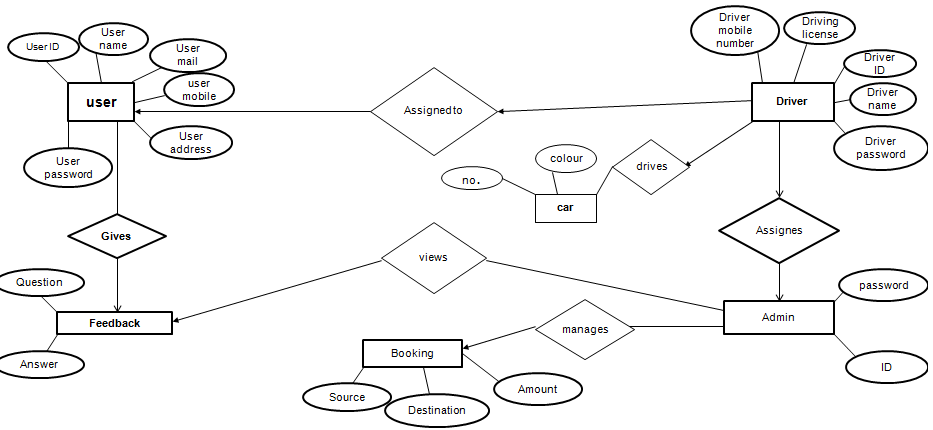


Fig 5.1 System Flow Diagram

## **ER Diagram: -**

An ER model, often shown as an ER diagram, illustrates entities and their connections, frequently applied in computing to structure data within databases or information systems. It outlines interconnected entities within a particular field, including entity types and their associations. These diagrams, often utilizing Chen's notation, are essential in software engineering to depict the business needs for data storage and manipulation. The ER model acts as a blueprint for organizing database structures, commonly implemented in relational databases.



**Fig.5.2: Entity Relation Diagram**

## **Data Flow Diagram: -**

A Data Flow Diagram (DFD) visually represents how data moves within an information system, capturing its process flow. It serves as an initial step to provide an overview of the system's structure, which can be further developed later. DFDs are useful for visualizing and understanding data processing within the system.

A Data Flow Diagram (DFD) is a structured instrument for analysis and design, similar to flowcharting. It portrays a network illustrating the flow of data and processes within a system, employing symbols that don't represent physical implementation. Its aim is to clarify system requirements and emphasize important changes, initiating the design process by dissecting specifications into detailed layers. DFDs provide an abstract representation of the logic in information or process-oriented systems, commonly referred to as logical data flow diagrams.

5.3.1 External Entity: -

An external entity within the context of a data flow diagram serves as either a source or destination of data flow. However, it's important to note that only those entities actively involved in either initiating or receiving data are represented within the diagram. These entities are symbolized by rectangular boxes, distinguishing them from other elements in the diagram. The rectangular shape is chosen deliberately to provide clarity and consistency in visual representation.

5.3.2 Process: -

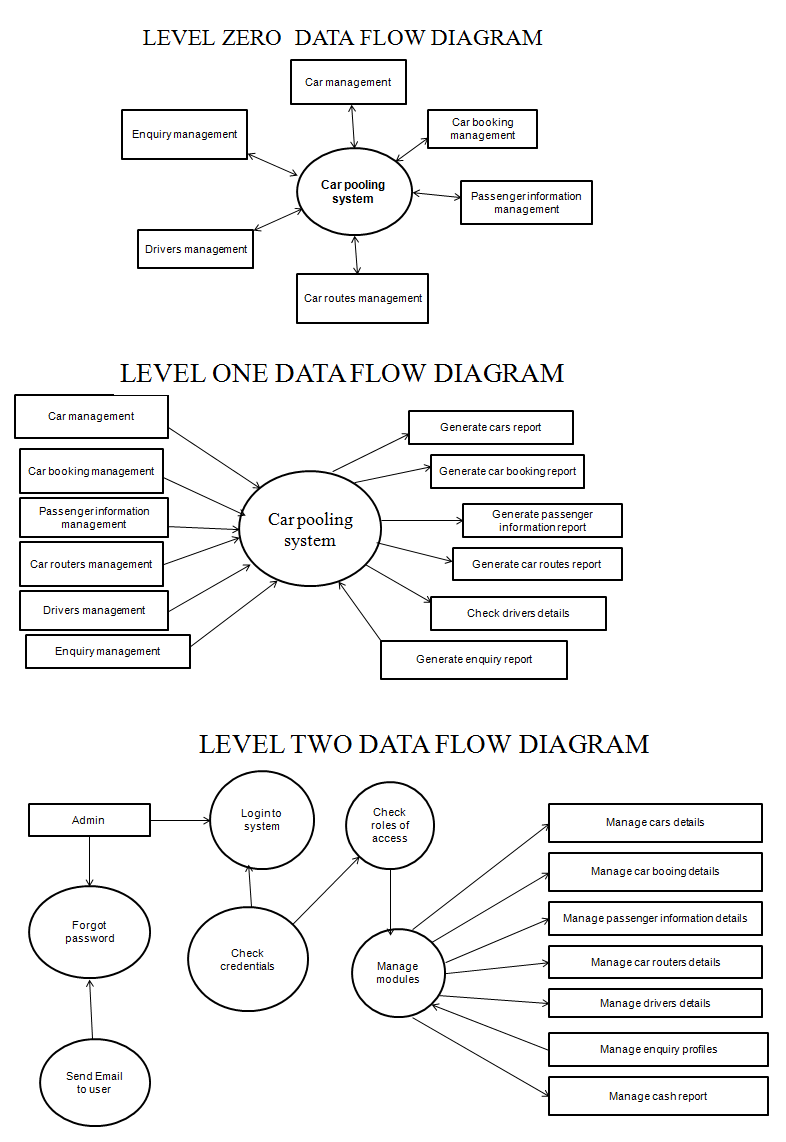
A process in a data flow diagram means a change of data within a system or data manipulation. It is represented by an oval sign, different from other signs used in the diagrams.

5.3.3 Data flow: -

The flow of data in a diagram represents the entire process through which certain information is transferred from one point to another. It is represented by a straight line with arrow heads showing the direction of flow. These are communication paths between processes and outside systems and may take form of textual, verbal or electronic data. A single data flow can be given names of the processes or data stores it links, or it can be characterized by its content.

## Data Store: -

A data store is an abstract concept which represents a storage location of information within a system and is represented diagrammatically by an open-ended rectangle with restricted width. It can also hold long-term documents like sales ledgers or short-term documents that are on their way to processing. It is important that each data store should have reference and be numbered so that it can be easily identified.



**CHAPTER 6**

**IOT PARKING**

A smart city integrates the usage of ICT in management of the city for a purpose of enhancing the efficiency in operations, informing the public and making the quality of service offered by the government better and the welfare of citizens enhanced. The general idea of smart city implies the enhancement of various elements and processes in cities with the help of smart technologies and improved data analysis which, in turn, contributes to the growth of the city’s economic efficiency and the quality of life of people living there. Here the essence is not how much of the pure technology is being incorporated but rather as to how this technology is being put to use.

The features which define smart city include cutting-edge innovation, a high-quality abode, new urbanization philosophy, new urban space, new urban form, advanced system, innovative economy, efficient organization, advancement, intelligent control, international perspective, sustainable development perspective, as well as efficiency, integration, humanization, computization, automation and Informa ionization.

* Basic infrastructure
* Develops smart solution to enhance infrastructure and services.
* It establishes area-based development as a central feature of governance.

# MOTIVATION FOR DOING THIS PROJECT

Smart City projects can in fact go even further and be used as strategic tools for advancing the economic growth of a region, which is what was attempted by the city of Nice, France in the Smart City sub-sector beginning from 2008. Its approach was to develop the sector using investments on the supporting infrastructure and offering of grants and incentives to attract more sector-specific investments that would spur the establishment of an economic hub, an industry cluster of excellence. It should therefore lead to the generation of $US 20 billion per year and create 100,000 sustainable workplaces in the long run, ultimately creating at least 20,000 workplaces within 20 years.

Apart from, economic argument of either reducing costs or enhancing revenues smart cities also include the following reasons why cities would wish to smart: Otherwise, to do with environmentalism, inclusion of smart cities to a city and attractiveness of a city to do business.Whether these rationales are altruistic or hedonistic, the Smart City market will continue to expand and how they are categorized shall change. Your city in the future: How “Smart” Are We Going to Get?

## **CHALLENGES FACED-**

**Challenge #1: Infrastructure**Smart Cities incorporate sensors to collect data as well as use the information collected for the sake of enhancing the wellbeing of residents. It involves statistics on traffic jam, criminal incidents, up to the general environment quality. High capital expenditure is required to lay down these sensors and even to maintain them.**Challenge #2: Security and Hackers**The use of IoT and sensor technology increases the intensity of risk to security as technologies grow. Even recent talks on cyber-terror threats to weak and antiquated power structures have compelled everyone to be a little more careful and skeptical about technology and safety.**Challenge #3: Privacy Concerns**In any major city it is a numbers game between the quality of life that can be enjoyed and the amount of privacy loss that has to be tolerated. However, if the people wish to live a more convenient and healthier life, free from violent conflicts, no one wants to feel like they’re being watched by “Big Brother” all the time.**Challenge #4: Social inclusion is a crucial factor that is an essential component of any society.**Smart transit programs where passengers are notified of each mode of transport available are ideal for a busy city. What about the population of that city where half of the people cannot even pay for mass transit or Uber? But what about an ageing population, which has little to no interaction with mobile devices or apps? How would smart technology get to these group of people and how would it impact on them?

## Methodology of car parking system: -

In this project we will be using IR (infrared) sensors to identify that whether there is a vehicle present within its sensing range or not. If the vehicle is detected then a LED will show indicating that the slot is full and if there is no obstacle(vehicle) then led will glow for showing that the slot is available. This availability of slots i.e. the output of the IR sensors, will be displayed in the LED. Whereas, we will be using the IR sensors to control the power and costing at the same time by Appling sensor in the bulb. And will be analyze weather the person is in connection with the sensor and help to make our life comfortable. In addition to this, we will use two more IR sensors and servo motor in order to let the vehicles in and out of the parking area. One IR sensor will be placed after the servo motor to identify the vehicles approaching the parking area and the other IR sensor will be placed before the servo motor to identify the vehicles leaving the parking area. And if the vehicle is detected by first IR/second IR, the servo motor will rotate up to 90 degrees in order to open the barrier to allow the vehicles in/out. However, when the vehicles are detected by both the IR sensors at the same time, the servo motor will not rotate and this means that the two vehicles might not collide.

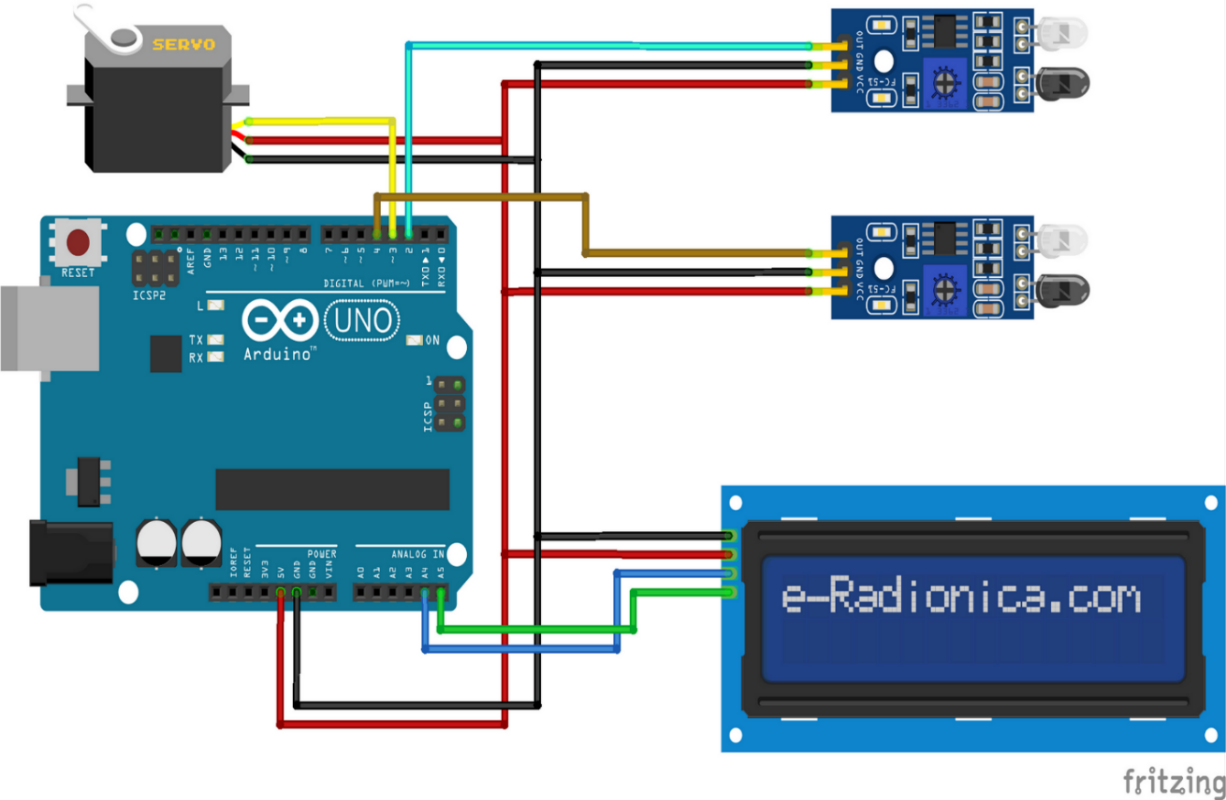


Fig 6.1- wires connections

**CHAPTER 7**

**SCOPE OF PROJECT**

Carpooling system is one of the best ways that can help in minimizing the pollution and the traffic hiccups in the cities. It also gives him an environmentally friendly way to travel. It also has the potential to expand the circle of acquaintances. Since many people today use personal car to travel due to time consumed by public transport system and facilities offered within a private car. Pre-registration also adds to security since only identified people are allowed to get into the vehicle for the investment of trust. The people registered can be divided into certain days/shifts as when they should bring their private vehicle and thus inconvenience the registered passengers for their daily transportation needs. Therefore, the proposed carpooling system will also be helpful to the reduction of environment pollution in our society.We are privileged to be living in a world where new technologies are being invented every day and the previous one was not valid anymore. Mass population shifts on new technology as they are easy to use since the makers develop them for fast accessing of the application. It is common knowledge that with every second edition of a book some changes are made and mistakes made in the previous edition are corrected. Similar is the situation in case of the modern technology. With each upgrade, new features are installed to help in the flow of activities within the system. In future steps to improve the system will be Accurate positioning of vehicle and developing an algorithm in such a way that each driver has enough trips at work.

**CHAPTER 8**

**TESTING AND COSTING**

Testing is the process of executing the software in order to identify the defects or bugs in it. Software testing is important because it helps to identify and eliminate errors that can hinder efficient operation of software. Testing, through a sequence of systematically performing several test cases, seeks to establish the presence of any defects or faults in the software. Appropriate testing means that all the errors that may have been pointed out in the course of development are worked out, thus creating a better software product. The final goal is to develop efficient and fault-free software that provides operations as per the user specifications and expectations.**8.1 Principles of Testing:**

Thus, the study adopted the following hypotheses: Null hypothesis: There is no difference in postpartum anxiety and depression levels between the mothers in the sample and those of other comparable studies.

1. It is also important for tests to meet customer needs and expectations.2. It is recommended that testing should be done by a different team from the development team.3. Since conducting thorough tests is not feasible, emphasize on testing that would complement the risk assessment.4. These tests should however be planned prior to implementation.

5. Follow the Pareto principle: An astounding 80/20 rule states that 80% of the errors are caused by 20% of the program components.6. Begin with minor chunking and gradually increase the size of the chunks.

**8.2 Steps of Software Testing:**

1. Analysis Required.2. Test Planning.3. Test case Developing.4. Test environment Setting up.5. Test Executing.6. close the test cycle.

**8.3 Types of testing:**

8.3.1. Unit Testing:

It focuses on the granular level of software design, possibly working with a single component or a set of components which are integrated into a system. Code testing can be done by the programmers where they feed the program with sample inputs and check for the corresponding outputs.

8.3.2. Integration Testing:

This means that addition of a new module can cause changes in the program throughout its flow. This form of testing, known as regression testing, ensures all components remain functional regardless of the addition of new segments within the total program.

8.3.3. Regression Testing:

Introducing a new module might affect the rest of the program. Regression testing helps verify that all other elements remain workable after new ones have been added and integrated, thus preserving the stability and integrity of the program.

8.3.4. Validation and Verification Testing:

In software testing verification and validation or V & V are procedures that seek to ensure that a software system meets the required specifications and performs the intended tasks. Described also as software quality control, these procedures check if the developed software meets the requirements and functions properly.

8.3.5. White Box Testing:

White Box Testing, also known as Clear Box Testing or Structural Testing, is a method of software testing where the tester is fully aware of internal structure, design, or implementation of the item under test.

8.3.6. Black Box Testing:

Black Box Testing also referred to as Behavioral Testing is a type of testing done without any prior knowledge of the internal structure of the software being tested. They are designed according to requirements and functionality and are primarily of a functional nature but can also include some non-functional characteristics.8.3.7. Regression Testing

Regression testing is among the most important processes in software development, as it checks whether changes to a program affect its previously tested functionality. Usually, this process is performed by independent code testing specialists in large organizations. These testers come up with test scripts and cases to test new code units into a test bucket. This test bucket is run before a new software version is released to ensure that existing capabilities have not been impacted. Because alterations or enhancements to code can introduce errors inadvertently, regression testing remains relevant across the SDLC. Mike Kelly, an expert in the field, also elaborates on the importance and need for regression testing.

#### 8.3.8. Accessibility Testing

Accessibility Testing is the process of ensuring that the software or the application can be operated and navigated by persons with certain disabilities like deaf, color blind, mentally impaired, blind, and the elderly. This process includes elements such as font size for the visually impaired and color/contrast for color blinded people.

#### 8.3.9 Ad-hoc Testing

Ad-hoc testing is performed without reference to test cases, plans or documentation. Its goal is to identify issues and interfere with the application execution by running various flows or random operations.

**CHAPTER 9**

**RESULT AND APPLICATION**

* 1. **Result**

The outcome includes demonstrating the sequence of operations and their presentation with different inputs. Firstly, the before login to the actual login page the user must register on the database filling the registration form as shown in below figure.

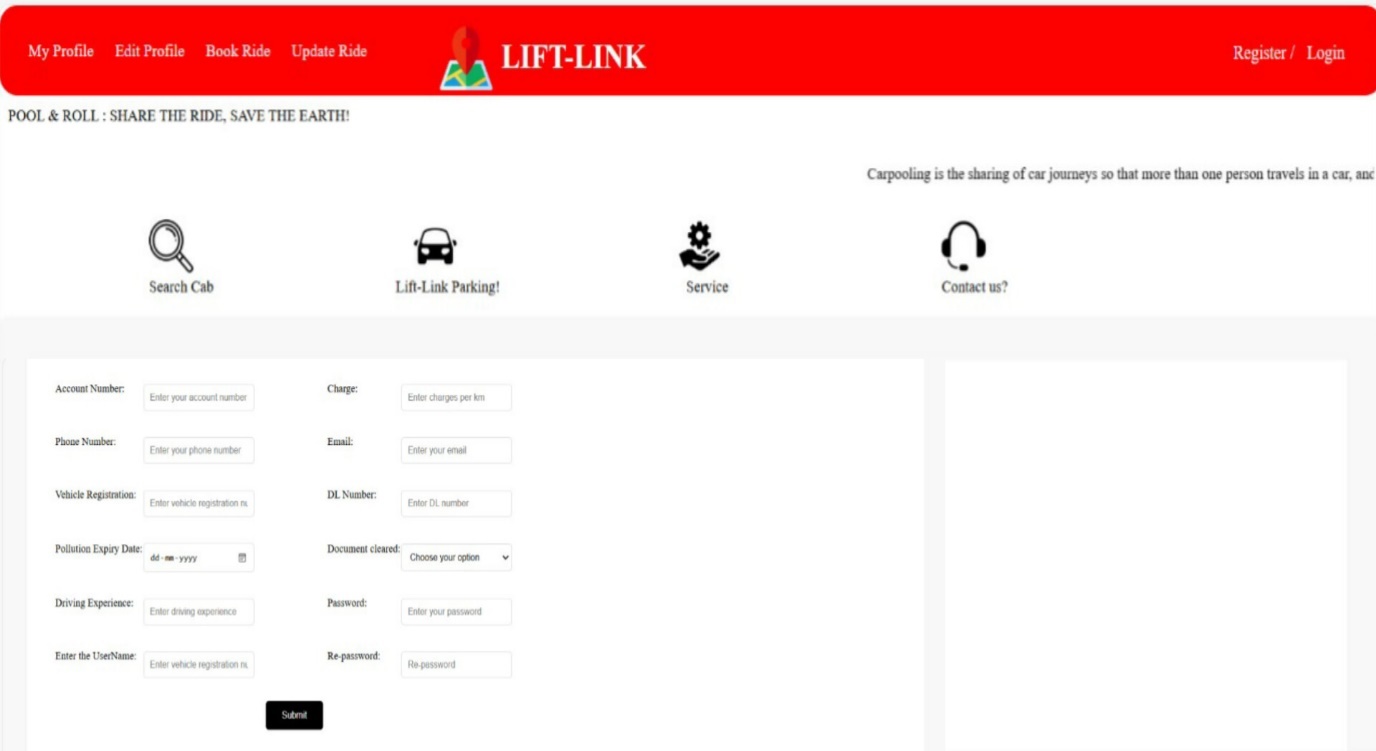


Figure 9.1 Registration form

After registering the database, the user must login to the page for accessing the further dashboard of the user as shown in below figure before login in there are two login setup the first login is for the client where client can book the ride can see the ride can update the profile and many more features but before accessing the user will be redirected to login page the figure is shown below.

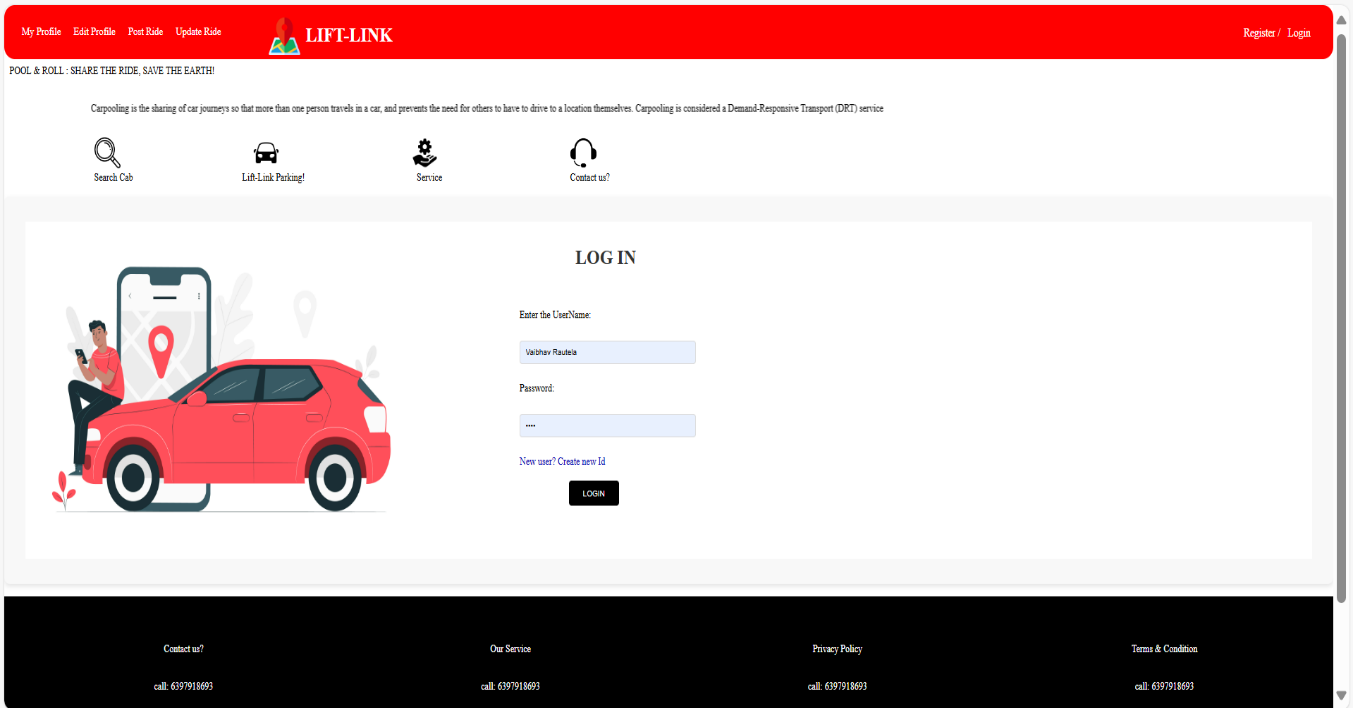


Figure 9.2 Log-in form

This is the rider login page where Rider can login if rider is not logged in it will be redirected to the registration page. You cannot move further before logging in.

As same the client must also log in to the profile before accessing the features of the model.

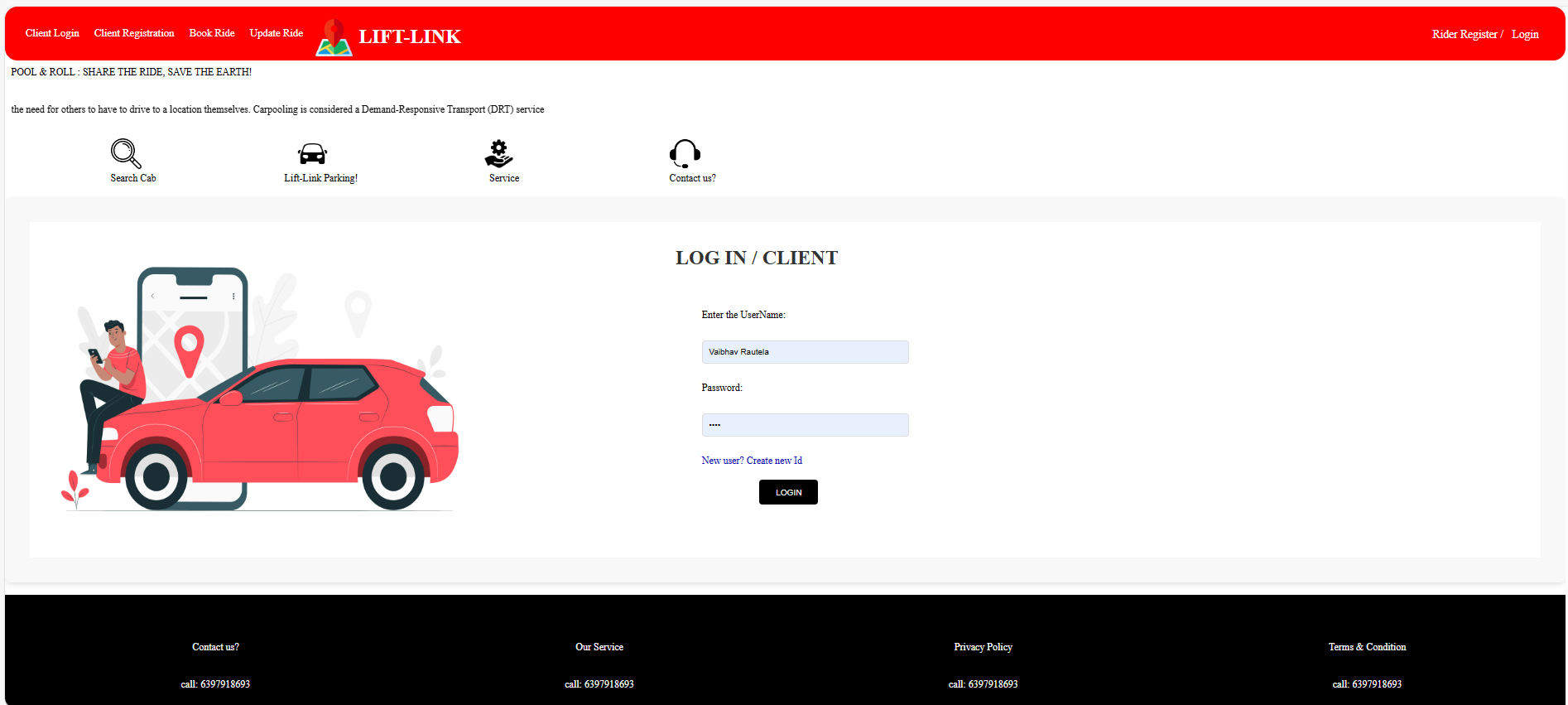


Figure 9.3 Client Log-in form

This page is for the client once the user is logged in to the profile the user has the features to access like edit profile, update ride, delete ride, show cancel ride, contact with [rautelavaibhav01@gamil.com](mailto:rautelavaibhav01@gamil.com). Once the user is login the software will be jumped to the dashboard the picture below shows the dashboard of the client.

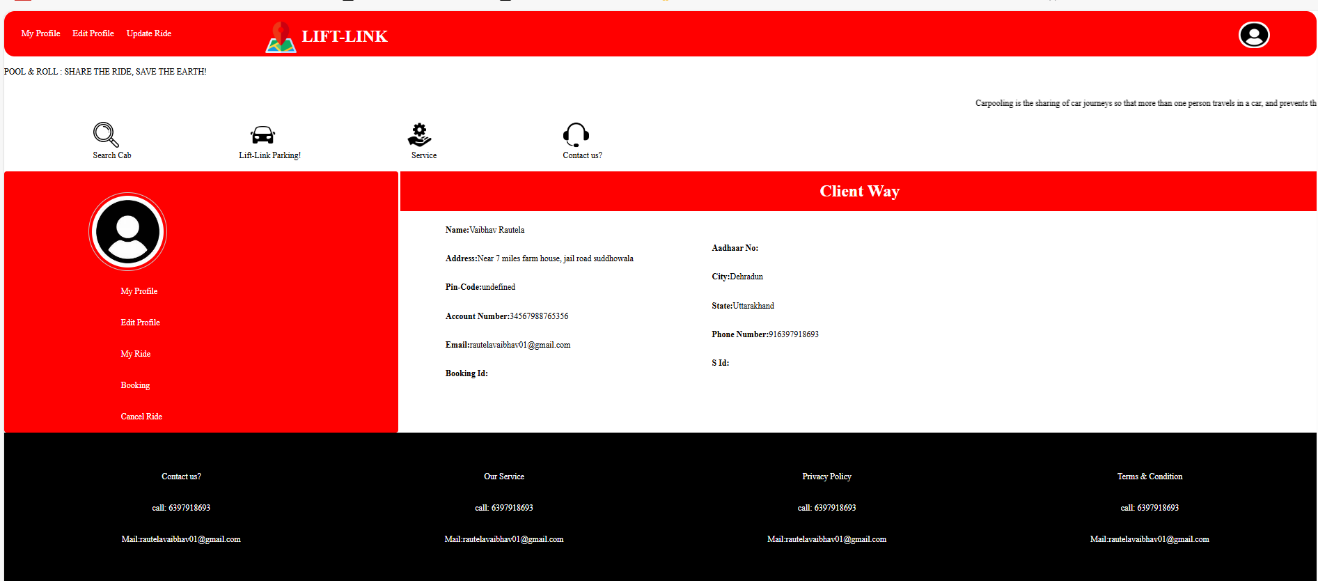


Figure 9.4 Dashboard

From the dashboard the you can go to booking where all the booking will be shown.

The picture below shows the booking where use can confirm their ride.

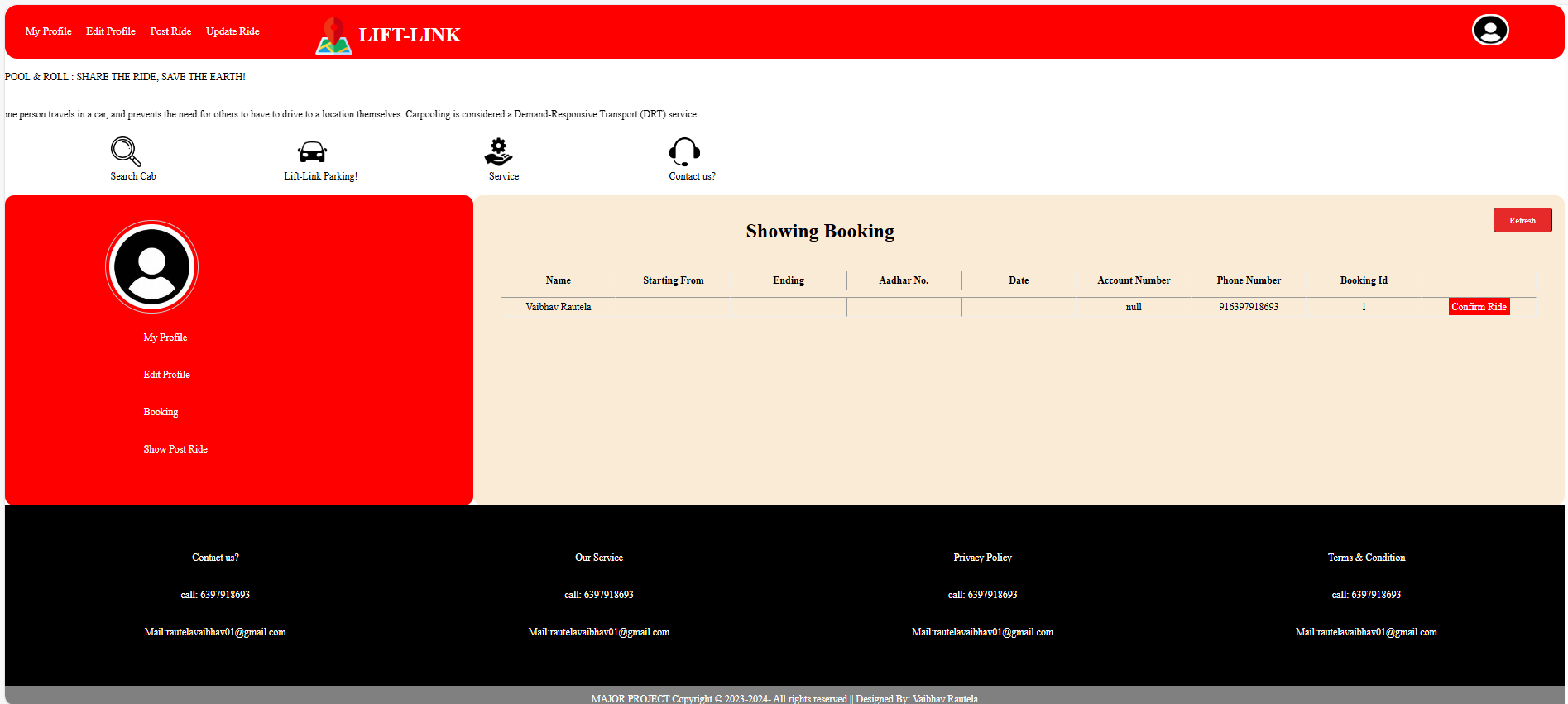


Figure 9.5 Showing Bookings.

The user can search the ride with the search option in the page the search page will be giving information. To which the rides are posted then the user can confirm the ride.

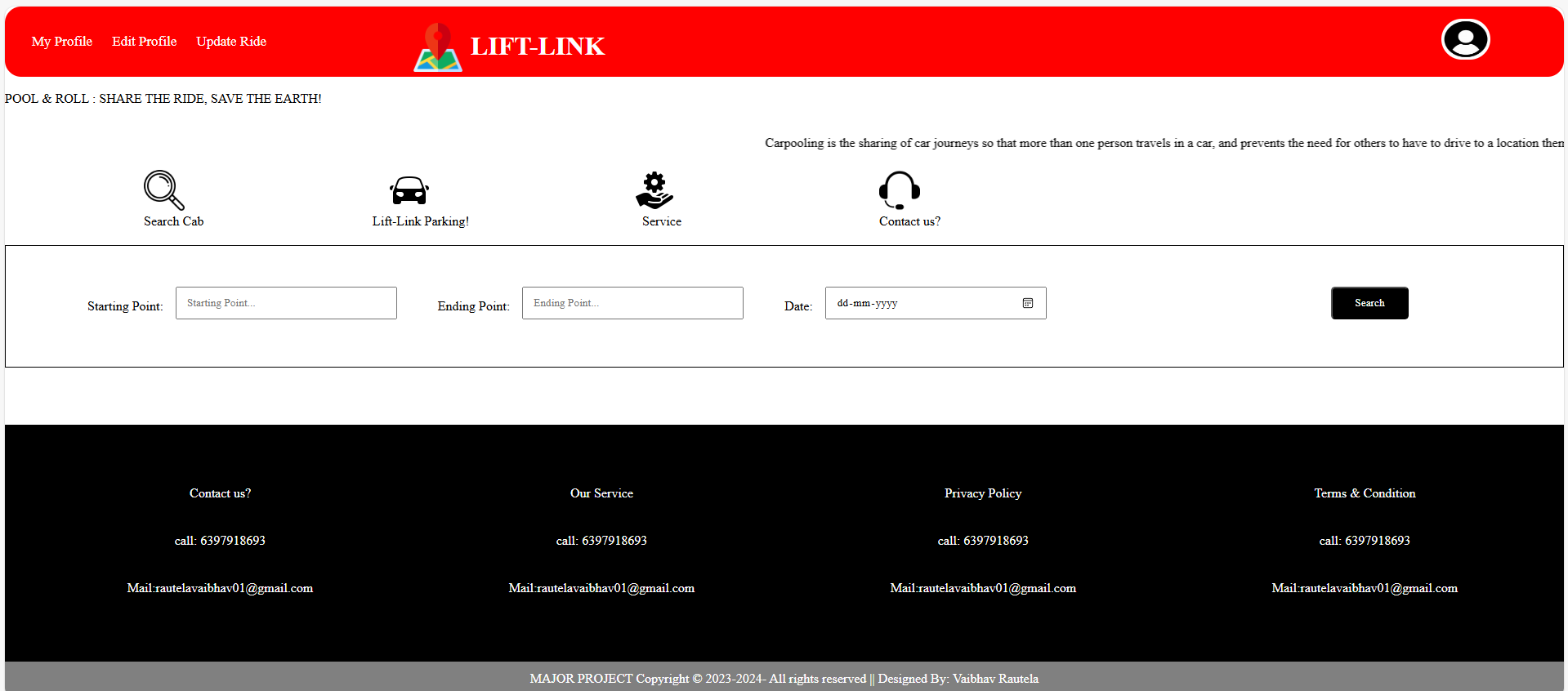


Figure 9.6 Search Ride

The user can see all the rides which are posted as shown in the pages. All the rides will be shown in the table.

After client posting the ride, the rider can cancel the rides which is booked earlier in the table it contains the cancel ride option as show in the figure below.

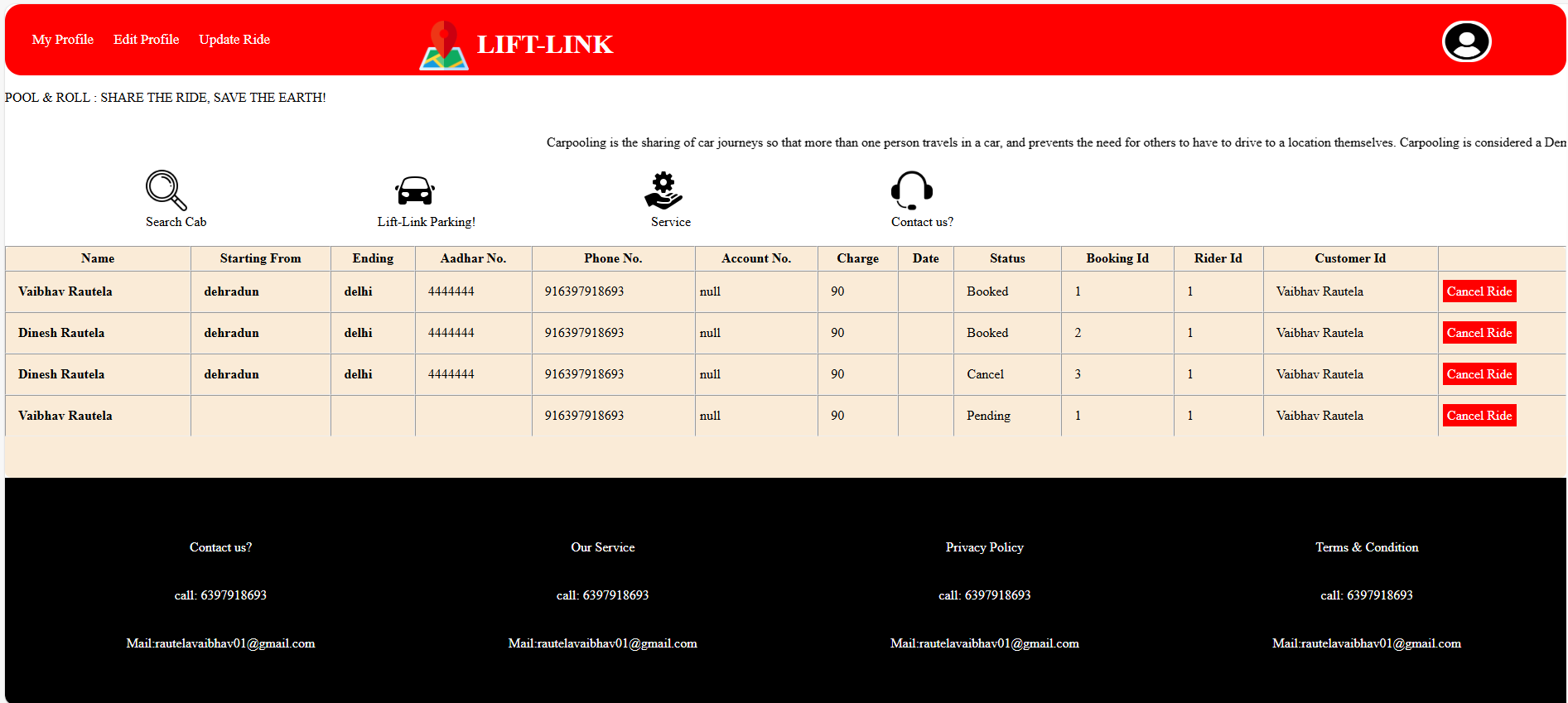


Figure 9.7 Rider confirmation

After client post the rider can cancel the ride which will be updated in my database at last the receipt will be shown when the rider posts it before the final call the user will see the receipt as shown in the figure below.

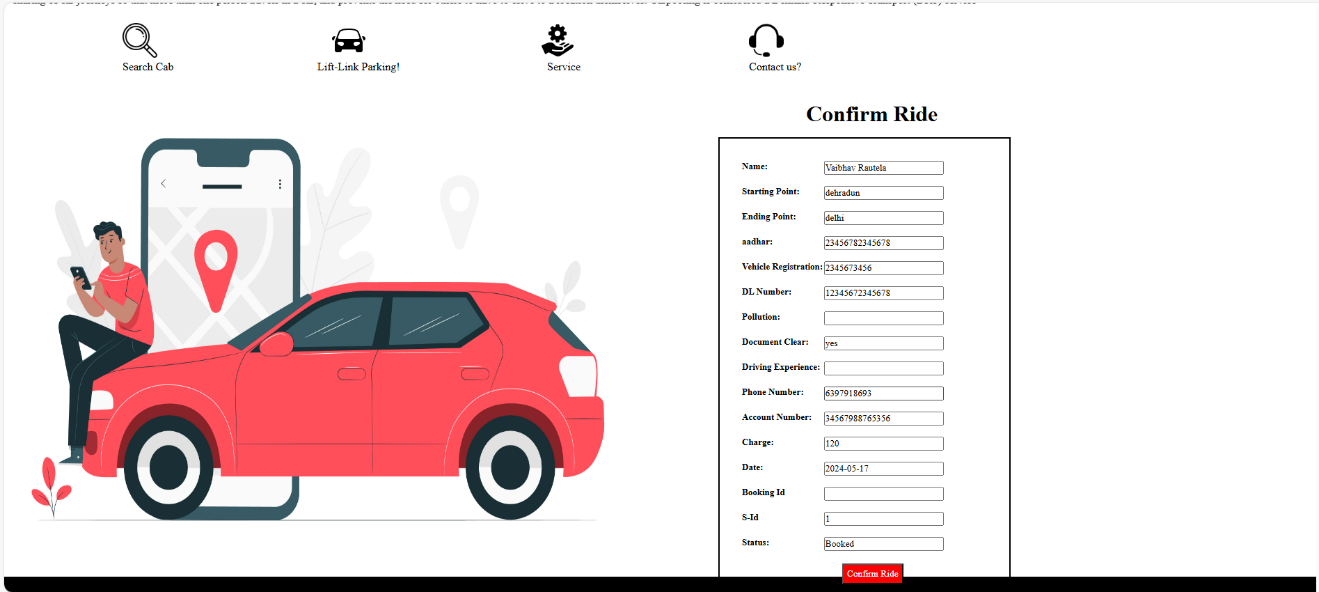


Figure 9.8 Confirm Ride form

This is the final page when the ride will be actually posted on the database. It is for the confirmation of ride.

User can re-update the password from clicking on profile on top right there you can choose change password/ update the password.

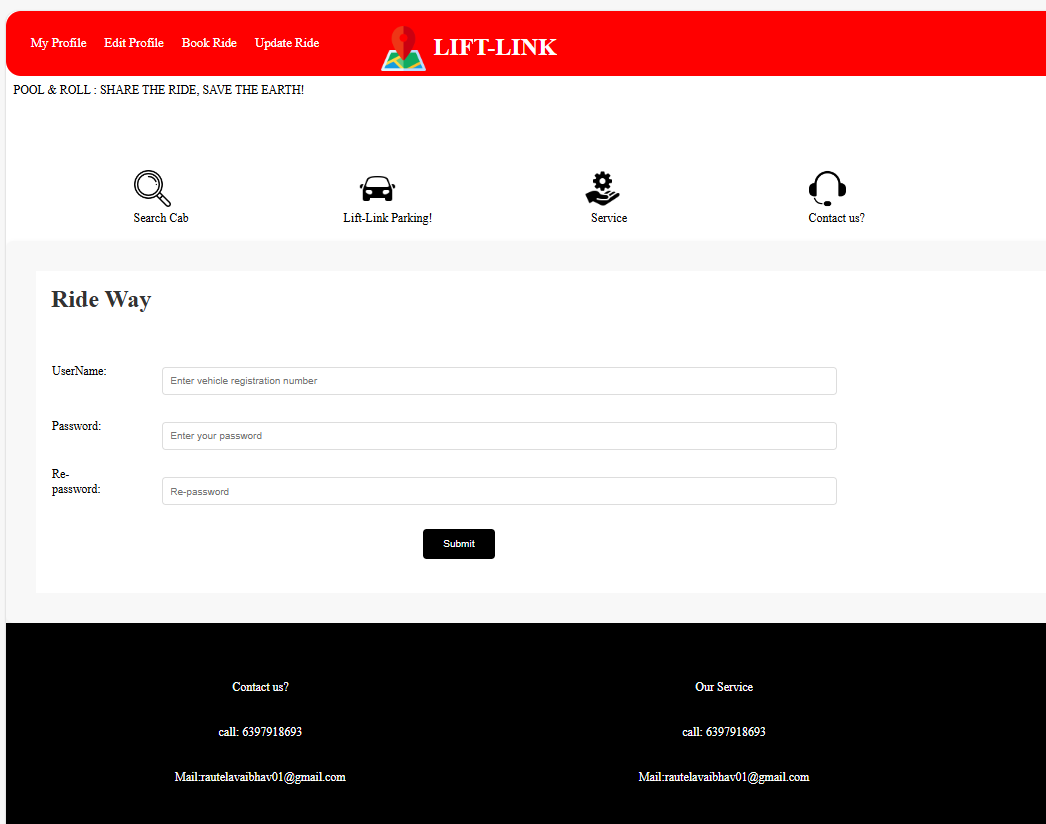


Figure 9.9 Reset Password

The ride that has been posted by user can be seen by my post ride from the dashboard and we can clearly see the ride that has been posted by the rider.

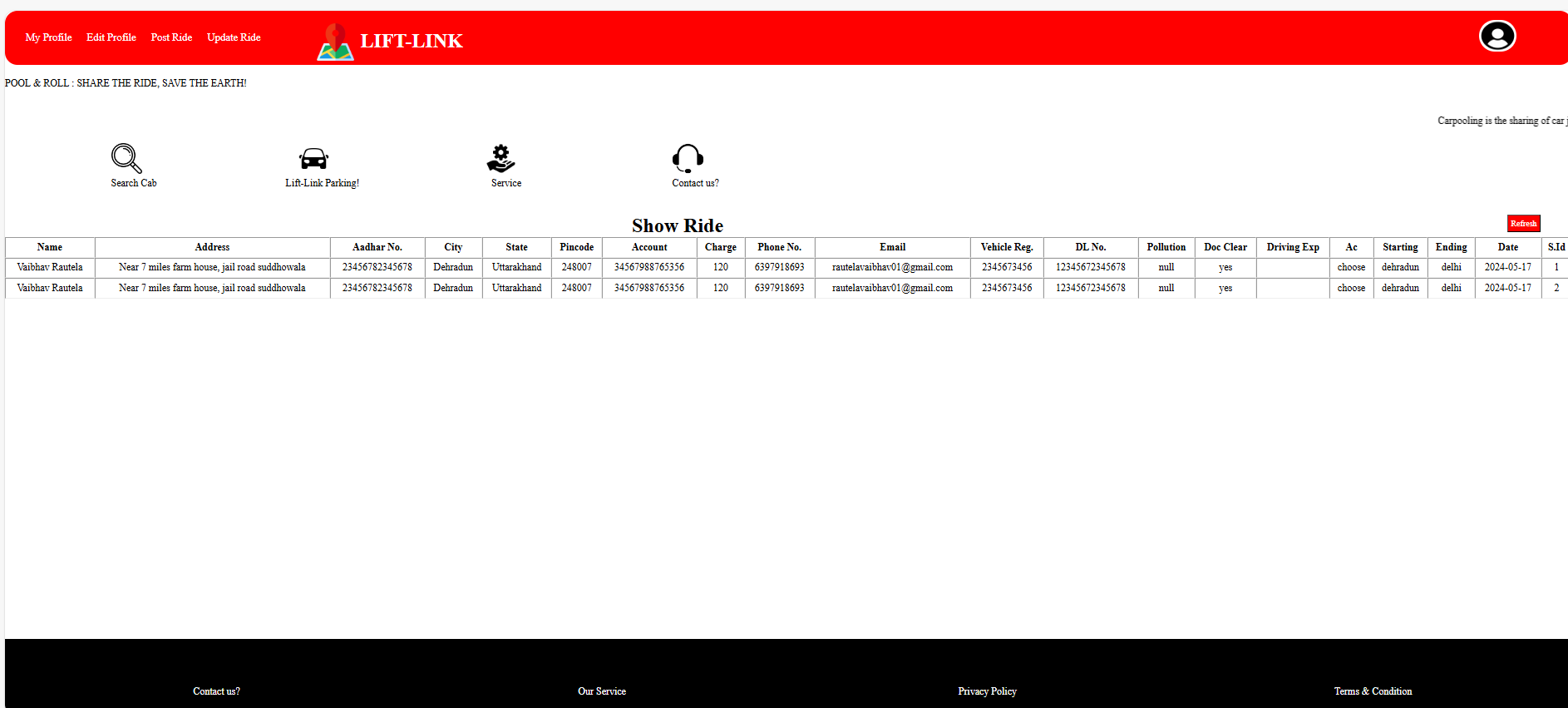


Figure 9.10 Showing Confirm form

Our aim cannot be successful until our user is not stress-free. Once the rider is going for a ride and as we have seen that the cities in the rural areas are tightly packed and no space is left for parking our cars. With the idea innovation of lift link user, the rider can park the car in our parking.

The user can choose the option from the selection bar once the user clicks on search all the ride that is booked will be shown the user can click on the get in ride. With this the rider will be send to the next geolocation page.

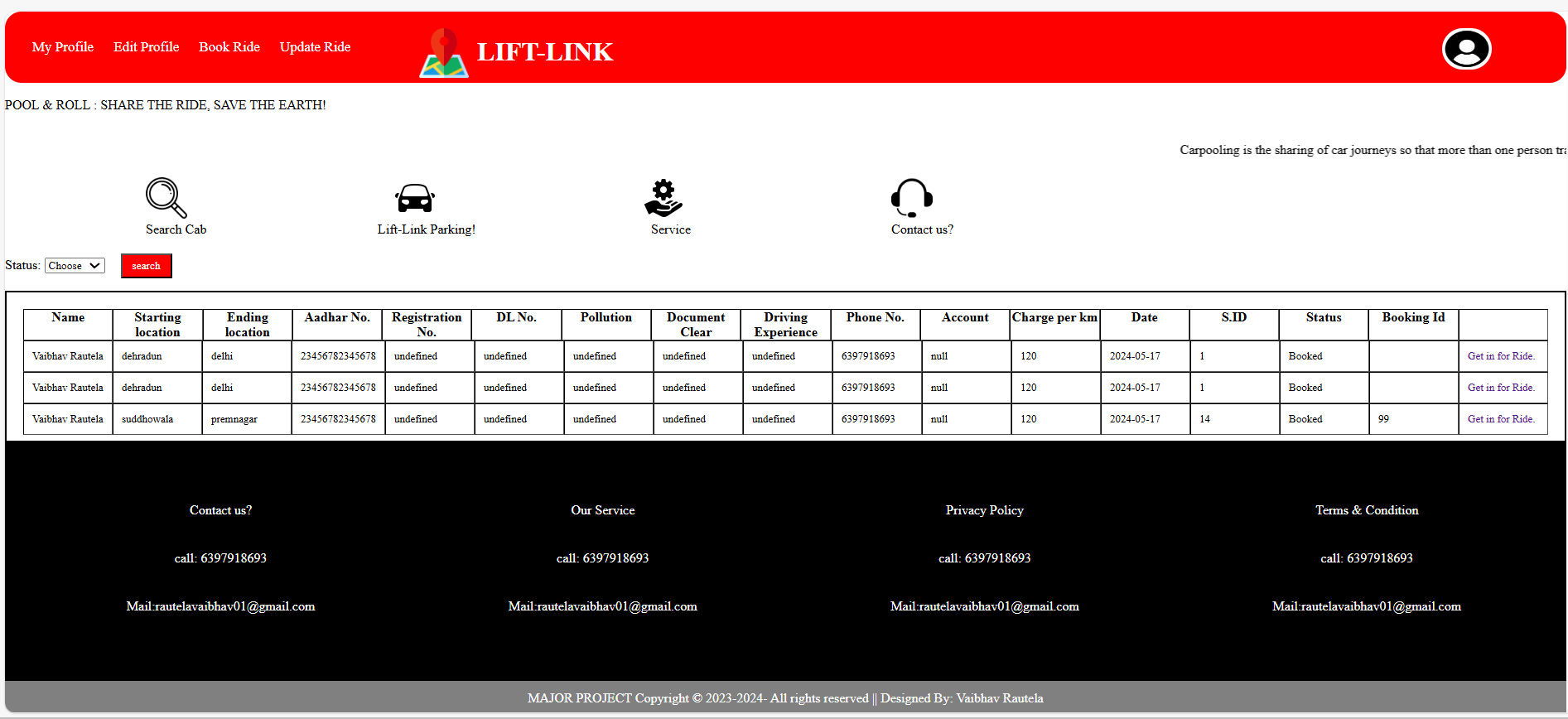


Figure 9.11 Booked Rides page

On location the person A can, be able to catch the person B from the site and can follow up to each other as shown below in the figure we can see the proper location of the person.

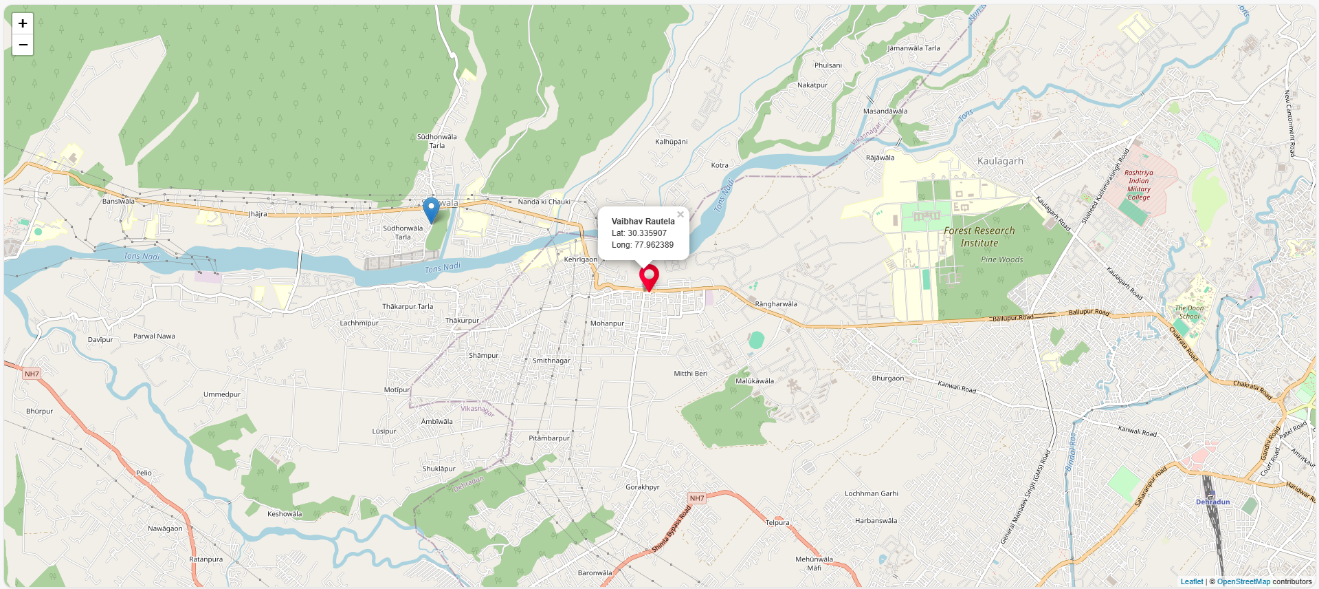


Figure 9.12 Geolocation

WORKING MODEL:

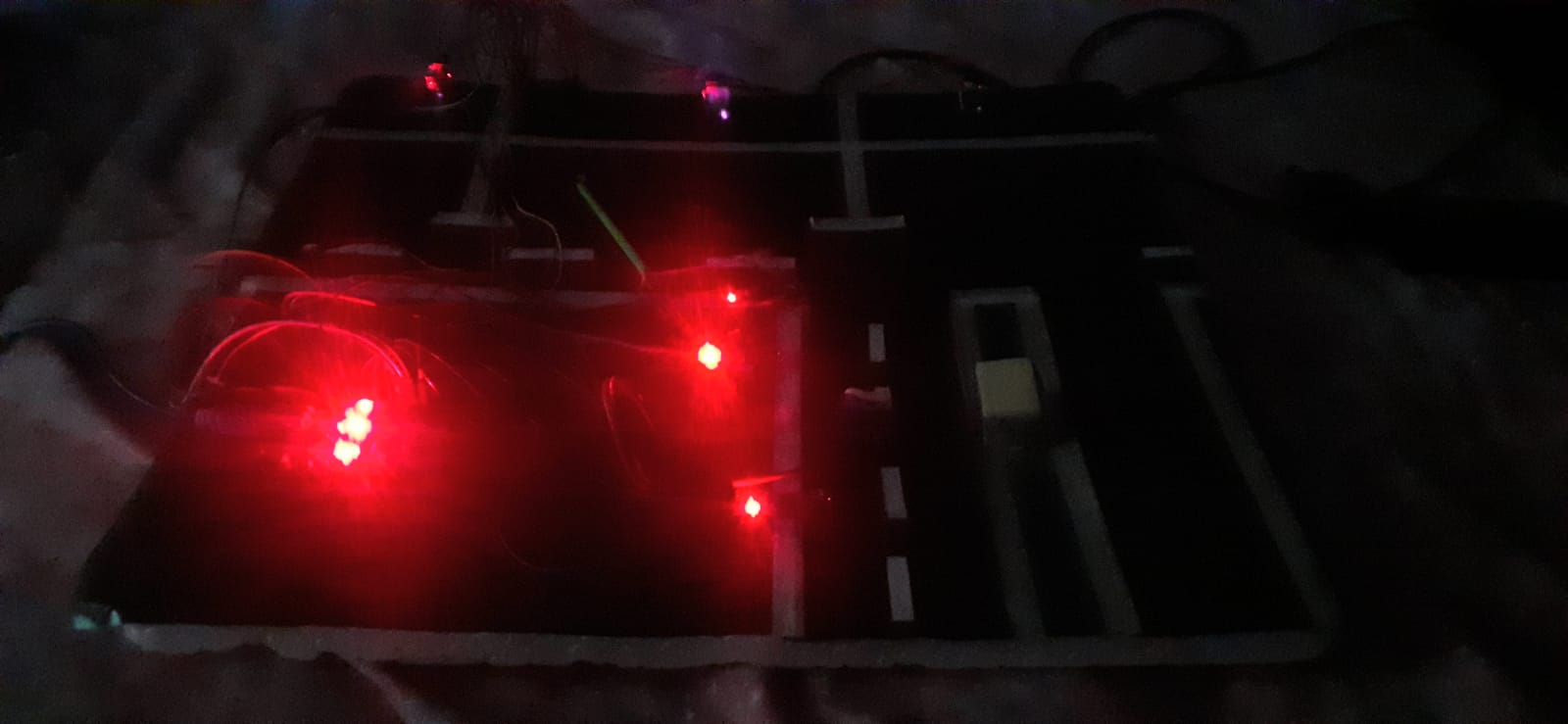


Figure 9.13 IoT sensor’s

This model shows the optimized solution for parking with optimized source of energy the below picture shows the model of IOT.

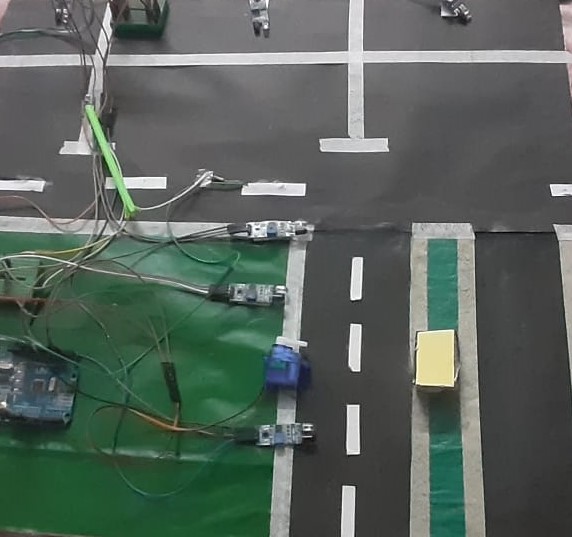


Figure 9.14 Parking Models

The IR sensor is providing the reading in the form of (0,1) once ir reads it sends to the arduino Uno board the board sends the data to the servo motor making the servo motor write on 90 degrees. The Arduino uno board displays the output in the serial monitor. Showing whether the parking is full or empty.

Once the IR3 is reading the data it is sending the result in form of (0,1) to the arduino and arduino is making the LED glow. LED is attached with little register which is supplying the required 2Amp current on the LED.

**9.2. Costing of project:**

COCOMO stands for Constructive Cost Model and is utilized for software cost estimation in order to determine the amount of effort, the time to complete the development of the software, and the human resources required. It relies on the size of the project being discussed with size being measured in terms of thousands of source lines of code, (KLOC). COCOMO divides every project into three categories in order to estimate the amount of time and resources to be used.

Project three Categories in COCOMO:

* Organic Projects
* Team Size: Solo
* Experience: Average
* Semi-Detached Projects
* Team Size: Medium
* Experience: Mixed
* Requirements: A combination of flexible and strict
* Embedded Projects
* Team Size: Varies
* Experience: Varies
* Requirements: Strict, with a combination of hardware, software, and operational aspects

Equations:

Development Time(D)= (c^b) \* (E efforts applied) \* db [months]

People required (P) = Effort Applied / Development Time [count]

KLOC (thousands of source lines of code) represents the estimated number of lines of code to be delivered for a project. The coefficients a, b, c for different project types are as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Software project** | ***Ab*** | ***Bb*** | ***cob*** | ***Db.*** |
| Organic | 2.4 | 1.05 | 2.5 | 0.38 |
| Semi-detached | 3.0 | 1.12 | 2.5 | 0.35 |
| Embedded | 3.6 | 1.20 | 2.5 | 0.32 |

Table 9.1 tables showing values for Ab, Bb, Cob, db

Basic COCOMO provides quick software cost estimates but does not account for hardware constraints, personnel quality and experience, or modern tools and techniques.

For an embedded software project, the calculations are:

Effort Applied (E) = (3.6) \* (7) ^ (1.20) = 37.18 person-months

Here, ab is 3.6 and bb is 1.20 as mentioned in above table and KLOC i.e. lines of code are 7 which is expressed in thousand.

Development Time

(D)= 2.5 × (37.18) ^ (0.32) = 7.95 months

Here, cb is 2.5 and db is 0.32.

People required:

(P) = 37.18/7.95= 4.674

According to this calculation our project requires 4 Numbers of people.

### 9.2.1 Working Hours:

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **Work to perform** | **Date** | **Duration (Hrs)** |
| 1 | Project Selection | Daily | 15 |
| 2 | Flow chart | Daily | 15 |
| 3 | Development in Software | Daily | 20 |
| 4 | Testing of Module | Daily | 30 |
| 5 | Report Project | Daily | 20 |

### Table 9.2 table showing Working Hours

### 9.2.2 Costing:

Table 9.3: Costing

|  |  |
| --- | --- |
| Hours (in above table) | 100 |
| man power Cost | Rs. 25/- per Hour |
| Computer Rent | Rs. 20/- per Hour |
| Other Expenses and stationery | Rs. 1500/- |

### 9.2.3 Total Cost:

|  |  |
| --- | --- |
| man power (100\*25) | Rs. 2500/- |
| Total computer rent for months (100\*20) | Rs. 7000/- |
| Stationary & other expenses | Rs. 1500/- |
| Arduino and other sensor’s | Rs. 3500/- |
| Total | Rs. 14,500/- |

Table 9.4: Total Cost

**CHAPTER 10**

**CONCLUSIONS AND FUTURE SCOPE**

The purpose of the carpooling system is to minimize the use of fuel, which is a non-renewable source and also help in avoiding traffic congestion through sharing of cars. As an eco-social app, it helps users reduce the amount of time spent on the road. This paper presents a synthesis of different authors’ opinions on carpooling systems where it examines different methods such as carpooling through SMS alerts and GPS tracking. Additionally, there exist some areas that need enhancement on the project such as integration of real-time tracking location of the car and online payment options for the users among others.

Carpooling systems are highly advantageous because they help to minimize pollution and traffic jams resulting from an excessive number of cars in circulation. This method of shared transportation is, therefore, friendly to the environment as people are encouraged to travel from one destination to another using public means. This is because by choosing carpooling means, people get a chance to converse and communicate with people of different states and cultures.

Since public transport is scarce and their schedules are unpredictable most working people use their personal cars as their main source of transport. Carpooling provides a solution to this problem by offering a dependable option that may be even more time-sensitive and convenient. Passengers planning to use the carpooling system are first asked to input their details. This is important for the safety of people during their travels, as well as for building passengers’ confidence. Having identified fellow travellers and the driver makes the experience safer and more comfortable.

Furthermore, the systems avails that involve car-pooling play an instrumental role in decreasing pollution around our environment. Through sharing of rides, it means fewer cars are on the road hence reduce pollution of the environment with toxic gases, smoke among others. Thus, the pollution of various resources is less, which has a favourable effect on the well-being of the community and environmental conditions. Furthermore, the data entered when registering a driver or vehicle also helps in increasing safety measures since such details can always be accessed by the customers whenever they are needed, thus improving customers’ satisfaction.Another significant gain that comes with minimizing the number of people that commute to work is the saving of time. HOV lanes, typically used for multi-occupancy vehicles such as buses and cars carrying people beyond a given number, assist in avoiding ‘n’ regular traffic and hence affords faster and efficient journeys. This also reduces or eliminates the reliance on several things like fuel and traffic hence making traveling more comfortable and less worrisome.There is the economic benefit of going by carpooling since it costs users money that they would otherwise be used for toll taxes, fueling their personal cars, and car maintenance, among other expenses. Carpooling is effective because it splits the overall cost of traveling through sharing a vehicle effectively, reducing the cost for all. Carpooling also assists in conserving energy because instead of many drivers driving individually, just a few will be driving in circulation. When the number of workers increases, it means that it is going to be easier for them if they can share the journey with other workers. Through sharing the wheel, people can simply switch to the passenger seat in a car, which is beneficial, as it reduces stress and allows for effective work during the trip.Thus, the carpooling systems are a real, reasonable, and eco-friendly driving model in contrast to careering as a single occupancy vehicle. They cut down polluting of water and air besides being helpful in cutting traffic jams, increase security of an individual through pre-registration, reduces time and expenses that are used while traveling to work daily, and time associated stress. Carpooling means people can drive together to work or on other business and back in one car, hence encouraging people to be environmentally conscious and feel more united while getting to work or back home in comfort.

**CHAPTER 11**

**REFERENCES**

[1] shilpa bilawane,pranali jambhulkar,” Information system using android platform”, International Journal Of Engineering And Computer Science 2017

[2] Mulla Kajal,Mahadik Awanti, Pandharpatte Sonali, Kalantre Rashmi, Bansode Swapnali, “Online Training and Placement System” ,International Research Journal of Engineering and Technology.

[3] CaiZhongxi, “Campus Employment Information Network Development based on Android Platform” International Conference on Intelligent Transportation, 2015.

[4] DikshaVarshney,BhumikaSharma,SomyaJain“Campus Recruitment Management: Platform based on dynamic electronic commerce”,2014.

[5] Divyesh P, et al. A Smart Real Time Ridesharing Android Application. International Journal on Recent and Innovation Trends in Computing and Communication 2016; 4:188-192.

[6] Arpita D. Real-Time Carpooling System for Android Platform. International Journal of Engineering and Innovative Technology (IJEIT). 2012:436-437.

[7] Nixon (2009). Learning, MySQL, JavaScript, CSS & HTML5: A Step-by-Step Guide to Creating Dynamic Websites, third Edition

[8] Duckett j (2014). JavaScript and jQuery: Interactive Front-End Web Development, Wiley

[9] Holzer’s (2008). Ajax: The Complete Reference, McGraw Hill Education, Second Edition

[10] Haddad, Yoram & Cohen, Yuval & Goldsmith, Ronen. (2013). A Dynamic Real Time CarSharing System.

[11] Osman, Mohd & Muhd Zain, Nur Zaid & Paidi, Zulfikar & Sendek, Khairul Anwar. (2017). Online Car Rental System using Web-Based and SMS Technology.

[12] Bresciani, Chiara & Colori, Alberto & Costa, Francesca & Leu, Alessandro & Superblock. (2018). Carpooling: facts and new trends. 1-4. 10.23919/EETA.2018.8493206

**CHAPTER 12**

**APPENDIX**

**Code for the Iot Proto-Type:**

#include <Servo.h>

#include <Wire.h>

Servo tap\_servo;

int sensor\_pin1=2;

int sensor\_pin2=4;

int ir\_car1 =5;

int ir\_car2 =6;

int ir\_car3 =7;

int tap\_servo\_pin =3;

int S1=0, S2=0, S3=0;

int slot = 3;

int val1;

int val2;

void setup()

{

pinMode(sensor\_pin1,INPUT);

pinMode(sensor\_pin2,INPUT);

tap\_servo.attach(tap\_servo\_pin);

pinMode(8,INPUT);

pinMode(9,INPUT);

pinMode(10,INPUT);

pinMode(11,OUTPUT);

pinMode(12,OUTPUT);

pinMode(13,OUTPUT);

}

void loop(){

val1 = digitalRead(sensor\_pin1);

val2 = digitalRead(sensor\_pin2);

Serial.begin(9600);

if(val1==0&&val2==0)

{

tap\_servo.write(0);

}

else if(val1==1&&val2==0)

{

tap\_servo.write(90);slot=slot+1;

}

else if(val1==0&&val2==1)

{

tap\_servo.write(90);slot=slot-1;

}

else if(val1==1&&val2==1)

{

tap\_servo.write(0);

}

{

if(digitalRead(8)==1&&digitalRead(9)==1&&digitalRead(10)==1)

{

digitalWrite(11,1);

digitalWrite(12,1);

digitalWrite(13,1);

}

else if(digitalRead(8)==1&&digitalRead(9)==1)

{

digitalWrite(11,1);

digitalWrite(12,1);

digitalWrite(13,0);

}

else if(digitalRead(9)==1&&digitalRead(10)==1)

{

digitalWrite(11,0);

digitalWrite(12,1);

digitalWrite(13,1);

}

else if(digitalRead(8)==1&&digitalRead(10)==1)

{

digitalWrite(11,1);

digitalWrite(12,0);

digitalWrite(13,1);

}

else if(digitalRead(8)==1)

{

digitalWrite(11,1);

digitalWrite(12,0);

digitalWrite(13,0);

}

else if(digitalRead(9)==1)

{

digitalWrite(11,0);

digitalWrite(12,1);

digitalWrite(13,0);

}

else if(digitalRead(10)==1)

{

digitalWrite(11,0);

digitalWrite(12,0);

digitalWrite(13,1);

}

else

{

digitalWrite(11,0);

digitalWrite(12,0);

digitalWrite(13,0);

}

}

Read\_Sensor();

if(S1==1)

{

Serial.println("\nS1:Empty ");

}

else

{

Serial.println("\nS1:Fill");

if(S2==1)

{

Serial.print("\nS2:Fill");

}

else

{

Serial.print("\nS2:Empty");

}

if(S3==1){Serial.print("\nS3:Empty ");}

else

{

Serial.print("\nS3: Fill");

}

if(S3==0){

Serial.println(" Sorry Parking Full ");

}

delay(1500);

}

void Read\_Sensor()

{

S1=0, S2=0, S3=0;

if(digitalRead(ir\_car1) == 0){S1=1;}

if(digitalRead(ir\_car2) == 0){S2=1;}

if(digitalRead(ir\_car3) == 0){S3=1;}

}

**Software Code:**

**App.js:**

const express = require('express');

const http = require('http');

const mysql = require('mysql2');

const url = require("url");

const bcrypt = require('bcrypt');

const path = require("path");

const bodyParser = require('body-parser');

const users = require('./data').userDB;

const app = express();

const server = http.createServer(app);

const fs = require('fs');

app.use(bodyParser.urlencoded({ extended: false }));

app.use(express.static(path.join(\_\_dirname, './public')));

app.get('/', (req, res) => {

res.sendFile(path.join(\_\_dirname, './public/index.html'));

});

app.post('/registration', async (req, res) => {

var mydb = mysql.createConnection({ host: "localhost", user: "root", password: "root123", database: "liftlink" });

try {

var a1 = req.body.name;

var a2 = req.body.address;

var a3 = req.body.aadhar;

var a4 = req.body.city;

var a5 = req.body.state;

var a6 = req.body.pincodes;

var a7 = req.body.account;

var a8 = req.body.charge;

var a9 = req.body.phoneno;

var a10 = req.body.email;

var a11 = req.body.vehicle\_reg;

var a12 = req.body.dl\_no;

var a13 = req.body.pollution;

var a14 = req.body.document\_clear;

var a15 = req.body.driving\_expe;

var a16 = req.body.username;

var a17 = req.body.password;

var a18 = req.body.repassword;

mydb.connect(function (err){

if (err) throw err;

console.log("connected");

mydb.query("INSERT INTO registration

(name, address, aadhar, city, state, pincode, account, charge, phoneno, email, vehicle\_reg, dl\_no, pollution, document\_clear, driving\_expe, username, password, repassword,sid,BookingId)

values('" + a1 + "','" + a2 + "','" + a3 + "','" + a4 + "','" + a5 + "','"

+ a6 + "','" + a7 + "','" + a8 + "','" + a9 + "','" + a10 + "','" + a11 + "','" + a12 +

"','" + a13 + "','" + a14 + "','" + a15 + "','" + a16 + "','" + a17 + "','" + a18 + "')"

,function(err, result) {

if (err) throw err;

console.log("Record inserted.");

res.sendFile(path.join(\_\_dirname, 'public', 'login2.html'));

});

});

} catch (error) {

res.send("Internal server error");

}

});

app.post('/login', (req, res) => {

var mydb = mysql.createConnection({ host: "localhost", user: "root", password: "root123", database: "liftlink" });

const { username, password } = req.body;

const sql = "SELECT \* FROM registration WHERE username = ? AND password = ?";

const values = [username, password];

mydb.query(sql, values, (err, result) => {

if (err) {

console.error('Error executing SQL query:', err);

return res.status(500).send('Internal Server Error');

}

if (result.length > 0) {

res.redirect('/sucess');

} else {

res.redirect('/unsucess');

}

});

});

app.get('/login', (req, res) => {

res.sendFile(path.join(\_\_dirname, 'public', 'owner\_login.html'));

});

app.get('/unsucess', (req, res) => {

res.sendFile(path.join(\_\_dirname, 'public', 'owner\_login.html'));

});

app.get('/sucess', async (req, res) => {

var mydb = mysql.createConnection({ host: "localhost", user: "root", password: "root123", database: "liftlink" });

mydb.connect(function (err) {

if (err) throw err;

console.log("connected");

mydb.query(" select \* from registration", function (err, result) {

if (err) throw err;

const fs = require('fs');

const databasedata = JSON.stringify(result);

const filePath = path.join(\_\_dirname, './public/owner\_after\_login.json');

fs.writeFile(filePath, databasedata, 'utf8', (err) => {

if (err) {

console.error('Error writing file:', err);

return;

}

console.log('JSON data has been saved to', filePath);

});

res.sendFile(path.join(\_\_dirname, 'public', 'owner\_after\_login.html'));

});

});

});

app.post('/password', async (req, res) => {

var mydb = mysql.createConnection({ host: "localhost", user: "root", password: "root123", database: "liftlink" });

try {

const { username, password } = req.body;

const sql = "UPDATE registration SET password = ? WHERE username = ?";

await mydb.query(sql, [password, username]);

console.log("Password updated.");

res.sendFile(path.join(\_\_dirname, 'public', 'passwordreset.html'));

} catch (error) {

console.error('Error:', error);

res.status(500).send('Internal Server Error');

}

});

app.post('/profileupdate', async (req, res) => {

var mydb = mysql.createConnection({ host: "localhost", user: "root", password: "root123", database: "liftlink" });

try {

const { name, aadhar, address, city, pincode, state, account, charge, phoneno, email, vehicle\_reg, pollution, document\_clear, dl\_no, driving\_expe } = req.body;

const sql = "UPDATE registration SET

aadhar=?, address=?, city=?, pincode=?, state=?, account=?, charge=?, phoneno=?, email=?, vehicle\_reg=?,

pollution=?, document\_clear=?, dl\_no=?, driving\_expe=? WHERE name=?";

const values = [aadhar, address, city, pincode, state, account, charge, phoneno, email, vehicle\_reg, pollution, document\_clear, dl\_no, driving\_expe, name];

mydb.query(sql, values, (err, result) => {

if (err) {

console.error("Error updating record: ", err);

res.status(500).send("Error updating record");

return;

}

console.log("Record updated.");

res.sendFile(path.join(\_\_dirname, 'public', 'edit\_profile.html'));

});

} catch (error) {

console.error("Error:", error);

res.status(500).send("Update server error");

}

});

app.post('/postride', async (req, res) => {

var mydb = mysql.createConnection({ host: "localhost", user: "root", password: "root123", database: "liftlink" });

try {

const { name, address, aadhar, city, state, pincode, account, charge\_per\_unit, phoneno, email, vehicle\_reg, dl\_no, pollution, document\_clear, driving\_experience, ac, starting\_location, ending\_location, time\_taken,sid,star } = req.body;

const sql = "INSERT INTO postride (name, address, aadhar, city, state, pincode, account, charge\_per\_unit, phoneno,

email, vehicle\_reg, dl\_no, pollution, document\_clear, driving\_experience,ac, starting\_location,ending\_location,time\_taken,sid,star) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,?,?,?)";

const values = [name, address, aadhar, city, state, pincode, account, charge\_per\_unit, phoneno, email, vehicle\_reg, dl\_no, pollution, document\_clear, driving\_experience, ac, starting\_location, ending\_location, time\_taken,sid,star];

mydb.query(sql, values, (err, result) => {

if (err) {

console.error("Error inserting record: ", err);

res.status(500).send("Error inserting record");

return;

}

console.log("Record inserted.");

res.sendFile(path.join(\_\_dirname, 'public', 'post\_ride.html'));

});

} catch (error) {

console.error("Error:", error);

res.status(500).send("Error processing request");

}

});

app.post('/bookride', async (req, res) => {

var mydb = mysql.createConnection({ host: "localhost", user: "root", password: "root123", database: "liftlink" });

try {

const { name, aadhar, country, city, pincode, state, phoneno, email, ac, starting\_location, ending\_location, date\_taken, time\_taken, pin, booking\_number, status, rname, raadhar, rcountry, rcity, rpincode, rstate,

raccount, rcharge\_per\_unit, rphoneno, remail, rvehicle\_reg, rpollution, rdocument\_clear, rdl\_no, rdriving\_experience, rac, rstarting\_location, rending\_location, rtime\_taken} = req.body;

const sql = "INSERT INTO bookride (name, aadhar, country, city, pincode, state, phoneno, email, ac, starting\_location, ending\_location, date\_taken, time\_taken, pin, booking\_number, status, rname, raadhar, rcountry,

rcity, rpincode, rstate, raccount, rcharge\_per\_unit, rphoneno, remail, rvehicle\_reg, rpollution, rdocument\_clear, rdl\_no, rdriving\_experience, rac, rstarting\_location, rending\_location, rtime\_take) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,?)";

const values = [name, aadhar, country, city, pincode, state, phoneno, email, ac, starting\_location, ending\_location, date\_taken, time\_taken, pin, booking\_number, status, rname, raadhar, rcountry, rcity,

rpincode, rstate, raccount, rcharge\_per\_unit, rphoneno, remail, rvehicle\_reg, rpollution, rdocument\_clear, rdl\_no, rdriving\_experience, rac, rstarting\_location, rending\_location, rtime\_taken];

mydb.query(sql, values, (err, result) => {

if (err) {

console.error("Error inserting record: ", err);

res.status(500).send("Error inserting record");

return;

}

console.log("Record inserted.");

res.sendFile(path.join(\_\_dirname, 'public', 'book\_ride.html'));

});

} catch (error) {

console.error("Error:", error);

res.status(500).send("Error processing request");

}

});

app.post('/confirmride', async (req, res) => {

var mydb = mysql.createConnection({ host: "localhost", user: "root", password: "root123", database: "liftlink" });

try {

const { status, name } = req.body;

const sql = "UPDATE bookride SET status=? WHERE name=?";

const values = [status, name];

mydb.query(sql, values, (err, result) => {

if (err) {

console.error("Error updating record: ", err);

res.status(500).send("Error updating record");

return;

}

console.log("Record updated.");

res.sendFile(path.join(\_\_dirname, 'public', 'confirmride.html'));

});

} catch (error) {

console.error("Error:", error);

res.status(500).send("Update server error");

}

});

app.post('/rideupdate', async (req, res) => {

var mydb = mysql.createConnection({ host: "localhost", user: "root", password: "root123", database: "liftlink" });

try {

const {name, startingpoint, endingpoint, aadhar, phoneno, accountno, charge, date , BookingId, sid ,CustomerId } = req.body;

const sql = "UPDATE booking SET name=?, startingpoint=?, endingpoint=?, aadhar=?, phoneno=?, accountno=?, charge=?, date=? , sid=?,CustomerId=? WHERE BookingId=?";

const values = [name, startingpoint, endingpoint, aadhar, phoneno, accountno, charge, date , BookingId, sid ,CustomerId];

mydb.query(sql, values, (err, result) => {

if (err) {

console.error("Error updating record: ", err);

res.status(500).send("Error updating record");

return;

}

console.log("Record updated.");

res.sendFile(path.join(\_\_dirname, 'public', 'update\_ride.html'));

});

} catch (error) {

console.error("Error:", error);

res.status(500).send("Update server error");

}

});

app.post('/showride', async (req, res) => {

var mydb = mysql.createConnection({ host: "localhost", user: "root", password: "root123", database: "liftlink" });

mydb.connect(function (err) {

if (err) throw err;

console.log("connected");

mydb.query(" select \* from postride", function (err, result) {

if (err) throw err;

const fs = require('fs');

const databasedata = JSON.stringify(result);

const filePath = path.join(\_\_dirname, './public/showride.json');

fs.writeFile(filePath, databasedata, 'utf8', (err) => {

if (err) {

console.error('Error writing file:', err);

return;

}

console.log('JSON data has been saved to', filePath);

});

res.sendFile(path.join(\_\_dirname, 'public', 'show\_ride.html'));

});

});

});

app.post('/searchfun', async (req, res) => {

try {

var a1 = req.body.t1;

var a2 = req.body.t2;

var con = mysql.createConnection({ host: "localhost", user: "root", password: "root123", database: "liftlink" });

con.connect(function (err) {

if (err)

res.sendFile(path.join(\_\_dirname, 'public', 'aaa.html'));

console.log("connected");

con.query("select \* from postride where starting\_location ='" + a1 + "' and ending\_location ='" + a2 + "'", function (err, result) {

const fs = require('fs');

const jsonString = JSON.stringify(result);

const filePath = 'public/search1.json';

fs.writeFile(filePath, jsonString, 'utf8', (err) => {

if (err) {

console.error('Error writing file:', err);

return;

}

console.log('JSON data has been saved to', filePath);

});

if (result.length > 0) {

res.redirect('/searchresult');

} else {

res.redirect('/searchfun');

}

});

});

} catch

{

res.send("Internal server error");

}

});

app.get('/searchresult', (req, res) => {

res.sendFile(path.join(\_\_dirname, 'public', 'searchresult.html'));

});

app.post('/showbooking', async (req, res) => {

var mydb = mysql.createConnection({ host: "localhost", user: "root", password: "root123", database: "liftlink" });

var a1= "Pending";

mydb.connect(function (err) {

if (err) throw err;

console.log("connected");

mydb.query(" select \* from booking where status='"+a1+"'", function (err, result) {

if (err) throw err;

const fs = require('fs');

const databasedata = JSON.stringify(result);

const filePath = path.join(\_\_dirname, './public/writeshow.json');

fs.writeFile(filePath, databasedata, 'utf8', (err) => {

if (err) {

console.error('Error writing file:', err);

return;

}

console.log('JSON data has been saved to', filePath);

});

res.sendFile(path.join(\_\_dirname, 'public', 'showbooking.html'));

});

});

});

app.post('/updatebookingstatus', async (req, res) => {

try {

var a = req.body.BookingId;

var b = req.body.sid;

var c = req.body.status;

var con = mysql.createConnection({ host: "localhost", user: "root", password: "root123", database: "liftlink" });

con.connect(function (err) {

if (err) throw err;

console.log("connected");

con.query(" update booking set status='" + c + "' where BookingId='" + a + "' and sid='" + b + "'", function (err, result) {

if (err) throw err;

console.log("Database Updated");

});

});

res.sendFile(path.join(\_\_dirname, 'public', 'mybookingstatus.html'));

}

catch {

res.send("Internal server error");

}

});

app.post('/mybookingstatus', async (req, res) => {

try {

var a1 = req.body.status;

var con = mysql.createConnection({ host: "localhost", user: "root", password: "root123", database: "liftlink" });

con.connect(function (err) {

if (err) throw err;

console.log("connected");

con.query("select \* from booking where status='" + a1 + "'", function (err, result) {

const fs = require('fs');

const jsonString = JSON.stringify(result);

const filePath = 'public/mybookingstatus.json';

fs.writeFile(filePath, jsonString, 'utf8', (err) => {

if (err) {

console.error('Error writing file:', err);

return;

}

console.log('JSON data has been saved to', filePath);

});

res.sendFile(path.join(\_\_dirname, 'public', 'mybookingstatus.html'));

});

});

} catch

{

res.send("Internal server error");

}

});

app.post('/clregistration', async (req, res) => {

var mydb = mysql.createConnection({ host: "localhost", user: "root", password: "root123", database: "liftlink" });

try {

var a1 = req.body.name;

var a2 = req.body.address;

var a3 = req.body.aadhar;

var a4 = req.body.city;

var a5 = req.body.state;

var a6 = req.body.pincodes;

var a7 = req.body.account;

var a8 = req.body.charge;

var a9 = req.body.phoneno;

var a10 = req.body.email;

var a11 = req.body.vehicle\_reg;

var a12 = req.body.dl\_no;

var a13 = req.body.pollution;

var a14 = req.body.document\_clear;

var a15 = req.body.driving\_expe;

var a16 = req.body.username;

var a17 = req.body.password;

var a18 = req.body.repassword;

var a19 = req.body.sid;

var a20 = req.body.BookingId;

mydb.connect(function (err){

if (err) throw err;

console.log("connected");

mydb.query("INSERT INTO clregistration(name, address, aadhar, city, state, pincode, account, charge, phoneno, email, vehicle\_reg, dl\_no, pollution, document\_clear, driving\_expe, username, password, repassword,sid,BookingId) values('" + a1 + "','" + a2 + "','" + a3 + "','" + a4 + "','" + a5 + "','" + a6 + "','" + a7 + "','" + a8 + "','" + a9 + "','" + a10 + "','" + a11 + "','" + a12 + "','" + a13 + "','" + a14 + "','" + a15 + "','" + a16 + "','" + a17 + "','" + a18 + "','" + a19 + "','" + a20 + "')",function(err, result) {

if (err) throw err;

console.log("Record inserted.");

res.sendFile(path.join(\_\_dirname, 'public', 'login2.html'));

});

});

} catch (error) {

res.send("Internal server error");

}

});

app.post('/clientlogin', (req, res) => {

var mydb = mysql.createConnection({ host: "localhost", user: "root", password: "root123", database: "liftlink" });

var a1 = req.body.username;

var a2 = req.body.password;

mydb.query ( "SELECT \* FROM clregistration WHERE username='" + a1 + "' and password='" + a2 + "'",function(err, result) {

if (err) {

console.error('Error executing SQL query:', err);

return res.status(500).send('Internal Server Error');

}

const fs = require('fs');

const databasedata = JSON.stringify(result);

const filePath = path.join(\_\_dirname, './public/profile.json');

fs.writeFile(filePath, databasedata, 'utf8', (err) => {

if (err) {

console.error('Error writing file:', err);

return;

}

console.log('JSON data has been saved to', filePath);

});

if (result.length > 0) {

res.redirect('/access');

} else {

res.redirect('/unaccess');

}

});

});

app.get('/access', (req, res) => {

var mydb = mysql.createConnection({ host: "localhost", user: "root", password: "root123", database: "liftlink" });

mydb.connect(function (err) {

if (err) throw err;

console.log("connected");

mydb.query(" select \* from clregistration", function (err, result) {

if (err) throw err;

const fs = require('fs');

const databasedata = JSON.stringify(result);

const filePath = path.join(\_\_dirname, './public/profile.json');

fs.writeFile(filePath, databasedata, 'utf8', (err) => {

if (err) {

console.error('Error writing file:', err);

return;

}

console.log('JSON data has been saved to', filePath);

});

res.sendFile(path.join(\_\_dirname, 'public', 'dashboard2.html'));

});

});

});

app.get('/unaccess', (req, res) => {

res.sendFile(path.join(\_\_dirname, 'public', 'login2.html'));

});

app.post('/clientprofileupdate', async (req, res) => {

var mydb = mysql.createConnection({ host: "localhost", user: "root", password: "root123", database: "liftlink" });

try {

const { name, aadhar, address, city, pincode, state, account, phoneno, email,sid,BookingId } = req.body;

const sql = "UPDATE clregistration SET aadhar=?, address=?, city=?, pincode=?, state=?, account=?, phoneno=?, email=?, sid=?, BookingId=? WHERE name=?";

const values = [aadhar, address, city, pincode, state, account, phoneno, email, sid, BookingId, name];

mydb.query(sql, values, (err, result) => {

if (err) {

console.error("Error updating record: ", err);

res.status(500).send("Error updating record");

return;

}

console.log("Record updated.");

res.sendFile(path.join(\_\_dirname, 'public', 'clienteditprofile.html'));

});

} catch (error) {

console.error("Error:", error);

res.status(500).send("Update server error");

}

});

app.post('/book', async (req, res) => {

var mydb = mysql.createConnection({ host: "localhost", user: "root", password: "root123", database: "liftlink" });

try {

const { name, startingpoint, endingpoint, aadhar, phoneno, accountno, charge, date , BookingId, sid, status,CustomerId,lat,lon,accuracy } = req.body;

const sql = "INSERT INTO booking(name, startingpoint, endingpoint, aadhar, phoneno, accountno, charge, date , BookingId, sid, status,CustomerId,lat,lon,accuracy) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,?,?,?,?)";

const values = [name, startingpoint, endingpoint, aadhar, phoneno, accountno, charge, date , BookingId, sid, status, CustomerId,lat,lon,accuracy];

mydb.query(sql, values, function (err, result){

if (err) {

console.error("Error inserting record: ", err);

res.status(500).send("Error inserting record");

return;

}

console.log("Record inserted.");

if (result.length > 0) {

res.redirect('/geo');

} else {

res.redirect('/ungeo');

}

});

} catch (error) {

console.error("Error:", error);

res.status(500).send("Error processing request");

}

});

app.get('/geo', (req, res) => {

var mydb = mysql.createConnection({ host: "localhost", user: "root", password: "root123", database: "liftlink" });

mydb.connect(function (err) {

if (err) throw err;

console.log("connected");

mydb.query(" select \* from booking", function (err, result) {

if (err) throw err;

const fs = require('fs');

const databasedata = JSON.stringify(result);

const filePath = path.join(\_\_dirname, './public/geo.json');

fs.writeFile(filePath, databasedata, 'utf8', (err) => {

if (err) {

console.error('Error writing file:', err);

return;

}

console.log('JSON data has been saved to', filePath);

});

res.sendFile(path.join(\_\_dirname, 'public', 'dashboard2.html'));

});

});

});

app.get('/ungeo', (req, res) => {

var mydb = mysql.createConnection({ host: "localhost", user: "root", password: "root123", database: "liftlink" });

mydb.connect(function (err) {

if (err) throw err;

console.log("connected");

mydb.query(" select \* from booking", function (err, result) {

if (err) throw err;

const fs = require('fs');

const databasedata = JSON.stringify(result);

const filePath = path.join(\_\_dirname, './public/geo.json');

fs.writeFile(filePath, databasedata, 'utf8', (err) => {

if (err) {

console.error('Error writing file:', err);

return;

}

console.log('JSON data has been saved to', filePath);

});

res.sendFile(path.join(\_\_dirname, 'public', 'dashboard2.html'));

});

});});

app.post('/myride', async (req, res) => {

try {

var a1 = req.body.CustomerId;

var con = mysql.createConnection({ host: "localhost", user: "root", password: "root123", database: "liftlink" });

con.connect(function (err) {

if (err) throw err;

console.log("connected");

con.query("select \* from booking where CustomerId ='" + a1 + "'", function (err, result) {

const fs = require('fs');

const jsonString = JSON.stringify(result);

const filePath = 'public/myride.json';

fs.writeFile(filePath, jsonString, 'utf8', (err) => {

if (err) {

console.error('Error writing file:', err);

return;

}

console.log('JSON data has been saved to', filePath);

});

res.sendFile(path.join(\_\_dirname, 'public', 'myride2.html'));

});

});

} catch

{

res.send("Internal server error");

}

});

app.post('/mybooking', async (req, res) => {

try {

var a1 = req.body.CustomerId;

var a2 = "booked";

var con = mysql.createConnection({ host: "localhost", user: "root", password: "root123", database: "liftlink" });

con.connect(function (err) {

if (err) throw err;

console.log("connected");

con.query("select \* from booking where CustomerId ='" + a1 + "' AND status='"+a2+"'", function (err, result) {

const fs = require('fs');

const jsonString = JSON.stringify(result);

const filePath = 'public/mybooking.json';

fs.writeFile(filePath, jsonString, 'utf8', (err) => {

if (err) {

console.error('Error writing file:', err);

return;

}

console.log('JSON data has been saved to', filePath);

});

res.sendFile(path.join(\_\_dirname, 'public', 'mybooking2.html'));

});

});

} catch

{

res.send("Internal server error");

}

});

app.get('/cancel', (req, res) => {

res.sendFile(path.join(\_\_dirname, 'public', 'comfirmcancel.html'));

});

app.post('/comfirmcancel', async (req, res) => {

var mydb = mysql.createConnection({ host: "localhost", user: "root", password: "root123", database: "liftlink" });

try {

var a1 = req.body.BookingId;

var a2 = "Cancel";

const sql = "UPDATE booking SET status='"+a2+"' WHERE BookingId='"+a1+"'";

mydb.query(sql, (err, result) => {

if (err) {

console.error("Error updating record: ", err);

res.status(500).send("Error updating record");

return;

}

console.log("Record updated.");

res.sendFile(path.join(\_\_dirname, 'public', 'dashboard2.html'));

});

} catch (error) {

console.error("Error:", error);

res.status(500).send("Update server error");

}

});

app.post('/showcancel', async (req, res) => {

try {

var a1 = req.body.CustomerId;

var a2 = "Cancel";

var con = mysql.createConnection({ host: "localhost", user: "root", password: "root123", database: "liftlink" });

con.connect(function (err) {

if (err) throw err;

console.log("connected");

con.query("select \* from booking where CustomerId ='" + a1 + "' AND status='"+a2+"'", function (err, result) {

const fs = require('fs');

const jsonString = JSON.stringify(result);

const filePath = 'public/cancel.json';

fs.writeFile(filePath, jsonString, 'utf8', (err) => {

if (err) {

console.error('Error writing file:', err);

return;

}

console.log('JSON data has been saved to', filePath);

});

res.sendFile(path.join(\_\_dirname, 'public', 'cancel2.html'));

});

});

} catch

{

res.send("Internal server error");

}

});

app.post('/updatepostride', async (req, res) => {

var mydb = mysql.createConnection({ host: "localhost", user: "root", password: "root123", database: "liftlink" });

try {

const { name, address, aadhar, city, state, pincode, account, charge\_per\_unit, phoneno, email, vehicle\_reg, dl\_no, pollution, document\_clear, driving\_experience, ac, starting\_location, ending\_location, time\_taken } = req.body;

const sql = "UPDATE postride SET address=?,aadhar=?,city=?,state=?,pincode=?,account=?,charge\_per\_unit=?,phoneno=?,email=?,vehicle\_reg=?,dl\_no=?,pollution=?,document\_clear=?,driving\_experience=?,ac=?,starting\_location=?,ending\_location=?,time\_taken=? WHERE name=?";

const values = [name, address, aadhar, city, state, pincode, account, charge\_per\_unit, phoneno, email, vehicle\_reg, dl\_no, pollution, document\_clear, driving\_experience, ac, starting\_location, ending\_location, time\_taken];

mydb.query(sql, values, (err, result) => {

if (err) {

console.error("Error updating record: ", err);

res.status(500).send("Error updating record");

return;

}

console.log("Record updated.");

res.sendFile(path.join(\_\_dirname, 'public', 'updatepostride.html'));

});

} catch (error) {

console.error("Error:", error);

res.status(500).send("Update server error");

}

});

app.post('/rating', async (req, res) => {

var mydb = mysql.createConnection({ host: "localhost", user: "root", password: "root123", database: "liftlink" });

try {

var a1 = req.body.name;

var a2 = req.body.CustomerId;

var a3 = req.body.RiderId;

var a4 = req.body.date;

var a5 = req.body.BookingId;

var a6 = req.body.rate;

mydb.connect(function (err){

if (err) throw err;

console.log("connected");

mydb.query("INSERT INTO rating(name, CustomerId, RiderId, date, BookingId, rate) values('" + a1 + "','" + a2 + "','" + a3 + "','" + a4 + "','" + a5 + "','" + a6 + "')",function(err, result) {

if (err) throw err;

console.log("Record inserted.");

if (result.length > 0) {

res.redirect('/b');

} else {

res.redirect('/a');

}

});

});

} catch (error) {

res.send("Internal server error");

}

});

app.get('/a', async (req, res) => {

try {

var a1 =1;

var con = mysql.createConnection({ host: "localhost", user: "root", password: "root123", database: "liftlink" });

con.connect(function (err) {

if (err) throw err;

console.log("connected");

con.query("select \* from rating where RiderId ='" + a1 + "'", function (err, result) {

const fs = require('fs');

const jsonString = JSON.stringify(result);

const filePath = 'public/feedback.json';

fs.writeFile(filePath, jsonString, 'utf8', (err) => {

if (err) {

console.error('Error writing file:', err);

return;

}

console.log('JSON data has been saved to', filePath);

});

res.sendFile(path.join(\_\_dirname, 'public', 'rating2.html'));

});

});

} catch

{

res.send("Internal server error");

}

});

app.get('/b', async (req, res) => {

res.sendFile(path.join(\_\_dirname, 'public', 'rating.html'));

});

app.post('/rating2', async (req, res) => {

const path = require('path');

const fs = require('fs');

const filePath = path.resolve(\_\_dirname, 'public/feedback.json');

fs.readFile(filePath, 'utf8', (err, data) => {

if (err) {

console.error('Error reading the file:', err);

return;

}

let jsonData = JSON.parse(data);

let total = 0;

let count = jsonData.length;

for (let i = 0; i < count; i++) {

total = total + parseFloat(jsonData[i].rate); // Convert the rate to a float

};

const avg = total / count;

var a=1;

var con = mysql.createConnection({ host: "localhost", user: "root", password: "root123", database: "liftlink" });

con.connect(function (err) {

if (err) throw err;

console.log("connected");

con.query(" update postride set star='" + avg + "' where sid='" + a + "'", function (err, result) {

if (err) throw err;

console.log("Database Updated");

});

});

res.sendFile(path.join(\_\_dirname, 'public', 'owner\_after\_login.html'));

});

});

app.post('/mltime', async (req, res) => {

var mydb = mysql.createConnection({ host: "localhost", user: "root", password: "root123", database: "liftlink" });

try {

var a1 = req.body.name;

var a2 = req.body.CustomerId;

var a3 = req.body.RiderId;

var a4 = req.body.date;

var a5 = req.body.BookingId;

var a6 = req.body.traveltime;

mydb.connect(function (err){

if (err) throw err;

console.log("connected");

mydb.query("INSERT INTO time(name, CustomerId, RiderId, date, BookingId, time) values('" + a1 + "','" + a2 + "','" + a3 + "','" + a4 + "','" + a5 + "','" + a6 + "')",function(err, result) {

if (err) throw err;

console.log("Record inserted.");

if (result.length > 0) {

res.redirect('/ml');

} else {

res.redirect('/mltime2');

}

});

});

} catch (error) {

res.send("Internal server error");

}

});

app.get('/mltime2', async (req, res) => {

try {

var a1 =1;

var con = mysql.createConnection({ host: "localhost", user: "root", password: "root123", database: "liftlink" });

con.connect(function (err) {

if (err) throw err;

console.log("connected");

con.query("select \* from time where RiderId ='" + a1 + "'", function (err, result) {

const fs = require('fs');

const jsonString = JSON.stringify(result);

const filePath = 'public/time.json';

fs.writeFile(filePath, jsonString, 'utf8', (err) => {

if (err) {

console.error('Error writing file:', err);

return;

}

console.log('JSON data has been saved to', filePath);

});

res.sendFile(path.join(\_\_dirname, 'public', 'time2.html'));

});

});

} catch

{

res.send("Internal server error");

}

});

app.post('/r', async (req, res) => {

const path = require('path');

const fs = require('fs');

const filePath = path.resolve(\_\_dirname, 'public/time.json');

fs.readFile(filePath, 'utf8', (err, data) => {

if (err) {

console.error('Error reading the file:', err);

return;

}

let jsonData = JSON.parse(data);

let total = 0;

let count = jsonData.length;

for (let i = 0; i < count; i++) {

total = total + parseFloat(jsonData[i].time);

};

const avg = total / count;

let p=Math.round(avg);

var a=1;

var con = mysql.createConnection({ host: "localhost", user: "root", password: "root123", database: "liftlink" });

con.connect(function (err) {

if (err) throw err;

console.log("connected");

con.query(" update postride set time='" + p + "' where sid='" + a + "'", function (err, result) {

if (err) throw err;

console.log("Database Updated");

});

});

res.sendFile(path.join(\_\_dirname, 'public', 'owner\_after\_login.html'));

});

});

app.post('/continue', async (req, res) => {

var a1 = req.body.distance;

var a2 = req.body.time;

var a = req.body.riderid;

let speed=a1/a2;

var con = mysql.createConnection({ host: "localhost", user: "root", password: "root123", database: "liftlink" });

con.connect(function (err) {

if (err) throw err;

console.log("connected");

con.query(" update postride set speed='" + speed+ "' where sid='" + a + "'", function (err, result) {

if (err) throw err;

console.log("Database Updated");

});

});

res.sendFile(path.join(\_\_dirname, 'public', 'owner\_after\_login.html'));

});

server.listen(3000, () => {

console.log("Server is listening on port: 3000");

});

**COMPONENTS:**

BOOK RIDE.html

<!DOCTYPE html>

<html>

<head>

<link rel="stylesheet" href="book\_ride.css">

<link rel="icon" type="image/x-icon" href="/public/vaibhav.ico">

<title>Pooling system</title>

<script>

function enableInputs() {

let inputs = document.querySelectorAll('input[type="text"], input[type="number"], input[type="date"], select');

for (var i = 0; i < inputs.length; i++) {

inputs[i].removeAttribute('disabled');

}

}

function login(){

alert("Redirecting to Log-In Window ")

}

function showride() {

var xhttp;

if (window.XMLHttpRequest) {

xhttp = new XMLHttpRequest();

} else {

xhttp = new ActiveXObject("Microsoft.XMLHTTP");

}

xhttp.onreadystatechange = function () {

if (xhttp.readyState == 4 && xhttp.status == 200) {

const item = xhttp.responseText;

const data = JSON.parse(item);

document.getElementById("p1").innerHTML = data[0].name;

document.getElementById("p2").innerHTML = data[0].aadhar;

document.getElementById("p3").innerHTML = data[0].address;

document.getElementById("p4").innerHTML = data[0].city;

document.getElementById("p5").innerHTML = data[0].pincode;

document.getElementById("p6").innerHTML = data[0].state;

document.getElementById("p7").innerHTML = data[0].account;

document.getElementById("p8").innerHTML = data[0].phoneno;

document.getElementById("p9").innerHTML = data[0].email;

document.getElementById("p10").innerHTML = data[0].vehicle\_reg;

document.getElementById("p11").innerHTML = data[0].dl\_no;

document.getElementById("p12").innerHTML = data[0].pollution;

document.getElementById("p13").innerHTML = data[0].driving\_expe;

document.getElementById("p14").innerHTML = data[0].document\_clear;

}

}

xhttp.open("GET", "owner\_after\_login.json", true);

xhttp.send();

}

</script>

</script>

</head>

<body>

<header>

<div class="login">

<nav class="navtext">

<ul>

<a href="dashboard2.html" class="menu" >My Profile</a>

<a href="clienteditprofile.html" class="menu" target="\_blank">Edit Profile</a>

<a href="book\_ride.html" class="menu" >Book Ride</a>

<a href="update\_ride.html" class="menu" target="\_blank">Update Ride</a>

</ul>

</nav>

<img id="photo" src="location.png">

<h1 id="a">LIFT-LINK</h1>

<div class=login>

<div class="dropdown">

<img src="th.jpeg" alt="User Photo" class="dropbtn">

<div class="dropdown-content">

<a href="login2.html">Logout</a>

<a href="passwordreset.html">Change Password</a>

</div>

</div>

</div>

</div>

</header>

<br> <p>POOL & ROLL : SHARE THE RIDE, SAVE THE EARTH!</p> <br><br>

<marquee><p>Carpooling is the sharing of car journeys so that more than one person travels in a car, and prevents the need for others to have to drive to a location themselves. Carpooling is considered a Demand-Responsive Transport (DRT) service</p></marquee>

<div class="column3">

<nav>

<ul>

<div>

<a onclick="login()" href="searchride.html">

<img id="search"src="https://cdn-icons-png.flaticon.com/128/751/751463.png" alt="Icon" class="nav-icon"><br>

<div id="e">Search Cab</div></a>

</div>

<div>

<a onclick="login()" href="parking.html">

<img id="search"src="https://img.icons8.com/?size=100&id=12684&format=png"><br>

<div id="e1">Lift-Link Parking!</div></a>

</div>

<div>

<a onclick="login()" href="#">

<img id="search"src="https://img.icons8.com/?size=100&id=44471&format=png"><br>

<div id="e2">Service</div></a>

</div>

<div>

<aonclick="login()" href="https://mail.google.com/mail/u/1/#inbox?compose=DmwnWrRqjCHQxQhvQHKpdWvrqjcDFdxxwgXCjGVkcmqSKBfXrnwgHTXdJrnkGfbVbQxlfhCwrsSv">

<imgid="search"src="https://cdn-icons-png.flaticon.com/128/3249/3249904.png"><br>

<div id="e3">Contact us?</div></a>

</div>

</ul>

</nav>

</div>

</header>

<body>

<div id="set">

<div class="teama">

<div class="b">

<ul id="teamatext">

<div class="dp" style="text-align: center;"><img src="th.jpeg" alt="Paris">

<li>

<input type="file" name="photo">

</li>

</div>

<a href="dashboard2.html" class="p">My Profile</a>

<a href="clienteditprofile.html" class="p">Edit Profile</a>

<a href="myride.html" class="p">My Ride</a>

<a href="mybooking.html" class="p">Booking</a>

<a href="cancel.html" class="p">Cancel Ride</a>

</ul>

</div>

</div>

<div class="teamb">

<div class="d">

<form action="/bookride" method="POST">

<div class="update">

<h1 id="title">Book Ride</h1>

</div>

<table>

<tr>

<td class="td1">Name:</td>

<td><input type="text" name="name" id="p1" placeholder="Enter your name" ></td>

<td class="td">Aadhaar No:</td>

<td class="in"><input type="text" name="aadhar" id="p2" placeholder="Enter your Aadhaar number" ></td>

</tr>

<tr>

<td class="td1">Address:</td>

<td><input type="text" name="country" id="p3" placeholder="Enter your address." ></td>

<td class="td">City:</td>

<td class="in"><input type="text" name="city" id="p4" placeholder="Enter your city" ></td>

</tr>

<tr>

<td class="td1">Pin-Code:</td>

<td><input type="number" name="pincode" id="p5" placeholder="Enter your pin code" ></td>

<td class="td">State:</td>

<td class="in"><input type="text" name="state" id="p6" placeholder="Enter your state" ></td>

</tr>

<tr>

<td class="td1">Phone Number:</td>

<td><input type="text" name="phoneno" id="p8" placeholder="Enter your phone number" ></td>

<td class="td">Email:</td>

<td class="in"><input type="text" name="email" id="p9" placeholder="Enter your email" ></td>

</tr>

<tr>

<td class="td1">AC?:</td>

<td class="in">

<select name="ac" >

<option value="choose">Choose your option</option>

<option value="yes">Yes</option>

<option value="no">No</option>

</select>

</td>

</tr>

<tr>

<td class="td1">Destination A:</td>

<td><input type="text" name="starting\_location" placeholder="Starting Point..." ></td>

<td class="td">Destination B:</td>

<td><input type="text" name="ending\_location" placeholder="Ending Point" ></td>

</tr>

<tr>

<td class="td1">Date:</td>

<td><input type="date" name="date\_taken" placeholder="time\_taken" ></td>

<td class="td">Time:</td>

<td><input type="time" name="time\_taken" placeholder="time\_taken" ></td>

</tr>

<tr>

<td class="td1">PIN:</td>

<td><input type="password" name="pin" placeholder="enter 4 number pin" ></td>

<td class="td">Booking Number:</td>

<td><input type="text" name="booking\_number" placeholder="time\_taken" ></td>

</tr>

<div class="status">Status:</div>

<input type="text" name="Status" placeholder="status" >

</form>

<table>

<tr>

<td class="td1">Rider Name:</td>

<td><input type="text" name="rname" placeholder="Enter your name" ></td>

<td class="td">Rider Aadhaar No:</td>

<td class="in"><input type="text" name="raadhar" placeholder="Enter your Aadhaar number" ></td>

</tr>

<tr>

<td class="td1">Rider Country:</td>

<td><input type="text" name="rcountry" placeholder="Enter your Country" ></td>

<td class="td">Rider City:</td>

<td class="in"><input type="text" name="rcity" placeholder="Enter your city" ></td>

</tr>

<tr>

<td class="td1">Rider Pin-Code:</td>

<td><input type="number" name="rpincode" placeholder="Enter your pin code" ></td>

<td class="td">Rider State:</td>

<td class="in"><input type="text" name="rstate" placeholder="Enter your state" ></td>

</tr>

<tr>

<td class="td1">Rider Account Number:</td>

<td><input type="text" name="raccount" placeholder="Enter your account number" ></td>

<td class="td">Rider Charge:</td>

<td class="in"><input type="number" name="rcharge\_per\_unit" placeholder="Enter charges per km" ></td>

</tr>

<tr>

<td class="td1">Rider Phone Number:</td>

<td><input type="text" name="rphoneno" placeholder="Enter your phone number" ></td>

<td class="td">Rider Email:</td>

<td class="in"><input type="text" name="remail" placeholder="Enter your email" ></td>

</tr>

<tr>

<td class="td1">Rider Vehicle Registration:</td>

<td><input type="text" name="rvehicle\_reg" placeholder="Enter vehicle registration number" ></td>

<td class="td">Rider Pollution Expiry Date:</td>

<td><input type="date" name="rPollution" placeholder="Enter pollution expiry date" ></td>

</tr>

<tr>

<td class="td1">Rider Document cleared:</td>

<td class="in">

<select name="rdocument\_clear" >

<option value="choose">Choose your option</option>

<option value="yes">Yes</option>

<option value="no">No</option>

</select>

</td>

<td class="td">Rider DL Number:</td>

<td class="in"><input type="text" name="rdl\_no" placeholder="Enter DL number" ></td>

</tr>

<tr>

<td class="td1">Rider Driving Experience:</td>

<td><input type="number" name="rdriving\_experience" placeholder="Enter driving experience" ></td>

<td class="td">Rider AC?:</td>

<td class="in">

<select name="rac" >

<option value="choose">Choose your option</option>

<option value="yes">Yes</option>

<option value="no">No</option>

</select>

</td>

</tr>

<tr>

<td class="td1">Rider Starting Point:</td>

<td><input type="text" name="rstarting\_location" placeholder="Starting Point..." ></td>

<td class="td">Rider Ending Point:</td>

<td><input type="text" name="rending\_location" placeholder="Ending Point" ></td>

</tr>

<tr>

<td class="td1">Rider Time:</td>

<td><input type="time" name="rtime\_taken" placeholder="time\_taken" ></td>

</tr>

<br><br> <td><button id="submit" type="submit">Submit</button></td>

</table>

</div>

</div>

</div>

</body>

<footer>

<div class="footer-section">

<ul>

<li><a href="https://mail.google.com/">Contact us?</a></li>

<li><a href="https://mail.google.com/">call: 6397918693</a></li>

<li><a href="https://mail.google.com/">Mail:rautelavaibhav01@gmail.com </a></li>

</ul>

</div>

<div class="footer-section">

<ul>

<li><a href="#">Our Service</a></li>

<li><a href="https://mail.google.com/">call: 6397918693</a></li>

<li><a href="https://mail.google.com/">Mail:rautelavaibhav01@gmail.com </a></li>

</ul>

</div>

<div class="footer-section">

<ul>

<li><a href="#">Privacy Policy</a></li>

<li><a href="https://mail.google.com/">call: 6397918693</a></li>

<li><a href="https://mail.google.com/">Mail:rautelavaibhav01@gmail.com </a></li>

</ul>

</div>

<div class="footer-section">

<ul>

<li><a href="#">Terms & Condition</a></li>

<li><a href="https://mail.google.com/">call: 6397918693</a></li>

<li><a href="https://mail.google.com/">Mail:rautelavaibhav01@gmail.com </a></li>

</ul>

</div>

</footer>

<div id="foot">MAJOR PROJECT Copyright © 2023-2024- All rights reserved || Designed By: Vaibhav Rautela </div>

</html>

**CSS:**

\* {

margin: 0;

padding: 0;

font-family: "futura md bt";

text-decoration: none;

list-style: none;

box-sizing: border-box;

}

.login {

background-color: red;

padding: 5px;

margin: 0%;

display: flex;

text-decoration: none;

border-radius: 20px;

}

.navtext {

color: white;

padding: 12px;

font-weight: 500;

display: flex;

font-size: 17px;

}

.navtext ul {

list-style: none;

padding: 0;

}

.login .menu {

list-style-type: none;

}

.login ul .menu {

color: white;

text-decoration: none;

padding-right:20px ;

}

.navtext ul .menu:hover {

padding: 15px;

border-radius: 20px;

text-decoration: underline;

text-decoration-color: rgb(0, 0, 0);

text-decoration-thickness: 3px;

margin-top: 2px;

transition-duration: 0.4s;

}

#profile{text-decoration: underline;

text-decoration-color: blue;

text-decoration-thickness: 3px; }

.loginreg{

color: white;

border-radius: 5px;

padding: 10px 20px;

position:absolute;

top:25px;

right:23px;

text-decoration: none;

font-size: large;

}

header .login .loginreg:hover {

color:yellow;

transition-duration:1s ;

}

.update{

display: flex;

}

.status{

color:black;

}

#Update {

height: 40px;

width: 95px;

color: white;

background-color: rgb(202, 42, 42);

border-radius: 5px;

position:absolute;

top:25px;

right:15px;

}

#Update:hover {

background-color: rgb(43, 42, 42);

}

#sub{

position:relative;

top:90px;

left:30px;

}

.dp {

position: relative;

left: 11%;

display: inline-block;

border-radius: 50%; /\* Changed \*/

width: 150px;

height: 150px;

border: 1px solid #ccc;

padding: 5px;

box-sizing: border-box;

overflow: hidden;

}

.dp img {

width: 100%;

height: 100%;

object-fit: cover;

border-radius: 50%; /\* Added \*/

}

.dp input[type="file"] {

position: absolute;

left: 0;

top: 0;

width: 100%;

height: 100%;

opacity: 0;

cursor: pointer;

}

#title{

margin-top: 20px;

position: relative;

left:320px;

}

#private{

background-color: red;

height: 20px;

float: left;

}

.private{

color:white;

padding:20px;

display: inline-block;

}

#b{

background-color: brown;

align-items: center;

}

#p{

display: inline-block;

position: relative;

top: 18px;

}

#set {

display: flex;

}

.teama {

background-color: red;

border-radius: 15px;

position: relative;

height: 550px;

width: 30%;

z-index: 1;

}

.teamb {

background-color: wheat;

border-radius: 15px;

position: relative;

height: 1200px;

width: 70%;

text-align: center;

}

.teama .p {

position: relative;

left: 20%;

top:-15px;

color:white;

display: flex;

flex-direction: column;

}

.teama .p:hover {

padding: 10px;

}

#teamatext{

color:white;

line-height: 60px;

padding: 30px;

position: relative;

left: 55px;

top:10px;

}

.grey-background {

background-color: rgb(235, 234, 234);

padding: 10px;

margin:5px;

}

#navtext {

color: black;

display: inline-block;

}

.navbar {

background-color: red;

width: 100%;

height: 70px;

display: flex;

align-items: center;

justify-content: space-between;

padding: 0 50px;

position: relative;

}

.hamburger {

cursor: pointer;

display: block;

position: absolute;

left: 25px;

top:20px;

}

.hamburger div {

width: 30px;

height: 3px;

margin: 5px 0;

background: #000;

font-weight: bold;

}

.navbar .logo {

font-size: 30px;

text-transform: uppercase;

}

.navbar nav ul {

display: flex;

}

.navbar nav ul li a {

display: inline-block;

color: #000;

padding: 5px 0;

margin: 0 10px;

border: 3px solid transparent;

text-transform: uppercase;

transition: 0.2s;

color: white;

}

.navbar nav ul li a:hover,

.navbar nav ul li a.active {

border-bottom-color: black;

}

.navbar nav {

position: absolute;

width: 100%;

top: 70px;

left: -100%;

background: #fff;

padding: 0px;

line-height: 9.5;

width:20%;

transition: left 0.3s ease;

background: red;

}

#nav\_check:checked ~ .overlay {

background-color: rgba(0, 0, 0, 0.5);

z-index: 999;

position: fixed;

top: 0;

left: 0;

width: 100%;

height: 100%;

}

.navbar #nav\_check:checked ~ nav {

left: 0;

z-index: 999;

}

#navtext span {

display: inline-block;

color: #000;

padding: 0px 0;

margin: 0 20px;

border: 3px solid transparent;

text-transform: uppercase;

transition: 0.3s;

color: white;

border-radius: 10px;

}

#navtext span:hover {

padding: 20px;

}

.nav a {

text-decoration: none;

}

#search {

height: 50px;

}

.flex {

display: flex;

justify-content: flex-end;

}

#ya{

padding-right: 10px;

}

.td {

position: static;

top:120px;

display: flex;

justify-content: start;

left: 12px;

padding-left: 120px;

}

.td1{

display: flex;

justify-content: flex-start;

padding: 15px;

padding-left: 50px;

}

#submit {

height: 40px;

width: 95px;

color: white;

background-color: black;

border-radius: 5px;

position:absolute;

left:350px;

}

#submit:hover {

background-color: rgb(43, 42, 42);

}

.login {

background-color: red;

padding: 17px;

margin: 0%;

margin-bottom: 20px;

display: flex;

text-decoration: none;

}

.dropdown {

position: absolute;

top:20px;

display: inline-block;

right: 90px;

}

.dropdown-content {

display: none;

position: absolute;

background-color: #f9f9f9;

min-width: 160px;

box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);

z-index: 1;

}

.dropdown-content a {

color: black;

padding: 12px 16px;

text-decoration: none;

display: block;

}

.dropdown-content a:hover {

background-color: #f1f1f1;

}

.dropdown:hover .dropdown-content {

display: block;

}

.dropdown:hover .dropbtn {

background-color: #3e8e41;

}

.dropbtn {

position: absolute;

border-radius: 90px;

height: 50px;

width: 60px;

}

#photo{

position:absolute;

top:20px;

left:500px;

height: 60px;

width:60px;

}

#a {

text-align: left;

color: white;

position:absolute;

top:30px;

left:570px;

}

h1 {

display: inline-block;

margin: 0;

}

.column2 {

background-color: white;

padding: 10px;

margin: 0px;

}

.column3 {

display:flex;

flex-direction:inline;

padding: 20px;

}

.column3 a {

text-decoration: none;

color: black;

padding: 15px 50px;

border-radius: 10px;

}

.column3 nav ul{

display:flex;

flex-direction:inline;

}

.topic {

background-color: white;

padding: 20px;

margin: 20px;

}

.footer a {

margin: 50px;

text-decoration: none;

color: white;

background-color: black;

padding: 15px 50px;

border-radius: 15px;

}

nav ul {

padding: 0;

}

nav ul li {

display: inline;

}

nav ul li a {

text-decoration: none;

}

footer {

background-color: black;

color: white;

padding: 120px 0;

text-align: center;

}

.footer a:hover {

background-color: gray;

}

header .login .log {

color: white;

border-radius: 5px;

padding: 10px 20px;

position:absolute;

top:25px;

right:80px;

text-decoration: none;

height: 50px;

font-size: large;

}

header .login .log:hover {

color:yellow;

transition-duration:1s ;

}

.form-container {

background-color: #f8f8f8;

border-radius: 10px;

padding: 20px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

}

.form-container h1 {

margin-bottom: 20px;

color: #333;

}

.form-container input[type="text"],

.form-container input[type="number"],

.form-container input[type="date"],

.form-container input[type="password"],

.form-container select {

width: 100%;

padding: 10px;

margin-bottom: 5px;

margin-top: 30px;

border-radius: 5px;

box-sizing: border-box;

padding: 10px;

}

.form-container button {

width: 100%;

padding: 10px;

background-color: #007bff;

color: white;

border: none;

border-radius: 5px;

cursor: pointer;

}

.form-container button:hover {

background-color: #0056b3;

}

footer {

background-color: black;

color: white;

padding: 20px;

text-align: center;

display:flex;

flex-direction:inline;

}

#e{

position:absolute;

left:170px;

}

#e1{

position:absolute;

left:450px;

}

#e2{

position:absolute;

left: 780px;

}

#e3{

position:absolute;

left:1070px;

}

#search{

height:50px;

padding-left: 100px;

padding-right: 100px;

}

.footer-section ul {

list-style-type: none;

padding: 0;

}

.footer-section ul li {

margin-bottom: 40px;

}

.footer-section ul li a {

position:relative;

top:55px;

left:10px;

color: white;

text-decoration: none;

transition: color 0.3s ease;

}

.call{

position:absolute;

left:150px;

}

.footer-section{

height:250px;

width:90%;

}

.footer-section ul li a:hover {

color: grey; }

#foot{

background-color: grey;

padding: 12px;

color:white;

text-align: center;

}

**BOOK.html**

<!DOCTYPE html>

<html>

<head>

<link rel="stylesheet" href="book.css">

<link rel="icon" type="image/x-icon" href="/public/vaibhav.ico">

<title>Pooling system</title>

</head>

<script>

function showride() {

var xhttp;

if (window.XMLHttpRequest) {

xhttp = new XMLHttpRequest();

} else {

xhttp = new ActiveXObject("Microsoft.XMLHTTP");

}

xhttp.onreadystatechange = function () {

if (xhttp.readyState == 4 && xhttp.status == 200) {

const item = xhttp.responseText;

const data = JSON.parse(item);

document.getElementById("v1").value = data[0].name;

document.getElementById("v2").value = data[0].address;

document.getElementById("v3").value = data[0].aadhar;

document.getElementById("v4").value = data[0].city;

document.getElementById("v6").value = data[0].pincode;

document.getElementById("v7").value = data[0].account;

document.getElementById("v8").value = data[0].charge\_per\_unit;

document.getElementById("v9").value = data[0].phoneno;

document.getElementById("v10").value = data[0].email;

document.getElementById("v12").value = data[0].sid;

document.getElementById("v13").value = data[0].starting\_location;

document.getElementById("v14").value = data[0].ending\_location;

document.getElementById("v15").value = data[0].time\_taken;

}

}

xhttp.open("GET", "search1.json", true);

xhttp.send();

}

function getLocation() {

if (navigator.geolocation) {

navigator.geolocation.getCurrentPosition(showPosition);

} else {

document.getElementById("Lat").innerHTML = "Geolocation is not supported by this browser.";

}

}

function showPosition(position) {

document.getElementById("Lat").value = position.coords.latitude;

document.getElementById("Lon").value = position.coords.longitude;

document.getElementById("accuracy").value = position.coords.accuracy;

}

</script>

<body onload="showride(); getLocation()">

<header>

<div class="login">

<nav class="navtext">

<ul>

<a href="owner\_after\_login.html" class="menu" target="\_blank">My Profile</a>

<a href="edit\_profile.html" class="menu" target="\_blank">Edit Profile</a>

<a href="update\_ride.html" class="menu" target="\_blank">Update Ride</a>

</ul>

</nav>

<img id="photo" src="location.png">

<h1 id="a">LIFT-LINK</h1>

<div class="login">

<div class="dropdown">

<img src="th.jpeg" alt="User Photo" class="dropbtn">

<div class="dropdown-content">

<a href="owner\_login.html">Logout</a>

<a href="passwordreset.html">Change Password</a>

</div>

</div>

</div>

</div>

</header>

<br>

<p>POOL & ROLL : SHARE THE RIDE, SAVE THE EARTH!</p>

<br><br>

<marquee>

<p>Carpooling is the sharing of car journeys so that more than one person travels in a car, and prevents the

need for others to have to drive to a location themselves. Carpooling is considered a Demand-Responsive

Transport (DRT) service</p>

</marquee>

<div class="column3">

<nav>

<ul>

<div>

<a onclick="login()" href="searchride.html">

<img id="search" src="https://cdn-icons-png.flaticon.com/128/751/751463.png" alt="Icon"

class="nav-icon"><br>

<div id="e">Search Cab</div>

</a>

</div>

<div>

<a onclick="login()" href="parking.html">

<img id="search" src="https://img.icons8.com/?size=100&id=12684&format=png"><br>

<div id="e1">Lift-Link Parking!</div>

</a>

</div>

<div>

<a onclick="login()" href="#">

<img id="search" src="https://img.icons8.com/?size=100&id=44471&format=png"><br>

<div id="e2">Service</div>

</a>

</div>

<div>

<a onclick="login()"

href="https://mail.google.com/mail/u/1/#inbox?compose=DmwnWrRqjCHQxQhvQHKpdWvrqjcDFdxxwgXCjGVkcmqSKBfXrnwgHTXdJrnkGfbVbQxlfhCwrsSv">

<img id="search" src="https://cdn-icons-png.flaticon.com/128/3249/3249904.png"><br>

<div id="e3">Contact us?</div>

</a>

</div>

</ul>

</nav>

</div>

<div class="img">

<img class="pic" src="dashboard.jpg">

<form action="/book" method="POST">

<h1 id="title">Confirm Ride</h1>

<table>

<tr>

<td><b>Name:

<div class="a"><input type="text" name="name" id="v1" readonly></b></div>

</td>

<td><b>Address:

<div class="a"><input type="text" name="Address" id="v2" readonly></b></div>

</td>

<td><b>aadhar:

<div class="a"><input type="text" name="aadhar" id="v3"></b></div>

</td>

<td><b>City:

<div class="a"><input type="text" name="City" id="v4" readonly></b></div>

</td>

<td><b>Pin-Code:

<div class="a"><input type="text" name="pincode" id="v6" readonly></b></div>

</td>

<td><b>Account No.:

<div class="a"><input type="text" name="account" id="v7" readonly></b></div>

</td>

<td><b>Charge:

<div class="a"><input type="text" name="charge" id="v8" readonly></b></div>

</td>

<td><b>Phone Number:

<div class="a"><input type="text" name="phoneno" id="v9" readonly></b></div>

</td>

<td><b>Email:

<div class="a"><input type="text" name="email" id="v10" readonly></b></div>

</td>

<td><b>Customer Id:

<div class="a"><input type="text" name="CustomerId" ></b></div>

</td>

<td>

<b>Rider-Id:

<div class="a"><input type="text" name="sid" id="v12"></b></div>

</td>

<td>

<b>Booking Id:

<div class="a"><input type="text" name="BookingId" ></b></div>

</td>

<td><b>Starting Point:

<div class="a"><input type="text" name="startingpoint" id="v13"></b></div>

</td>

<td><b>Ending Point:

<div class="a"><input type="text" name="endingpoint" id="v14"></b></div>

</td>

<td><b>Date:

<div class="a"><input type="text" name="date" id="v15"></b></div>

</td>

<td>

<b>Status:

<div class="a"><input type="text" value="Pending" name="status" id="v16" readonly></b></div>

</td>

<input type="text" id="Lat" name="lat" >

<input type="text" id="Lon" name="lon" >

<input type="text" id="accuracy" name="accuracy" >

</tr>

<tr>

<td>

<div class="a"><input type="submit" value="Confirm Ride" id="butt"></div>

</td>

</tr>

</table>

</form>

</div>

<footer>

<div class="footer-section">

<ul>

<li><a href="https://mail.google.com/">Contact us?</a></li>

<li><a href="https://mail.google.com/">call: 6397918693</a></li>

<li><a href="https://mail.google.com/">Mail:rautelavaibhav01@gmail.com </a></li>

</ul>

</div>

<div class="footer-section">

<ul>

<li><a href="#">Our Service</a></li>

<li><a href="https://mail.google.com/">call: 6397918693</a></li>

<li><a href="https://mail.google.com/">Mail:rautelavaibhav01@gmail.com </a></li>

</ul>

</div>

<div class="footer-section">

<ul>

<li><a href="#">Privacy Policy</a></li>

<li><a href="https://mail.google.com/">call: 6397918693</a></li>

<li><a href="https://mail.google.com/">Mail:rautelavaibhav01@gmail.com </a></li>

</ul>

</div>

<div class="footer-section">

<ul>

<li><a href="#">Terms & Condition</a></li>

<li><a href="https://mail.google.com/">call: 6397918693</a></li>

<li><a href="https://mail.google.com/">Mail:rautelavaibhav01@gmail.com </a></li>

</ul>

</div>

</footer>

<div id="foot">MAJOR PROJECT Copyright © 2023-2024- All rights reserved || Designed By: Vaibhav Rautela </div>

</body>

</html>

Dashboard

<!DOCTYPE html>

<html>

<head>

<link rel="stylesheet" href="owner\_after\_login.css">

<link rel="icon" type="image/x-icon" href="/public/vaibhav.ico">

<title>Pooling system</title>

<script>

function showride() {

var xhttp;

if (window.XMLHttpRequest) {

xhttp = new XMLHttpRequest();

} else {

xhttp = new ActiveXObject("Microsoft.XMLHTTP");

}

xhttp.onreadystatechange = function () {

if (xhttp.readyState == 4 && xhttp.status == 200) {

const item = xhttp.responseText;

const data = JSON.parse(item);

document.getElementById("p1").innerHTML = data[0].name;

document.getElementById("p2").innerHTML = data[0].address;

document.getElementById("p3").innerHTML = data[0].aadhar;

document.getElementById("p4").innerHTML = data[0].city;

document.getElementById("p5").innerHTML = data[0].state;

document.getElementById("p6").innerHTML = data[0].pincode;

document.getElementById("p7").innerHTML = data[0].account;

document.getElementById("p9").innerHTML = data[0].phoneno;

document.getElementById("p10").innerHTML = data[0].email;

document.getElementById("p16").innerHTML = data[0].password;

document.getElementById("p17").innerHTML = data[0].username;

document.getElementById("p18").innerHTML = data[0].repassword;

document.getElementById("p19").innerHTML = data[0].sid;

document.getElementById("p20").innerHTML = data[0].BookingId;

}

}

xhttp.open("GET", "profile.json", true);

xhttp.send();

}

</script>

</head>

<header>

<div class="login">

<nav class="navtext">

<ul>

<a href="dashboard2.html" class="menu" target="\_blank">My Profile</a>

<a href="clienteditprofile.html" class="menu" target="\_blank">Edit Profile</a>

<a href="update\_ride.html" class="menu" target="\_blank">Update Ride</a>

<a href="rating.html" class="menu" >Feedback</a>

<a href="time.html" class="menu">Time</a>

<a href="continue.html" class="menu">Speed</a>

</ul>

</nav>

<img id="photo" src="location.png">

<h1 id="a">LIFT-LINK</h1>

<div class=login>

<div class="dropdown">

<img src="th.jpeg" alt="User Photo" class="dropbtn">

<div class="dropdown-content">

<a href="login2.html">Logout</a>

<a href="passwordreset.html">Change Password</a>

</div>

</div>

</div>

</div>

</header>

<br> <p>POOL & ROLL : SHARE THE RIDE, SAVE THE EARTH!</p> <br><br>

<marquee><p>Carpooling is the sharing of car journeys so that more than one person travels in a car, and prevents the need for others to have to drive to a location themselves. Carpooling is considered a Demand-Responsive Transport (DRT) service</p></marquee>

<div class="column3">

<nav>

<ul>

<div>

<a onclick="login()" href="searchride.html">

<img id="search"src="https://cdn-icons-png.flaticon.com/128/751/751463.png" alt="Icon" class="nav-icon"><br>

<div id="e">Search Cab</div></a>

</div>

<div>

<a onclick="login()" href="parking.html">

<img id="search"src="https://img.icons8.com/?size=100&id=12684&format=png"><br>

<div id="e1">Lift-Link Parking!</div></a>

</div>

<div>

<a onclick="login()" href="#">

<img id="search"src="https://img.icons8.com/?size=100&id=44471&format=png"><br>

<div id="e2">Service</div></a>

</div>

<div>

<a onclick="login()" href="https://mail.google.com/mail/u/1/#inbox?compose=DmwnWrRqjCHQxQhvQHKpdWvrqjcDFdxxwgXCjGVkcmqSKBfXrnwgHTXdJrnkGfbVbQxlfhCwrsSv">

<img id="search"src="https://cdn-icons-png.flaticon.com/128/3249/3249904.png"><br>

<div id="e3">Contact us?</div></a>

</div>

</ul>

</nav>

</div>

<div id="set">

<div class="teama">

<div class="b">

<ul id="teamatext">

<div class="dp" style="text-align: center;"><img src="th.jpeg" alt="Paris">

<li>

<input type="file" name="photo">

</li>

</div>

<a href="dashboard2.html" class="p">My Profile</a>

<a href="clienteditprofile.html" class="p">Edit Profile</a>

<a href="myride.html" class="p">My Ride</a>

<a href="mybooking.html" class="p">Booking</a>

<a href="cancel.html" class="p">Cancel Ride </a>

</ul>

</div>

</div>

<body onload="showride()">

<div class="teamb">

<div class="d">

<form action="/" method="POST">

<h1 id="title">Client Way</h1>

<table>

<tr>

<td class="td1"><b>Name:</b> <div id="p1"></div></td>

<td class="td"><b>Aadhaar No:</b> <div id="p3"></div> </td>

<td class="td1"><b>Address:</b> <div id="p2"></div> </td>

<td class="td"><b>City:</b> <div id="p4"></div></td>

<td class="td1"><b>Pin-Code:</b><div id="p6"></div> </td>

<td class="td"><b>State:</b> <div id="p5"></div> </td>

<td class="td1"><b>Account Number:</b> <div id="p7"></div> </td>

<td class="td"><b>Phone Number:</b> <div id="p9"></div> </td>

<td class="td1"><b>Email:</b> <div id="p10"></div> </td>

<td class="td"><b>S Id:</b> <div id="p19"></div> </td>

<td class="td1"><b>Booking Id:</b> <div id="p20"></div> </td>

</tr>

</table>

</form>

</div>

</div>

</div>

</body>

<footer>

<div class="footer-section">

<ul>

<li><a href="https://mail.google.com/">Contact us?</a></li>

<li><a href="https://mail.google.com/">call: 6397918693</a></li>

<li><a href="https://mail.google.com/">Mail:rautelavaibhav01@gmail.com </a></li>

</ul>

</div>

<div class="footer-section">

<ul>

<li><a href="#">Our Service</a></li>

<li><a href="https://mail.google.com/">call: 6397918693</a></li>

<li><a href="https://mail.google.com/">Mail:rautelavaibhav01@gmail.com </a></li>

</ul>

</div>

<div class="footer-section">

<ul>

<li><a href="#">Privacy Policy</a></li>

<li><a href="https://mail.google.com/">call: 6397918693</a></li>

<li><a href="https://mail.google.com/">Mail:rautelavaibhav01@gmail.com </a></li>

</ul>

</div>

<div class="footer-section">

<ul>

<li><a href="#">Terms & Condition</a></li>

<li><a href="https://mail.google.com/">call: 6397918693</a></li>

<li><a href="https://mail.google.com/">Mail:rautelavaibhav01@gmail.com </a></li>

</ul>

</div>

</footer>

<div id="foot">MAJOR PROJECT Copyright © 2023-2024- All rights reserved || Designed By: Vaibhav Rautela </div>

</html>

**MY BOOKING STATUS:**

<!DOCTYPE html>

<html>

<head>

<link rel="stylesheet" href="mybookingstatus.css">

<title>Pooling system</title>

<link rel="stylesheet" href="mybookingstatus2.css">

<link rel="icon" type="image/x-icon" href="/public/vaibhav.ico">

</head>

<script>

function showride() {

var xhttp = new XMLHttpRequest();

xhttp.onreadystatechange = function () {

if (xhttp.readyState == 4 && xhttp.status == 200) {

const item = xhttp.responseText;

const data = JSON.parse(item);

let text = "<table border='1' cellpadding='0' cellspacing='0'>";

text +="<th>Name</th>

<th>Starting location</th>

<th>Ending location</th>

<th>Aadhar No.</th>

<th>Registration No.</th>

<th>DL No.</th><th>Pollution</th>

<th>Document Clear</th>

<th>Driving Experience</th>

<th>Phone No.</th>

<th>Account</th>

<th>Charge per km</th>

<th>Date</th>

<th>S.ID</th>

<th>Status</th>

<th>Booking Id</th>"

for (let x in data) {

text += "<tr><td >" +data[x].name + "</b><br></td>

<td >"+ data[x].startingpoint + "</b></td>

<td >" +data[x].endingpoint + "</b></td>

<td >" + data[x].aadhar + "</td>

<td >" + data[x].vehicle\_reg + "</td>

<td>" + data[x].dll\_no + "</td>

<td >" + data[x].pollution + "</td>

<td >" + data[x].document\_clear + "</td>

<td >" + data[x].driving\_exper + "</td>

<td >" + data[x].phoneno + "</td>

<td >" + data[x].accountno + "</td>

<td >" + data[x].charge + "</td>

<td >" + data[x].date + "</td>

<td >"+data[x].sid + "</b></td>

<td >" + data[x].status +"</td>

<td>" + data[x].BookingId + "</b></td>

<td><a id='c' href='geolocation.html'>Get in for Ride.</a></td></tr>";

}

text += "</table>";

document.getElementById("v1").innerHTML = text;

}

}

xhttp.open("GET", "mybookingstatus.json", true);

xhttp.send();

}

</script>

<header>

<div class="login">

<nav class="navtext">

<ul>

<a href="owner\_after\_login.html" class="menu" target="\_blank">My Profile</a>

<a href="edit\_profile.html" class="menu" target="\_blank">Edit Profile</a>

<a href="book\_ride.html" class="menu" target="\_blank">Book Ride</a>

<a href="update\_ride.html" class="menu" target="\_blank">Update Ride</a>

</ul>

</nav>

<img id="photo" src="location.png">

<h1 id="a">LIFT-LINK</h1>

<div class="login">

<div class="dropdown">

<img src="th.jpeg" alt="User Photo" class="dropbtn">

<div class="dropdown-content">

<a href="owner\_login.html">Logout</a>

<a href="passwordreset.html">Change Password</a>

</div>

</div>

</div>

</div>

</header>

<br>

<p>POOL & ROLL : SHARE THE RIDE, SAVE THE EARTH!</p> <br><br>

<marquee>

<p>Carpooling is the sharing of car journeys so that more than one person travels in a car, and prevents the

need for others to have to drive to a location themselves. Carpooling is considered a Demand-Responsive

Transport (DRT) service</p>

</marquee>

<div class="column3">

<nav>

<ul>

<div>

<a onclick="login()" href="searchride.html">

<img id="search" src="https://cdn-icons-png.flaticon.com/128/751/751463.png" alt="Icon"

class="nav-icon"><br>

<div id="e">Search Cab</div>

</a>

</div>

<div>

<a onclick="login()" href="parking.html">

<img id="search" src="https://img.icons8.com/?size=100&id=12684&format=png"><br>

<div id="e1">Lift-Link Parking!</div>

</a>

</div>

<div>

<a onclick="login()" href="#">

<img id="search" src="https://img.icons8.com/?size=100&id=44471&format=png"><br>

<div id="e2">Service</div>

</a>

</div>

<div>

<a onclick="login()"

href="https://mail.google.com/mail/u/1/#inbox?compose=DmwnWrRqjCHQxQhvQHKpdWvrqjcDFdxxwgXCjGVkcmqSKBfXrnwgHTXdJrnkGfbVbQxlfhCwrsSv">

<img id="search" src="https://cdn-icons-png.flaticon.com/128/3249/3249904.png"><br>

<div id="e3">Contact us?</div>

</a>

</div>

</ul>

</nav>

</div>

<body onload="showride()">

<form action="/mybookingstatus" method="POST" >

<label>Status:</label>

<select name="status">

<option value="choose">Choose</option>

<option value="pending">Pending</option>

<option value="cancel">Cancel</option>

<option value="Booked">Booked</option>

</select>

<input type="submit" value="search" id="h">

<div class="info">

<div id="v1"></div>

<div id="v2"></div>

<div id="v3"></div>

<div id="v4"></div>

<div id="v5"></div>

<div id="v6"></div>

<div id="v7"></div>

<div id="v8"></div>

<div id="v9"></div>

<div id="v10"></div>

<div id="v11"></div>

<div id="v12"></div>

<div id="v13"></div>

<div id="v14"></div>

<div id="v15"></div>

<div id="v16"></div>

</div>

</form>

</body>

<footer>

<div class="footer-section">

<ul>

<li><a href="https://mail.google.com/">Contact us?</a></li>

<li><a href="https://mail.google.com/">call: 6397918693</a></li>

<li><a href="https://mail.google.com/">Mail:rautelavaibhav01@gmail.com </a></li>

</ul>

</div>

<div class="footer-section">

<ul>

<li><a href="#">Our Service</a></li>

<li><a href="https://mail.google.com/">call: 6397918693</a></li>

<li><a href="https://mail.google.com/">Mail:rautelavaibhav01@gmail.com </a></li>

</ul>

</div>

<div class="footer-section">

<ul>

<li><a href="#">Privacy Policy</a></li>

<li><a href="https://mail.google.com/">call: 6397918693</a></li>

<li><a href="https://mail.google.com/">Mail:rautelavaibhav01@gmail.com </a></li>

</ul>

</div>

<div class="footer-section">

<ul>

<li><a href="#">Terms & Condition</a></li>

<li><a href="https://mail.google.com/">call: 6397918693</a></li>

<li><a href="https://mail.google.com/">Mail:rautelavaibhav01@gmail.com </a></li>

</ul>

</div>

</footer>

<div id="foot">MAJOR PROJECT Copyright © 2023-2024- All rights reserved || Designed By: Vaibhav Rautela </div>

</html>

**GEOLOCATION**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Realtime Location Tracker</title>

<!-- leaflet css -->

<link rel="stylesheet" href="https://unpkg.com/leaflet@1.7.1/dist/leaflet.css" />

<style>

body {

margin: 0;

padding: 0;

}

#map {

width: 100%;

height: 100vh;

}

</style>

</head>

<body onload="initializeMap()">

<div id="map"></div>

<!-- leaflet js -->

<script src="https://unpkg.com/leaflet@1.7.1/dist/leaflet.js"></script>

<script>

function initializeMap() {

let lat = 14.0860746;

let long = 100.608406;

let zoomLevel = 13; // Initial zoom level

if (localStorage.getItem('latitude') && localStorage.getItem('longitude')) {

lat = parseFloat(localStorage.getItem('latitude'));

long = parseFloat(localStorage.getItem('longitude'));

}

var map = L.map('map').setView([lat, long], zoomLevel);

var osm = L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {

attribution: '&copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors'

});

osm.addTo(map);

if (!navigator.geolocation) {

console.log("Your browser doesn't support geolocation feature!");

} else {

navigator.geolocation.watchPosition(getPosition, showError, {

enableHighAccuracy: true,

timeout: 10000,

maximumAge: 0

});

}

var userMarker, geoMarker;

function getPosition(position) {

var lat = position.coords.latitude;

var long = position.coords.longitude;

var accuracy = position.coords.accuracy;

localStorage.setItem('latitude', lat);

localStorage.setItem('longitude', long);

if (userMarker) {

map.removeLayer(userMarker);

}

userMarker = L.marker([lat, long]);

userMarker.bindPopup(`<b>Your location:</b><br>Lat: ${lat}<br>Long: ${long}<br>Accuracy: ${accuracy} meters`).openPopup();

var featureGroup = L.featureGroup([userMarker]).addTo(map);

map.fitBounds(featureGroup.getBounds());

map.setView([lat, long], zoomLevel);

console.log("Your coordinate is: Lat: " + lat + " Long: " + long + " Accuracy: " + accuracy);

}

function showError(error) {

switch (error.code) {

case error.PERMISSION\_DENIED:

console.log("User denied the request for Geolocation.");

break;

case error.POSITION\_UNAVAILABLE:

console.log("Location information is unavailable.");

break;

case error.TIMEOUT:

console.log("The request to get user location timed out.");

break;

case error.UNKNOWN\_ERROR:

console.log("An unknown error occurred.");

break;

}

}

function geo() {

var xhttp;

if (window.XMLHttpRequest) {

xhttp = new XMLHttpRequest();

} else {

xhttp = new ActiveXObject("Microsoft.XMLHTTP");

}

xhttp.onreadystatechange = function () {

if (xhttp.readyState == 4 && xhttp.status == 200) {

const item = xhttp.responseText;

const data = JSON.parse(item);

var name = data[0].name

var lat = data[0].lat;

var long = data[0].lon;

if (geoMarker) {

map.removeLayer(geoMarker);

}

var redIcon = L.icon({

iconUrl: 'locationb.png',

iconSize: [41, 41], // size of the icon

iconAnchor: [12, 41], // point of the icon which will correspond to marker's location

popupAnchor: [1, -34], // point from which the popup should open relative to the iconAnchor

shadowUrl: 'https://cdnjs.cloudflare.com/ajax/libs/leaflet/1.7.1/images/marker-shadow.png',

shadowSize: [41, 41] // size of the shadow

});

geoMarker = L.marker([lat, long], { icon: redIcon });

geoMarker.bindPopup(`<b>${name}</b><br>Lat: ${lat}<br>Long: ${long}`).openPopup();

var featureGroup = L.featureGroup([geoMarker]).addTo(map);

map.fitBounds(featureGroup.getBounds());

console.log("Geo coordinate is: Lat: " + lat + " Long: " + long);

}

};

xhttp.open("GET", "geo.json", true);

xhttp.send();

}

geo();

}

</script>

</body>

</html>