

# VEHICLE SHARING SYSTEM

*Project Report Submitted By*

**JOBIN JOSEPH**

**Reg. No: AJC20MCA-2045**

*In Partial fulfillment for the Award of the Degree Of*

**MASTER OF COMPUTER APPLICATIONS (2 YEAR)  
(MCA)**

**APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY**



**AMAL JYOTHI COLLEGE OF ENGINEERING  
KANJIRAPPALLY**

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with 'A' grade. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

**2020-2022**

**DEPARTMENT OF COMPUTER APPLICATIONS**

**AMAL JYOTHI COLLEGE OF ENGINEERING**

**KANJIRAPPALLY**



**CERTIFICATE**

This is to certify that the Project report, “**VEHICLE SHARING SYSTEM**” is the bonafide work of **JOBIN JOSEPH (Reg.No: AJC20MCA-2045)** in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2021-22.

**Dr. Juby Mathew**

**Internal Guide**

**Ms. Grace Joseph**

**Coordinator**

**Rev. Fr. Dr. Rubin Thottupurathu Jose**

**Head of the Department**

**External Examiner**

## **DECLARATION**

I hereby declare that the project report “**Vehicle Sharing System**” is a bonafided work done at Amal Jyothi College of Engineering, towards the partial fulfilment of the requirements for the award of the Degree of Master of Computer Applications (2Year) from APJ Abdul Kalam Technological University, during the academic year 2021-2022.

**Date: 25/07/2022**

**KANJIRAPPALLY**

**JOBIN JOSEPH**

**Reg. No: AJC20MCA-2045**

## ACKNOWLEDGEMENT

First and foremost, I thank God almighty for his eternal love and protection throughout the project. I take this opportunity to express my gratitude to all who helped me in completing this project successfully. It has been said that gratitude is the memory of the heart. I wish to express my sincere gratitude to our manager **Rev. Fr. Dr. Mathew Paikatt** and Principal **Dr. Lillykutty Jacob** for providing good faculty for guidance.

I owe a great depth of gratitude towards our Head of the Department **Rev. Fr. Dr. Rubin Thottupurathu Jose** for helping us. I extend my whole hearted thanks to the project coordinators **Rev. Fr. Dr. Rubin Thottupurathu Jose** and **Ms. Grace Joseph** for their valuable suggestions and for overwhelming concern and guidance from the beginning to the end of the project. I would also like to express sincere gratitude to my guide, **Dr. Juby Mathew** for his inspiration and helping hand.

I thank our beloved teachers for their cooperation and suggestions that helped me throughout the project. I express my thanks to all my friends and classmates for their interest, dedication, and encouragement shown towards the project. I convey my hearty thanks to my family for the moral support, suggestions, and encouragement to make this venture a success.

JOBIN JOSEPH

## **ABSTRACT**

The aim of this project is to make a **Vehicle Sharing Service** Application. This vehicle Sharing system is a web-based application to provide us with a simple riding platform between the vehicle owner and passengers (Vehicle users). The vehicle owner can post a travel destination prior to the travel date and if anyone is interested to travel in the same location, he/she can contact the vehicle owner and book the ride with an amount specified by the owner. This ride booking and sharing services allow customers to arrange transportation quickly and at a reasonable amount. The vehicle users can search travelling location and can book the ride. This System consists of a website and an android application of user module. This System keeps the data in a centralized way which is available to all the users at the same time. This system is an effort to reduce consumption of fuel, our most important non-renewable resource and traffic congestion on roads by encouraging people to use Vehicle sharing. So, it is an environment-friendly social application and also helps people to reduce their journey time.

# CONTENT

Sl. No	Topic	Page No
<b>1</b>	<b>INTRODUCTION</b>	<b>1</b>
<b>1.1</b>	<b>PROJECT OVERVIEW</b>	<b>2</b>
<b>1.2</b>	<b>PROJECT SPECIFICATION</b>	<b>2</b>
<b>2</b>	<b>SYSTEM STUDY</b>	<b>5</b>
<b>2.1</b>	<b>INTRODUCTION</b>	<b>5</b>
<b>2.2</b>	<b>EXISTING SYSTEM</b>	<b>6</b>
<b>2.3</b>	<b>DRAWBACKS OF EXISTING SYSTEM</b>	<b>7</b>
<b>2.4</b>	<b>PROPOSED SYSTEM</b>	<b>7</b>
<b>2.5</b>	<b>ADVANTAGES OF PROPOSED SYSTEM</b>	<b>7</b>
<b>3</b>	<b>REQUIREMENT ANALYSIS</b>	<b>9</b>
<b>3.1</b>	<b>FEASIBILITY STUDY</b>	<b>10</b>
<b>3.1.1</b>	<b>ECONOMICAL FEASIBILITY</b>	<b>10</b>
<b>3.1.2</b>	<b>TECHNICAL FEASIBILITY</b>	<b>10</b>
<b>3.1.3</b>	<b>BEHAVIORAL FEASIBILITY</b>	<b>10</b>
<b>3.2</b>	<b>SYSTEM SPECIFICATION</b>	<b>11</b>
<b>3.2.1</b>	<b>HARDWARE SPECIFICATION</b>	<b>11</b>
<b>3.2.2</b>	<b>SOFTWARE SPECIFICATION</b>	<b>11</b>
<b>3.3</b>	<b>SOFTWARE DESCRIPTION</b>	<b>11</b>
<b>3.3.1</b>	<b>PHP</b>	<b>11</b>
<b>3.3.2</b>	<b>MYSQL</b>	<b>12</b>
<b>3.3.3</b>	<b>ANDROID STUDIO</b>	<b>13</b>
<b>3.3.4</b>	<b>JAVA</b>	<b>14</b>
<b>4</b>	<b>SYSTEM DESIGN</b>	<b>15</b>
<b>4.1</b>	<b>INTRODUCTION</b>	<b>16</b>
<b>4.2</b>	<b>UML DIAGRAM</b>	<b>16</b>
<b>4.2.1</b>	<b>USE CASE DIAGRAM</b>	<b>17</b>
<b>4.2.2</b>	<b>ACTIVITY DIAGRAM</b>	<b>20</b>
<b>4.2.3</b>	<b>CLASS DIAGRAM</b>	<b>21</b>
<b>4.2.4</b>	<b>OBJECT DIAGRAM</b>	<b>22</b>
<b>4.2.5</b>	<b>STATECHART DIAGRAM</b>	<b>23</b>
<b>4.2.6</b>	<b>SEQUENCE DIAGRAM</b>	<b>24</b>

<b>4.2.7</b>	<b>COMPONENT DIAGRAM</b>	<b>26</b>
<b>4.2.8</b>	<b>DEPLOYMENT DIAGRAM</b>	<b>27</b>
<b>4.3</b>	<b>USER INTERFACE DESIGN</b>	<b>28</b>
<b>4.4</b>	<b>DATA BASE DESIGN</b>	<b>33</b>
<b>5</b>	<b>SYSTEM TESTING</b>	<b>39</b>
<b>5.1</b>	<b>INTRODUCTION</b>	<b>40</b>
<b>5.2</b>	<b>TEST PLAN</b>	<b>41</b>
<b>5.2.1</b>	<b>UNIT TESTING</b>	<b>42</b>
<b>5.2.2</b>	<b>INTEGRATION TESTING</b>	<b>42</b>
<b>5.2.3</b>	<b>VALIDATION TESTING</b>	<b>42</b>
<b>5.2.4</b>	<b>USER ACCEPTANCE TESTING</b>	<b>43</b>
<b>6</b>	<b>IMPLEMENTATION</b>	<b>50</b>
<b>6.1</b>	<b>INTRODUCTION</b>	<b>51</b>
<b>6.2</b>	<b>IMPLEMENTATION PROCEDURE</b>	<b>51</b>
<b>6.2.1</b>	<b>USER TRAINING</b>	<b>52</b>
<b>6.2.2</b>	<b>TRAINING ON APPLICATION SOFTWARE</b>	<b>52</b>
<b>6.2.3</b>	<b>SYSTEM MAINTENANCE</b>	<b>53</b>
<b>7</b>	<b>CONCLUSION &amp; FUTURE SCOPE</b>	<b>57</b>
<b>7.1</b>	<b>CONCLUSION</b>	<b>58</b>
<b>7.2</b>	<b>FUTURE SCOPE</b>	<b>58</b>
<b>8</b>	<b>BIBLIOGRAPHY</b>	<b>59</b>
<b>9</b>	<b>APPENDIX</b>	<b>62</b>
<b>9.1</b>	<b>SAMPLE CODE</b>	<b>63</b>
<b>9.2</b>	<b>SCREEN SHOTS</b>	<b>75</b>
<b>9.3</b>	<b>PLAGIARISM REPORT</b>	<b>84</b>

## List of Abbreviation

IDE	-	Integrated Development Environment
HTML	-	Hyper Text Markup Language.
CSS	-	Cascading Style Sheet
SQL	-	Structured Query Language
UML	-	Unified Modeling Language



## **CHAPTER 1**

### **INTRODUCTION**

## 1.1 PROJECT OVERVIEW

The project Vehicle sharing system includes registration of vehicle owners and passengers. The vehicle owners can add their vehicle details and travel details. The vehicle owner can post a travel destination prior to the travel date and if anyone is interested to travel in the same location, he/she can contact the vehicle owner and book the ride with an amount specified by the owner.

The Vehicle sharing system can be accessed using a username and password. It is accessible either by a vehicle owners or users. The data can be retrieved easily. The interface is very user-friendly.

## 1.2 PROJECT SPECIFICATION

The purpose of the project entitled as “Vehicle Sharing System” connects vehicle owners to people in their vicinity who needs a vehicle for transportation. The ride fare is fixed by the vehicle owner within the range of kilometers

The system includes 4 modules. They are:

### 1.Admin:

- **Login:** Admin need to login.
- **Manage Vehicle owners:** Admin can view and manage vehicle owners.
- **Manage Vehicle Users:** Admin can view and manage vehicle users (passengers).
- **Feedback and complaints:** Admin can able to manage feedbacks and complaints of vehicle users and vehicle owners.
- **Payment History:** Admin can able to view all the payment details.

Admin can also recover his/her own password.

## 2. Vehicle Owner:

- **Login/Register:** Vehicle owner can login using their id or can register as a new owner.
- **Dashboard:** Vehicle owner can view his/her profile, Edit profile, Travel history.
- **Add Vehicle:** Vehicle owner can able to add or delete their vehicle details.
- **Add travel details:** Vehicle owner can post their travel details.
- **Approve or reject:** Owners can able to approve or reject travel request.
- **Track:** Owners can view and track there travel requests and bookings.
- **feedbacks:** View feedbacks from users.
- **History:** View old booking and details.

Vehicle owners can update his/her profile, recover the password.

## 3. Vehicle Users:

- **Login:** Vehicle Users can login using their id or can register as a new owner.
- **Dashboard:** Vehicle owner can view his/her profile, Edit profile, Travel history.
- **Search:** users can search the travel destination and can able to request the ride.
- **Request Ride:** passengers can search and find the vehicle and also they can request the ride.
- **payment:** user can pay the amount after the approval of travel request. Payment can be done using online/COD.
- **Track:** View and manage travel requests.
- **Bookings:** View and manage the completed travel history.
- **Feedbacks:** Vehicle users can able to give feedbacks and ratings after the travel.

Vehicle users can update his profile, recover the password.

**4. Vehicle user android app:**

User module is developed as an android application. Vehicle users can able to register and login through this application. With this application the user can search the travel location and they can request the ride. The user can also view the status of their bookings.

## **CHAPTER 2**

### **SYSTEM STUDY**

## 2.1 INTRODUCTION

System analysis is the process of acquiring and analyzing data, diagnosing issues, and using the data to suggest system changes. The system users and system developers must communicate extensively during this problem-solving process. Any system development process should start with a system analysis or research. The system is meticulously examined and assessed. The system analyst assumes the role of an interrogator and delves deeply into how the current system functions. The input to the system is identified, and the system is seen as a whole. The various procedures can be linked to the outputs from the organizations. Understanding the issue, identifying the pertinent and important variables, evaluating and synthesizing the many elements, and selecting the best or, at the very least, most acceptable course of action are all part of system analysis.

The process must be thoroughly studied using a variety of methodologies, including questionnaires and interviews. To reach a conclusion, the information gathered by these sources must be carefully examined. Understanding how the system works is the conclusion. The current system is the name of this system. Now, the current system is carefully examined, and issue areas are found. The designer now acts as a problem-solver and works to resolve the issues the business is having. Proposals are made in place of the solutions. The proposal is then analytically compared to the current system, and the best one is chosen. The user is given the opportunity to approve or reject the suggestion. On user request, the proposal is assessed and appropriate revisions are made. As soon as the user is content with the suggestion, this loop breaks.

The process of acquiring and analyzing data in order to use it for future system studies is known as preliminary study. Initial research is a problem-solving activity that necessitates close coordination between system users and developers. It conducts a number of feasibility studies. These investigations provide an approximate estimate of the system activities, which can be used to determine the tactics to be used for an efficient system research and analysis.

## 2.2 EXISTING SYSTEM

Existing system is specific to particular vehicles like cars or bikes. But in this system any private vehicles can be added.

## 2.3 DRAWBACKS OF EXISTING SYSTEM

- Only cars and bike sharing is possible.
- Less secure and take lot of time.
- Less user friendly.

## 2.4 PROPOSED SYSTEM

The proposed system is user friendly. Good communication is maintained between vehicle owners and driver. All the user's requests can be viewed by the vehicle owners immediately. Details of the driver and vehicle are maintained in the database. High level security is assigned in the proposed system. These services are to be provided in an efficient, cost-effective manner, with the goal of reducing the time and resources currently required for such tasks. The ride fare is fixed by the vehicle owner within the range of kilometers travelled and will even take care of all administrative issues.

## 2.5 ADVANTAGES OF PROPOSED SYSTEM

The system is very simple in design and to implement. The system requires very low system resources and the system will work in almost all configurations. It has got following features:

### ➤ **Eliminate the chance for critical data misuse: -**

Only Admin have access to the vehicle owners and users' data, no other members are permitted. Therefore, the data will be kept private.

---

➤ **Ensure data accuracy: -**

The suggested solution does away with human mistakes made when entering user information during registration.

➤ **Better service: -**

Hard copy storage won't be a strain for the system. For performing the same activity, we can also save time and resources. The data can be kept for a longer time without losing any information.



## **CHAPTER 3**

### **REQUIREMENT ANALYSIS**

### **3.1 FEASIBILITY STUDY**

In this stage, the project's viability is evaluated, and a business proposal is presented with a very generic project plan and some cost projections. The proposed system's practicality must be investigated during system analysis. This will guarantee that the suggested solution won't burden the business. Understanding the main system requirements is crucial for the feasibility analysis.

Three key considerations involved in the feasibility analysis are:

#### **3.1.1 Economic Feasibility**

This study is being done to see what kind of financial impact the organization's system will have. The corporation has a finite amount of money to invest in the system's research and development. The costs must be supported by evidence. As a result, the developed system came in under budget, which was made possible by the fact that most of the technology were public domain. All that must be purchased are the customized goods.

#### **3.1.2 Technical Feasibility**

This study is being done to evaluate the system's technological requirements, or its technical viability. Any system that is created must not place a heavy burden on the available technical resources. As a result, the client will face high expectations. The created system must have a low bar for implementation, requiring either no changes at all or very few.

#### **3.1.3 Behavioral Feasibility**

The goal of the study is to determine how much the user accepts the system. This includes the instruction needed for the user to operate the system effectively. The system shouldn't make the user feel threatened; instead, they should view it as a need. The techniques used to inform and acquaint the user with the system are the only factors that affect the level of acceptance by the users. As the system's ultimate user, his confidence must be increased so that he may offer some helpful criticism, which is encouraged.

## 3.2 SYSTEM SPECIFICATION

### 3.2.1 Hardware Specification

Processor	- Intel core i3
RAM	- 4 GB
Hard disk	- 1 TB

### 3.2.2 Software Specification

Front End	- HTML, CSS, BOOTSTRAP, JQUERY, Android Studio IDE
Backend	- MYSQL
Client on PC	- Windows 7 and above, Android 5 lollipop and above
Technologies used	- JS, HTML5, AJAX, PHP, CSS, JAVA

## 3.3 SOFTWARE DESCRIPTION

### 3.3.1 PHP

PHP is a server-side scripting language used for general-purpose programming as well as web development. More than 244 million websites and 2.1 million web servers currently use PHP. The PHP group now produces the reference implementation of PHP, which was first developed by Rasmus Ledorf in 1995. PHP, a recursive acronym that once meant for personal home page, now stands for PHP: Hypertext Preprocessor. A web server's PHP processor module interprets PHP code to produce the final web page. In order to handle data, PHP commands can be directly inserted into an HTML source file rather than calling an external file. Due to limitations on the use of the name PHP, it has also developed to include a command-line interface capability and can be used standalone, which makes it incompatible with the GNU General Public License (GPL). Most web servers support the free deployment of PHP, which is also available as a standalone shell on practically all platforms and operating systems.

### 3.3.2 MySQL

Oracle Corporation created, distributed, and provided support for MySQL, the most well-known Open-Source SQL database management system. The most recent details regarding MySQL software are available on the MySQL website.

- **MySQL is a database management system.**

A systematic collection of data is called a database. It might be anything, such as a straightforward grocery list, a photo gallery, or the enormous amount of data in a business network. A database management system like MySQL Server is required in order to add, access, and process data that is stored in a computer database. Database management systems, whether used as stand-alone programmes or as a component of other applications, are essential to computing because computers are excellent at processing vast volumes of data.

- **MySQL databases are relational.**

Instead of placing all the data in one huge warehouse, a relational database keeps the data in individual tables. Physical files that are designed for speed contain the database structures. The logical model provides a flexible programming environment with objects like databases, tables, views, rows, and columns. One-to-one, one-to-many, unique, compulsory or optional, and "pointers" between distinct tables are a few examples of the rules you might build up to regulate the relationships between various data fields. With a well-designed database, your application won't ever encounter inconsistent, duplicate, orphan, out-of-date, or missing data since the database enforces these rules. MySQL stands for "Structured Query Language" with the SQL prefix. The most popular standard language for accessing databases is SQL. Depending on your programming environment, you might explicitly enter SQL (for example, to generate reports), incorporate SQL statements into other languages' code, or use a language-specific API that obscures the SQL syntax. By way of the ANSI/ISO SQL Standard, SQL is defined. Since its inception in 1986, the SQL standard has gone through various iterations. In this document, "SQL92" refers to the 1992 standard, "SQL: 1999" to the 1999 standard, and "SQL: 2003" to the most recent version of the standard. The SQL Standard as it exists at any one time is referred to as "the SQL standard."

- **MySQL software is Open Source.**

Anyone can use and modify the software because it is open source. The MySQL software is available for free download and usage online by anyone. You are free to examine the source code and modify it as necessary. The GPL (GNU General Public License) is used by the MySQL software to specify what you are allowed to do and are not allowed to do with the software in certain circumstances. You can purchase a commercially licenced version from us if the GPL makes you uncomfortable or if you need to integrate MySQL code into a for-profit application. For further details, see the MySQL Licensing Overview.

- **The MySQL Database Server is very fast, reliable, scalable, and easy to use.**

You ought to give it a shot if that is what you're after. In addition to your other apps, web servers, and other software, MySQL Server can function smoothly on a desktop or laptop while requiring little to no maintenance. You can modify the settings to utilise all the RAM, CPU power, and I/O capacity if you dedicate an entire machine to MySQL.

- **MySQL Server works in client/server or embedded systems.**

The MySQL Database Software is a client/server system that includes a multi-threaded SQL server that supports several client programmes and libraries, administration tools, and a broad variety of application programming interfaces (APIs). Additionally, we provide MySQL Server as an integrated multi-threaded library that you can link into your programme to create a standalone offering that is smaller, faster, and simpler to operate.

### 3.3.3 Android Studio

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems. It is a replacement for the Eclipse Android Development Tools (ADT) as the primary IDE for native Android application development. Android Studio was announced on May 16, 2013 at the Google I/O conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from

version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0.

### 3.3.4 Java

Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended to let programmers *write once, run anywhere*, meaning that compiled Java code can run on all platforms that support Java without the need to recompile. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The syntax of Java is similar to C and C++, but has fewer low-level facilities than either of them. The Java runtime provides dynamic capabilities (such as reflection and runtime code modification) that are typically not available in traditional compiled languages. As of 2019, Java was one of the most popular programming languages in use according to GitHub, particularly for client–server web applications, with a reported 9 million developers.

Java was originally developed by James Gosling at Sun Microsystems and released in May 1995 as a core component of Sun Microsystems' Java platform. The original and reference implementation Java compilers, virtual machines, and class libraries were originally released by Sun under proprietary licenses. As of May 2007, in compliance with the specifications of the Java Community Process, Sun had relicensed most of its Java technologies under the GPL-2.0-only license. Oracle offers its own Hotspot Java Virtual Machine, however the official reference implementation is the OpenJDK JVM which is free open-source software and used by most developers and is the default JVM for almost all Linux distributions.

## **CHAPTER 4**

### **SYSTEM DESIGN**

## 4.1 INTRODUCTION

Any engineered system or product's development process begins with design. A creative process is design. The secret to an efficient system is a decent design. The process of using different methodologies and concepts to specify a process or a system in enough detail to allow for its physical implementation is referred to as "design." One way to describe it is as the process of using different methodologies and concepts to specify a device, a process, or a system in enough detail to allow for its physical reality. Regardless of the development paradigm that is employed, software design forms the technical core of the software engineering process. The architectural detail needed to construct a system or product is developed through the system design. This programme has also through the best possible design phase, fine tuning all efficiency, performance, and accuracy levels, as in the case of any systematic technique. A user-oriented document is converted into a document for programmers or database staff throughout the design phase. The two stages of system design development are logical design and physical design.

## 4.2 UML DIAGRAM

A common language known as UML is used to specify, visualize, build, and document the software system artefacts. The Object Management Group (OMG) was responsible for developing UML, and a draught of the UML 1.0 definition was presented to the OMG in January 1997.

Unified Modeling Language is known as UML. Compared to other popular programming languages like C++, Java, COBOL, etc., UML is unique. A visual language called UML is used to create software blueprints. A general-purpose visual modelling language for software system visualization, specification, construction, and documentation is what UML is known as. UML is not just used to represent software systems, despite the fact that this is its most common application. It is also used to model systems that are not software-based. For instance, the manufacturing facility's process flow, etc. Although UML is not a programming language, tools can be used to generate code using UML diagrams in a variety of languages. The analysis and design of objects-oriented systems are directly related to UML.



UML has been standardized to the point where it is now an OMG standard. A comprehensive UML diagram that depicts a system is made up of all the elements and relationships. The most crucial aspect of the entire procedure is the UML diagram's aesthetic impact. It is completed by using all the additional components. The following nine diagrams are part of UML.

- Class diagram
- Object diagram
- Use case diagram
- Sequence diagram
- Activity diagram
- State chart diagram
- Deployment diagram
- Component diagram

#### **4.2.1 USE CASE DIAGRAM**

A use case diagram is a visual representation of the interactions between system components. A approach for identifying, outlining, and organizing system requirements is called a use case. The word "system" here refers to a thing that is being created or run, like a website for mail-order product sales and services. UML (Unified Modeling Language), a standard language for the modelling of real-world objects and systems, uses use case diagrams.

The planning of overall requirements, validating a hardware design, testing and debugging a software product in development, producing an online help reference, or carrying out a task focused on customer support are just a few examples of system objectives. For instance, use cases in a product sales context can involve customer service, item ordering, catalogue updating, and payment processing. There are four elements in a use case diagram.

- The actors, often people involved in the system who are defined according to their roles; the boundary, which establishes the system of interest in relation to its environment.

- 
- The relationships between and among the actors and the use cases, which are the precise roles played by the players within and around the system.

Use case diagrams are created to depict a system's functional requirements. To create an effective use case diagram after identifying the aforementioned things, we must adhere to the following rules.

- A use case's naming is very significant. The name should be selected in a way that makes it clear what functions are being performed.
- Give the actors names that fit them.
- Clearly depict links and dependencies in the diagram.
- Keep in mind that the diagram's primary function is to indicate the needs; do not attempt to include all possible relationships.

Fig 1 : Use case diagram for vehicle sharing

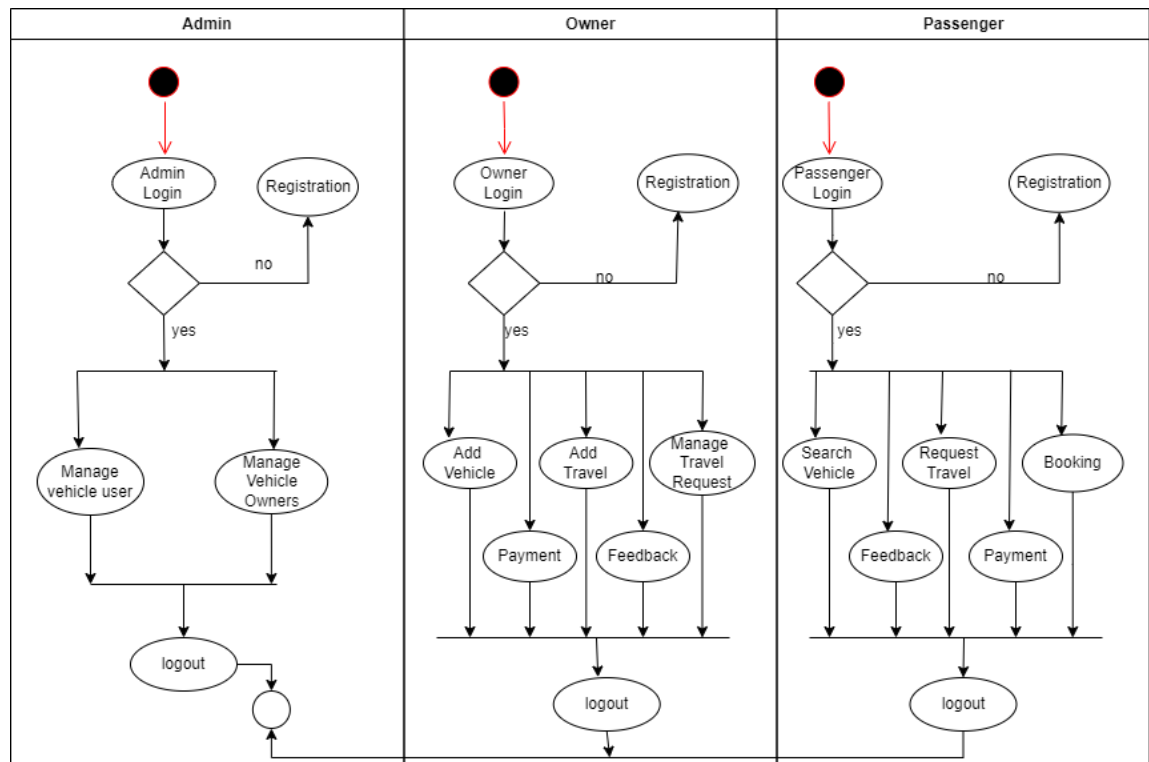


## 4.2.2 ACTIVITY DIAGRAM

An activity diagram is essentially a flowchart that shows how one activity leads to another. The action might be referred to as a system operation.

One operation leads to the next in the control flow. This flow may be parallel, contemporaneous, or branched. Activity diagrams use many features, such as fork, join, etc., to cope with all types of flow control.

Fig 2 : Activity diagram for Vehicle sharing System.

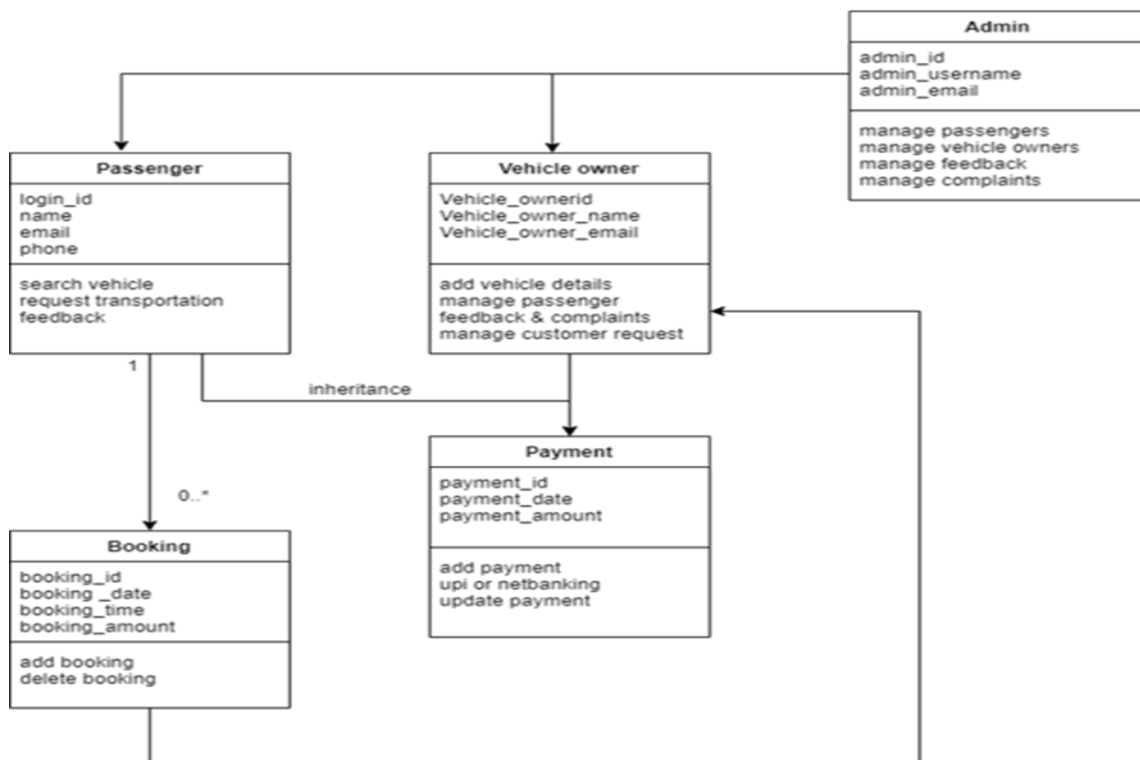


### 4.2.3 CLASS DIAGRAM

Static diagrams include class diagrams. It represents the application's static view. Class diagrams are used to create executable code for software applications as well as for visualising, explaining, and documenting various elements of systems.

The characteristics and functions of a class are described in a class diagram, along with the restrictions placed on the system. Because they are the only UML diagrams that can be directly translated with object-oriented languages, class diagrams are extensively utilised in the modelling of object-oriented systems.

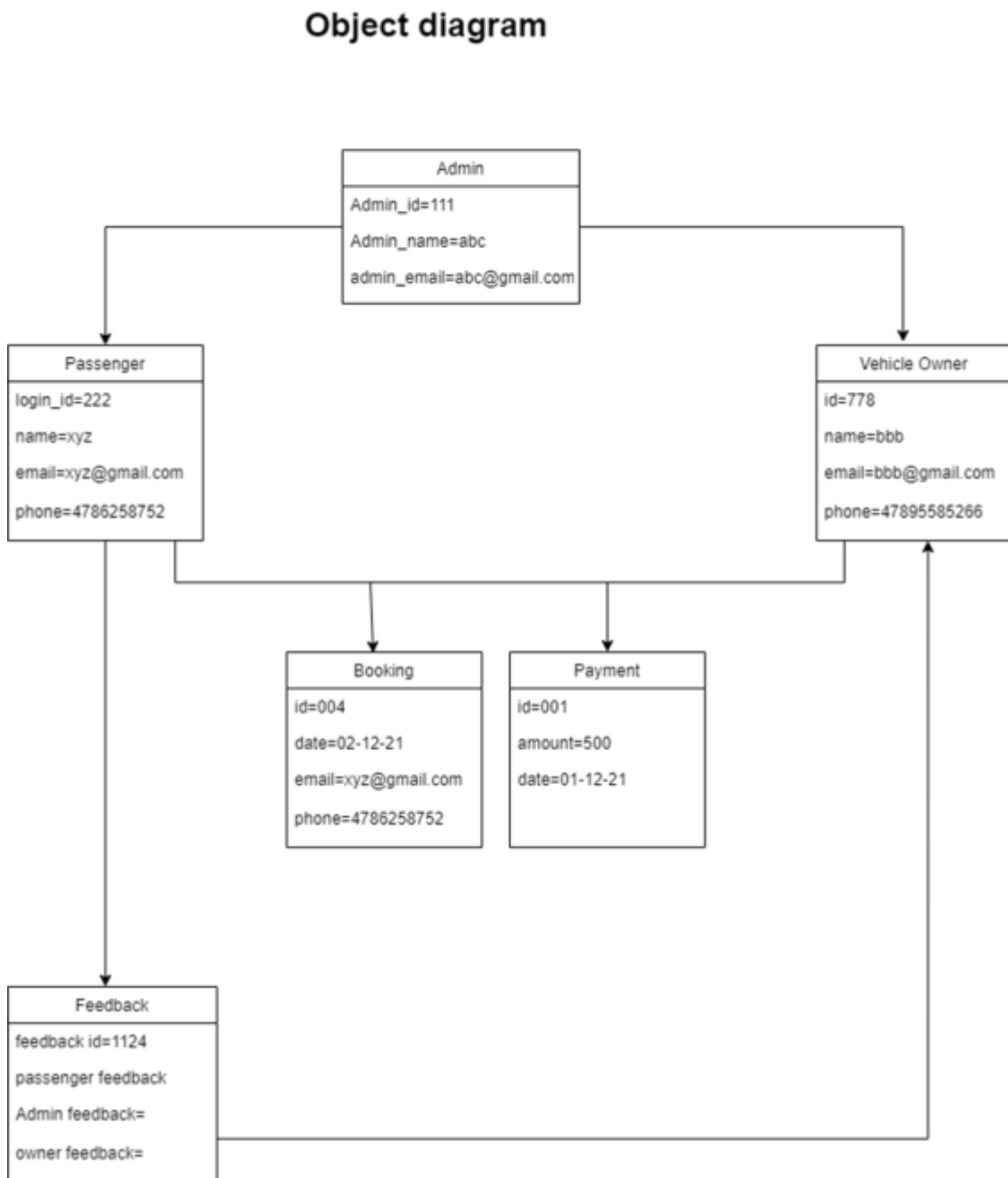
Fig 3: Class diagram for vehicle Sharing



#### 4.2.4 OBJECT DIAGRAM

An instance of a class diagram is represented by an object diagram. Class and object diagrams both use the same fundamental ideas. The static view of a system is also represented by object diagrams, but this static view represents a momentary snapshot of the system.

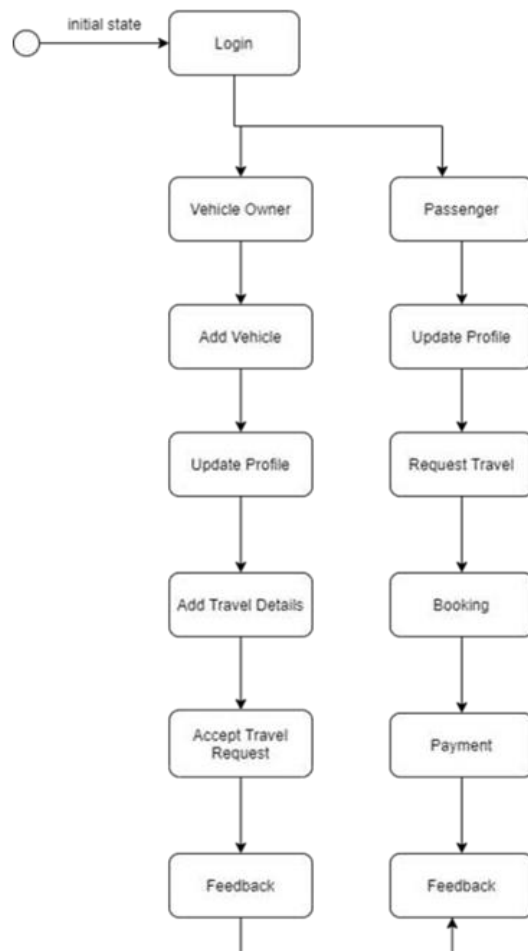
Fig 4: Object diagram for Vehicle Sharing System.



### 4.2.5 STATE CHART DIAGRAM

One of the five UML diagrams used to depict a system's dynamic nature is the state chart diagram. Throughout an object's existence, they define several states, and these states are altered by events. The reactive systems can be modelled with state chart diagrams. A system that reacts to internal or external events is known as a reactive system.

Fig 5 : State chart diagram for vehicle sharing system



## 4.2.6 SEQUENCE DIAGRAM

A sequence diagram essentially shows how things interact with one another sequentially, or the order in which these interactions occur. A sequence diagram can also be referred to as event diagrams or event scenarios. Sequence diagrams show the actions taken by the components of a system in chronological order. Businesspeople and software engineers frequently use these diagrams to record and comprehend the requirements for new and current systems.

### Sequence Diagram Notations –

- i. **Actors** – In a UML diagram, an actor represents a particular kind of role in which it communicates with the system's objects. An actor is always beyond the purview of the system that we want to use the UML diagram to represent. We employ actors to portray a variety of roles, including those of human users and other outside subjects. In a UML diagram, an actor is represented using a stick person notation. In a sequence diagram, there might be several actors.
- ii. **Lifelines** – A named piece that shows a specific participant in a sequence diagram is called a lifeline. In essence, a lifeline represents each incident in a sequence diagram. The top of a sequence diagram is where the lifeline pieces are placed.
- iii. **Messages** – Using messages, communication between objects is demonstrated. The messages are displayed on the lifeline in chronological sequence. Arrows are how messages are represented. A sequence diagram's main components are lifelines and messages.

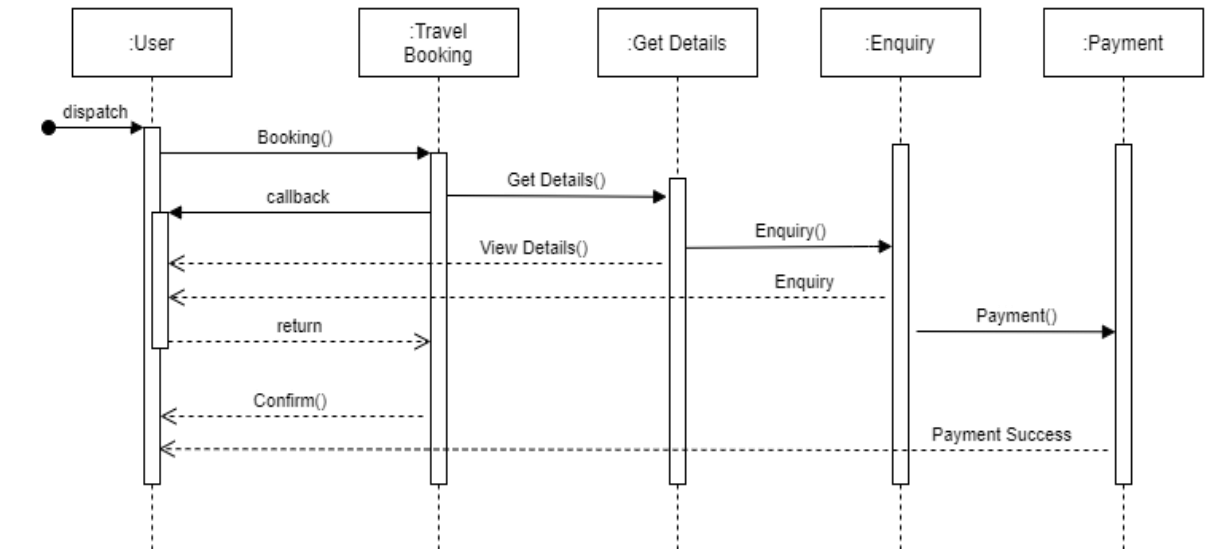
Messages can be broadly classified into the following categories:

- Synchronous messages
- Asynchronous Messages
- Create message
- Delete Message
- Self-Message
- Reply Message
- Found Message



- iv. **Guards** – In the UML, we utilise guards to model circumstances. When we need to limit the flow of messages under the guise of a condition being met, we use them. Software engineers rely on guards to inform them of the limitations imposed by a system or specific process.

### Sequence Diagram



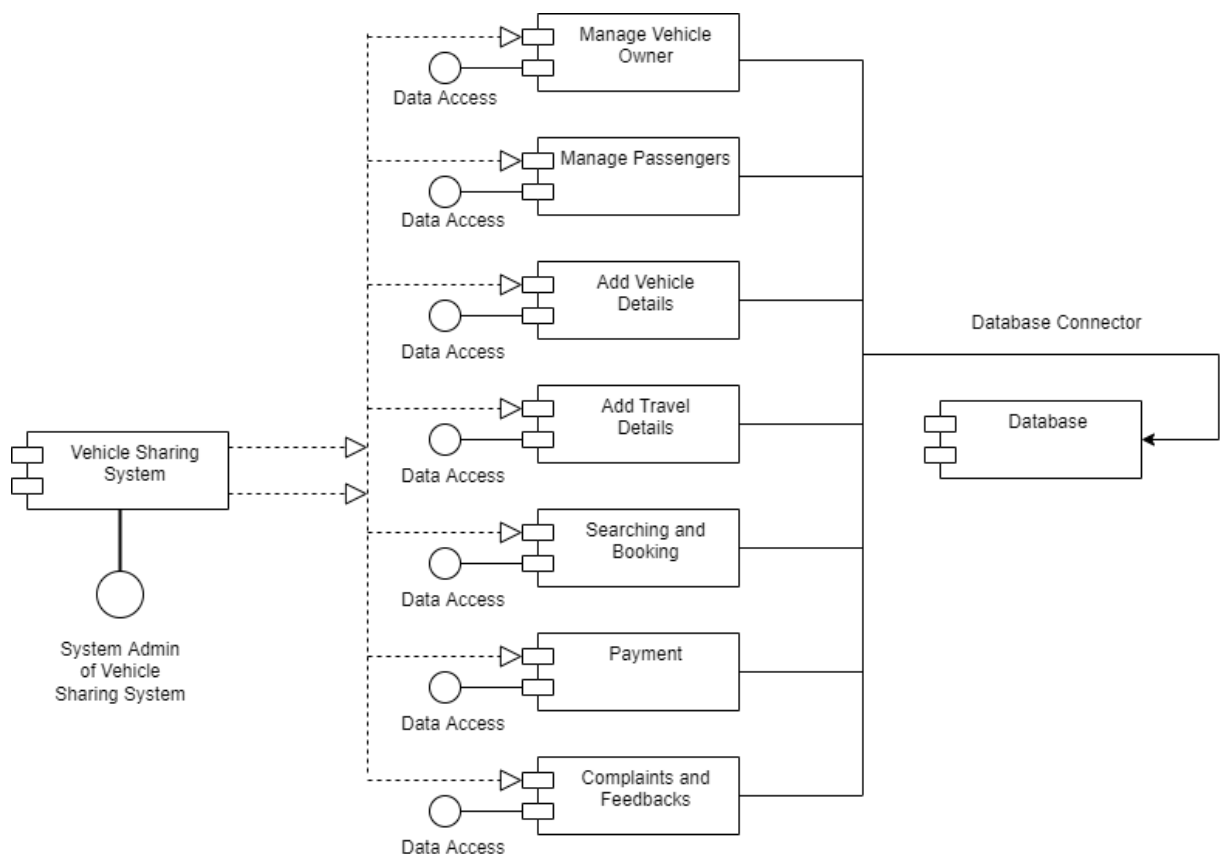
### Uses of sequence diagrams –

- They are also used to display specifics of UML use case diagrams, as well as to describe and depict the logic underlying complex functions, operations, or procedures.
- Used to comprehend the precise operation of present or upcoming systems.
- Visualize the flow of information between different system elements or objects.

Fig 6 : Sequence diagram for Vehicle sharing system

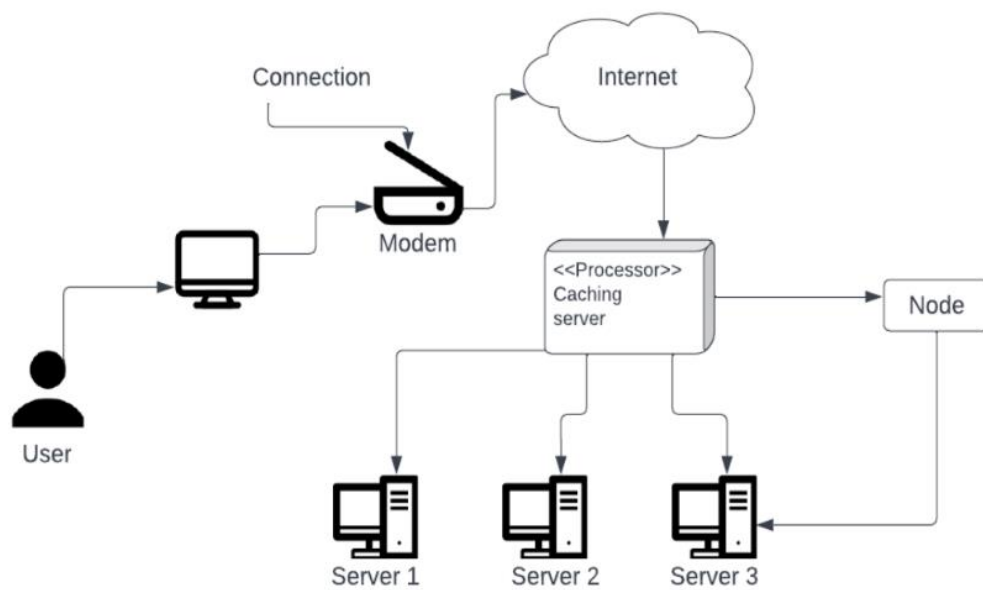
### 4.2.7. COMPONENT DIAGRAM

A component diagram, often called a UML component diagram, shows how the physical parts of a system are wired up and organised. Component diagrams are frequently used to model implementation details and confirm that all necessary system functions have been accounted for.



#### 4.2.8. DEPLOYMENT DIAGRAM

An execution architecture of a system, containing nodes like hardware or software execution environments, and the middleware linking them, is shown in a deployment diagram, a form of UML diagram. Typically, deployment diagrams are used to represent the actual hardware and software of a system.



### 4.3 USER INTERFACE DESIGN

#### 4.3.1-INPUT DESIGN

Form Name : Registration



The registration form is titled "Registration" and is set against a solid red background. It contains the following fields and controls:

- name**: A text input field.
- Email**: A text input field.
- Phone no**: A text input field.
- Gender**: Three radio button options labeled "male", "female", and "Other".
- DOB**: A text input field.
- profile pic**: A text input field followed by a circular profile picture placeholder.
- Government Id**: A text input field.
- User type**: A text input field.
- Sign up**: A green rounded rectangular button.

Form Name : login form



The login form is titled "Login" and is set against a solid red background. It contains the following fields and controls:

- Username**: A text input field.
- Password**: A text input field.
- Login**: A light gray rounded rectangular button.

Form Name: User Home page

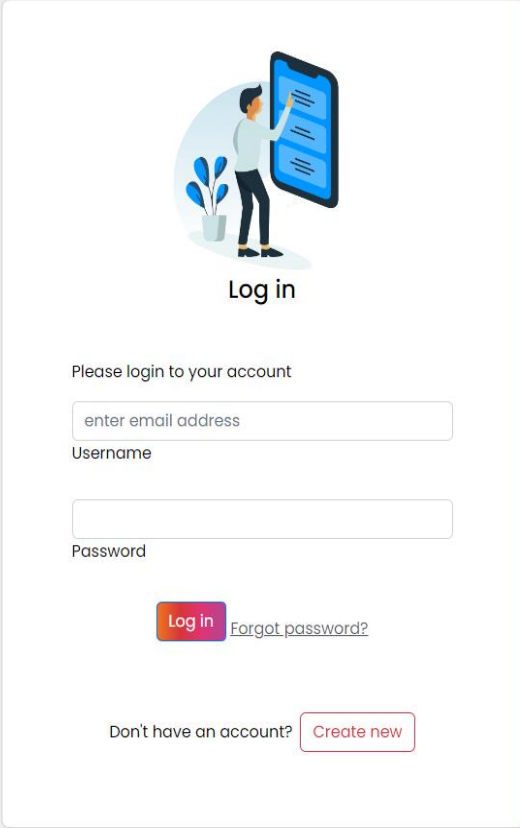


Form Name: Add travel details

The image displays a web application interface for adding travel details. At the top, there is a dark red navigation bar with the following menu items: Home, Post Travel, Add Vehicle, Account, contact, About, and Logout. Below the navigation bar, the main content area has a dark blue background with a faint illustration of a blue car with a red roof rack. Overlaid on this background is a form titled "Add Travel Details". The form contains the following fields: Date, Time, From, To, Amount, Vehicle, and No of seats. Each field is represented by a white text label followed by a white input box. At the bottom right of the form is a green button with the text "ADD" in white capital letters.

### 4.3.2 OUTPUT DESIGN

#### Login page



**Log in**

Please login to your account

enter email address

Username

Password

[Log in](#) [Forgot password?](#)

Don't have an account? [Create new](#)

**We are more than just a company**


The vehicle Sharing system is a web-based application to provide us with a simple riding platform between the vehicle owner and a user. The vehicle owner can post a travel destination prior to the travel date and if anyone is interested to travel in the same location, he/she can contact the vehicle owner and book the ride with an amount specified by the owner. Personal ride booking and sharing services allow customers to arrange transportation quickly and at a reasonable amount. This application typically matches a customers location with the nearest available vehicle. This project consists of a website and its mobile application. This website keeps the data in a centralized way which is available to all the users at the same time. This system is an effort to reduce consumption of fuel, our most important non-renewable resource and traffic congestion on roads by encouraging people to use Vehicle sharing. So it is an environment-friendly social application and also helps people to reduce their journey time.

## Registration page

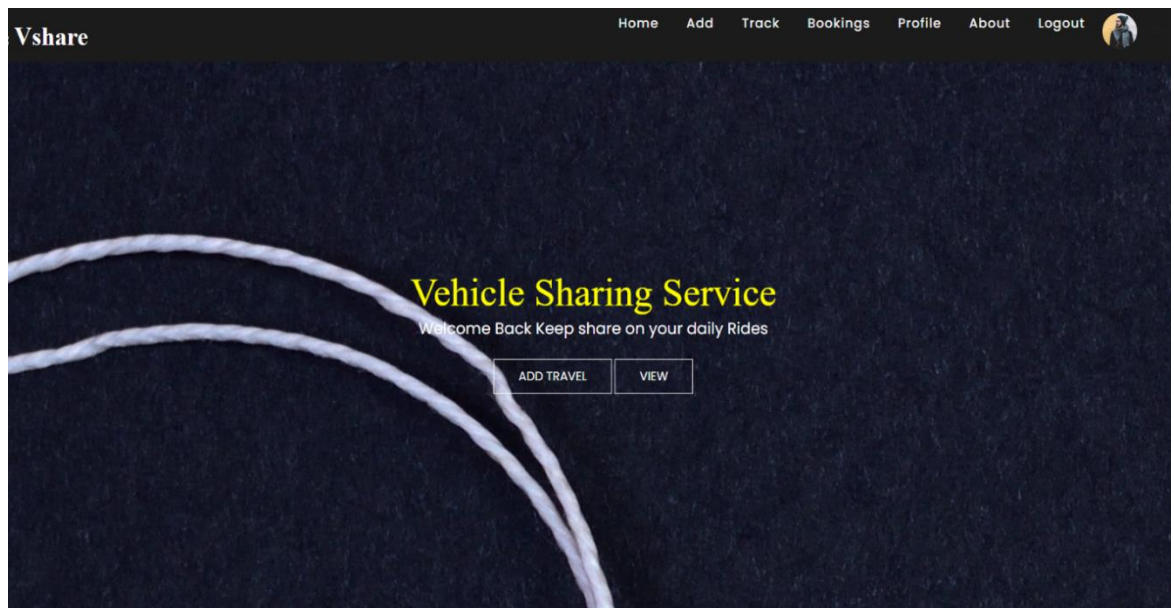
**Vshare** Home ABOUT

### Sign Up

Full Name*	User Name*	User Type*
<input type="text"/>	<input type="text"/>	Vehicle Owner ▾
Email address*	phone*	Date Of Birth*
<input type="text"/>	<input type="text"/>	mm/dd/yyyy 📅
Place*	State*	Country*
<input type="text"/>	Kerala ▾	India ▾
District*	Gender*	Aadhaar No*
<input type="text"/>	MALE ▾	<input type="text"/>
Licence Id*	Password*	Re Enter Password*
<input type="text"/>	<input type="text"/>	<input type="text"/>

 No file chosen

## Home page



## Add Travel details

The screenshot shows the 'Add Travel Details' form within the 'Vshare' application. The form is titled 'Add Travel Details' and is located below the navigation bar. It contains several input fields and a button. The fields are: 'From\*' (text input), 'To\*' (text input), 'Route map\*' (text input with a placeholder 'You can enter places between From and To with each separated by Space'), 'Date\*' (text input with a placeholder 'mm/dd/yyyy' and a calendar icon), 'Time\*' (text input with a placeholder '--:--:--' and a clock icon), 'No of Seats Available\*' (text input), 'Rate /seat\*' (text input), 'Luggage space Available\*' (dropdown menu with 'YES' selected), and 'Select\_Vehicle\*' (dropdown menu with '-- Select an option --' selected). A blue 'Add Travel' button is positioned at the bottom center of the form.



## 4.4 DATABASE DESIGN

A database is a structured system with the capacity to store information and allows users to retrieve stored information quickly and effectively. Any database's primary goal is its data, which demands protection.

There are two stages to the database design process. The user needs are obtained in the first step, and a database is created to as clearly as possible meet these criteria. This process, known as information level design, is carried out independently of all DBMSs.

The design for the specific DBMS that will be used to construct the system in issue is converted from an information level design to a design in the second stage. Physical Level Design is the stage where the characteristics of the particular DBMS that will be used are discussed. Parallel to the system design is a database design. The database's data arrangement aims to accomplish the two main goals listed below.

- Data Integrity
- Data independence

### 4.4.1 Relational Database Management System (RDBMS)

In a relational model, the database is shown as a set of relations. Each relation resembles a file or table of records with values. A row is referred to as a tuple, a column heading is referred to as an attribute, and the table is referred to as a relation in formal relational model language. A relational database is made up of a number of tables, each with its own name. In a story, each row represents a group of associated values.

#### Relations, Domains & Attributes

A relation is a table. Tuples are the units of a table's rows. An ordered group of  $n$  elements is a tuple. Attributes are referred to as columns. Every table in the database has relationships already established between them. This guarantees the integrity of both referential and entity relationships. A group of atomic values make up a domain  $D$ . Choosing a data type from which the domain's data values are derived is a typical way to define a domain. To make it easier to understand the values of the domain, it is also helpful to give it a name. Each value in a relation is atomic and cannot be broken down.

### Relationships

- Key is used to create table relationships. Primary Key and Foreign Key are the two principal keys that are most crucial. With the use of these keys, relationships for entity integrity and referential integrity can be created.
- Entity Integrity forbids the use of null values for any Primary Key.
- No Primary Key may contain null values, according to Referential Integrity.
- Referential Integrity: A Primary Key value in the same domain must correspond to each unique Foreign Key value. Super Key and Candidate Keys are additional keys.

#### 4.4.2 Normalization

The simplest possible grouping of data is used to put them together so that future changes can be made with little influence on the data structures. The formal process of normalizing data structures in a way that reduces duplication and fosters integrity. Using the normalization technique, superfluous fields are removed and a huge table is divided into several smaller ones. Anomalies in insertion, deletion, and updating are also prevented by using it. Keys and relationships are two notions used in the standard form of data modelling. A row in a table is uniquely identified by a key. Primary keys and foreign keys are the two different kinds of keys. A primary key is an element, or set of components, in a table that serves as a means of distinguishing between records from the same table. A column in a table known as a foreign key is used to uniquely identify records from other tables. Up to the third normal form, all tables have been normalized. As the name implies, it denotes putting things in the normal form. The application developer via normalization tries to achieve a sensible organization of data into proper tables and columns and where names can be easily correlated to the data by the user. Normalization eliminates repeating groups at data and thereby avoids data redundancy which proves to be a great burden on the computer resources. These include:

- ✓ Normalize the data.
- ✓ Choose proper names for the tables and columns.
- ✓ Choose the proper name for the data.

### **First Normal Form**

According to the First Normal Form, each attribute in a tuple must have a single value from the attribute's domain and its domain must only include atomic values. The 1NF forbids "relations within relations" or "relations as attribute values within tuples," in other words. By 1NF, only single atomic or indivisible values are allowed for attribute values. Putting the data into First Normal Form is the first step. Data that is of a similar type in each table can be moved into separate tables to solve this problem. According to the project's requirements, a Primary Key or Foreign Key is assigned to each table. For each nested or non-atomic attribute, we create additional relations in this. For each nested relation or non-atomic attribute, new relations are formed in this process. This got rid of data groups that were repeated. If a relation solely meets the constraints that include the primary key, it is said to be in first normal form.

### **Second Normal Form**

No non-key attribute should be functionally dependent on a portion of the main key for relations where the primary key has several attributes, according to Second Normal Form. This involves breaking down each partial key into its dependent characteristics and setting up a new relation for each one. Keep the original primary key and any properties that are entirely dependent on it in your database. This procedure aids in removing data that depends only on a small portion of the key. If and only if a relation satisfies all the requirements for first normal form for the primary key and all of the non-primary key qualities of the relation are completely dependent on the primary key alone, then that relation is said to be in second normal form.

### **Third Normal Form**

A non-key attribute of a Relation should not be functionally determined by another non-key property or by a collection of non-key attributes, according to the Third Normal Form. In other words, the primary key shouldn't be transitively dependent on anything. In this, we break down the relation into its component parts and build up the non-key qualities that functionally determine the other non-key attributes. To eliminate anything that does not totally dependent on the Primary Key, this step is conducted. A relation is only considered to be in third normal form if it is in second normal form, and furthermore, the relation's non-key characteristics should not depend on other non-key attributes.

**1 . tbl\_registration**

FIELD NAME	DATATYPE	SIZE	DESCRIPTION	CONSTRAINT
id	INT	10	login id	PRIMARY KEY
name	VARCHAR	50	Full name	NOT NULL
username	VARCHAR	50	username	NOT NULL
email	VARCHAR	30	Email	NOT NULL
phone	INT	20	Phone number	NOT NULL
dob	DATE	5	Date of birth	NOT NULL
place	VARCHAR	20	Place	NOT NULL
state	VARCHAR	20	State	NOT NULL
country	VARCHAR	20	Country	NOT NULL
district	VARCHAR	20	District	NOT NULL
gender	VARCHAR	20	Gender	NOT NULL
aadhar	INT	15	Aadhar number	NOT NULL
licence_id	INT	20	License id	NOT NULL
password	VARCHAR	30	Password	NOT NULL
path	VARCHAR	50	Profile picture path	NOT NULL
status	INT	15	Account is blocked or not	NOT NULL

**2. tbl\_addvehicle**

FIELD NAME	DATATYPE	SIZE	DESCRIPTION	CONSTRAINT
vid	INT	10	Vehicle id	PRIMARY KEY
uid	INT	10	Owner id	FOREIGN KEY
Model_name	VARCHAR	20	Vehicle model name	NOT NULL
Model_number	VARCHAR	20	Model number	NOT NULL
Vehicle_number	VARCHAR	20	Vehicle number	NOT NULL
Build_year	INT	20	Build year	NOT NULL
Fuel_type	VARCHAR	20	Fuel type	NOT NULL
Vehicle_type	VARCHAR	20	Vehicle type	NOT NULL
imagepath	VARCHAR	20	Vehicle image path	NOT NULL

**3. tbl\_addtraveldetails**

FIELD NAME	DATATYPE	SIZE	DESCRIPTION	CONSTRAINT
Tvlid	int	10	Travel id	PRIMARY KEY
Tuid	int	10	User id	FOREIGN KEY
Tvid	int	10	Vehicle id	FOREIGN KEY
Froms	varchar	50	From	NOT NULL
Tos	varchar	50	To	NOT NULL
Routs	varchar	50	Routs	NOT NULL
Dates	date	5	Date	NOT NULL
Times	datetime	5	Time	NOT NULL
no_of_seats	int	11	Seats needed	NOT NULL
amount	int	11	Amount	NOT NULL

**4. tbl\_place**

FIELD NAME	DATATYPE	SIZE	DESCRIPTION	CONSTRAINT
placeid	int	10	Place id	PRIMARY KEY
Places	varchar	50	places	NOT NULL
Routes	varchar	50	Travel routs	NOT NULL

**5. tbl\_vhlbooking**

FIELD NAME	DATATYPE	SIZE	DESCRIPTION	CONSTRAINT
bkid	int	11	Booking id	PRIMARY KEY
ownerid	int	11	Owner id	FOREIGN KEY
userid	int	11	User id	FOREIGN KEY
travelid	int	11	Travel id	FOREIGN KEY
frm	varchar	30	From	NOT NULL
tto	varchar	30	To	NOT NULL
rrouts	varchar	30	Routs	NOT NULL
pickupdest	varchar	30	Pickup destination	NOT NULL
ttlseat	varchar	30	Total seats	NOT NULL
ddate	date	5	Date	NOT NULL
ttime	time	5	Time	NOT NULL
aamount	int	20	Amount	NOT NULL
sseatsneeded	int	11	Seats needed	NOT NULL
ownername	varchar	20	Owner name	NOT NULL
owner_email	varchar	20	Owner email	NOT NULL
owner_phone	int	30	Owner phone	NOT NULL
owner_place	varchar	20	Owner place	NOT NULL
Paymemt_status	int	11	Payment status	NOT NULL
proof	varchar	20	proof	NOT NULL
bkstatus	int	11	Booking status	NOT NULL

**6.tbl\_feedback**

FIELD NAME	DATATYPE	SIZE	DESCRIPTION	CONSTRAINT
fbkid	int	10	Feedback id	PRIMARY KEY
bookingid	int	11	Booking id	FOREIGN KEY
userid	int	11	User id	FOREIGN KEY
ownerid	int	11	Owners id	FOREIGN KEY
feedback	varchar	100	Feedback	NOT NULL
rateing	int	11	User rating	NOT NULL

**7. tbl\_payment**

FIELD NAME	DATATYPE	SIZE	DESCRIPTION	CONSTRAINT
pid	int	10	Payment id	PRIMARY KEY
P_userid	int	10	User_id	FOREIGN KEY
P_ownerid	int	20	Owner_id	FOREIGN KEY
P_tvlid	int	20	Travel id	FOREIGN KEY
P_amount	int	20	Total amount	NOT NULL

## **CHAPTER 5**

### **SYSTEM TESTING**

## 5.1 INTRODUCTION

Software testing is the process of carefully controlling the execution of software in order to determine whether it behaves as intended. The words verification and validation are frequently used in conjunction with software testing. Validation is the process of examining or evaluating a product, including software, to determine whether it complies with all relevant specifications. One type of verification, software testing, uses methods including reviews, analyses, inspections, and walkthroughs as well. Verifying that what has been specified matches what the user truly desired is the process of validation.

The processes of static analysis and dynamic analysis are additional ones that are frequently related to software testing. Static analysis examines the software's source code, searching for issues and obtaining statistics without actually running the code. Dynamic analysis examines how software behaves while it is running in order to offer data like execution traces, timing profiles, and test coverage details.

Testing is a collection of activities that can be planned ahead of time and carried out in a methodical manner. Testing starts with individual modules and progresses to the integration of the full computer-based system. There are many rules that can be used as testing objectives, and testing is necessary for the system testing objectives to be successful. They are:

Testing is a process of executing a program with the intent of finding an error.

- A good test case is one that has high possibility of finding an undiscovered error.
- A successful test is one that uncovers an undiscovered error.

If a test is successfully carried out in accordance with the aforementioned aims, it will reveal software bugs. Additionally, testing shows that the software functions seem to operate in accordance with the specification and that the performance requirements seem to have been satisfied. There are three ways to test program.

- For correctness
- For implementation efficiency
- For computational complexity



Test for correctness are supposed to verify that a program does exactly what it was designed to do. This is much more difficult than it may at first appear, especially for large programs.

## 5.2 TEST PLAN

A test plan suggests a number of required steps that need be taken in order to complete various testing methodologies. The activity that is to be taken is outlined in the test plan. A computer programme, its documentation, and associated data structures are all created by software developers. It is always the responsibility of the software developers to test each of the program's separate components to make sure it fulfils the purpose for which it was intended. In order to solve the inherent issues with allowing the builder evaluate what they have developed, there is an independent test group (ITG). Testing's precise goals should be laid forth in quantifiable language. The test plan should include information on the mean time to failure, cost to find and fix defects, remaining defect density or frequency of occurrence, and test work hours per regression test.

The levels of testing include:

- ❖ Unit testing
- ❖ Integration Testing
- ❖ Data validation Testing
- ❖ Output Testing

### 5.2.1 Unit Testing

The smallest unit of software design—the software component or module—is the focus of unit testing, which concentrates verification work. Important control pathways are examined in order to find faults inside the module's border using the component level design description as a guide. The scope set for unit testing and the relative complexity of tests. Unit testing can be carried out simultaneously for numerous components and is white-box focused. Information flow into and out of the programme unit under test is monitored by the modular interface to ensure appropriate operation. To make sure that temporary data is kept in its original form during all phases of an algorithm's execution, the local data structure is inspected. To confirm that each statement in a module has been executed at least once, boundary conditions are evaluated. Finally, each path for managing errors is examined.

Before starting any other test, tests of data flow over a module interface are necessary. All other tests are irrelevant if data cannot enter and depart the system properly. An important duty during the unit test is the selective examination of execution pathways. Error circumstances must be foreseen in good design, and error handling paths must be put up to cleanly reroute or halt work when an error does arise. The final step of unit testing is boundary testing. Software frequently fails at its limits.

In the Sell-Soft System, unit testing was carried out by treating each module as a distinct entity and subjecting them to a variety of test inputs. The internal logic of the modules had some issues, which were fixed. Each module is tested and run separately after coding. All unused code was eliminated, and it was confirmed that every module was functional and produced the desired outcome.

### **5.2.2 Integration Testing**

Integration testing is a methodical approach for creating the program's structure while also carrying out tests to find interface issues. The goal is to construct a programme structure that has been determined by design using unit tested components. The programme as a whole is tested. Correction is challenging since the size of the overall programme makes it challenging to isolate the causes. As soon as these mistakes are fixed, new ones arise, and the process repeats itself in an apparently unending cycle. All of the modules were integrated after unit testing was completed in the system to check for any interface inconsistencies. A distinctive programme structure also developed when discrepancies in programme structures were eliminated.

### **5.2.3 Validation Testing or System Testing**

The testing process comes to an end here. This involved testing the entire system in its entirety, including all forms, code, modules, and class modules. Popular names for this type of testing include system tests and black box testing.

The functional requirements of the software are the main emphasis of the black box testing approach. To completely exercise all functional requirements for a programme, the software engineer can create sets of input conditions using Black Box testing.

The following sorts of problems are targeted by black box testing: erroneous or missing functions, interface errors, data structure or external data access errors, performance errors, initialization errors, and termination errors.

### 5.2.4 Output Testing or User Acceptance Testing

The system under consideration has its user acceptance assessed; in this case, it must meet the needs of the company. When creating and making modifications as needed, the software should stay in touch with the user and viewpoint system. The following points are considered in this:

Output screen designs, input screen designs, and

The testing mentioned above is carried out using several test data types. In the system testing process, the preparation of test data is crucial. The system under study is evaluated using the test data after it has been prepared. During system testing, faults in the test data are once again found and fixed using the testing procedures described above. The fixes are also logged for use in the future.

### Automation Testing

Automation testing is the process of testing software and other tech products to ensure it meets strict requirements. Essentially, it's a test to double-check that the equipment or software does exactly what it was designed to do. It tests for bugs, defects, and any other issues that can arise with product development. Automation testing can be run at any time of the day. It uses scripted sequences to examine the software. It then reports on what's been found, and this information can be compared with earlier test runs.

### Benefits of Automation Testing

Detailed reporting capabilities - Automation testing uses well-crafted test cases for various scenarios. These scripted sequences can be incredibly in-depth, and provide detailed reports that simply wouldn't be possible when done by a human.

Improved bug detection - One of the main reasons to test a product is to detect bugs and other defects. Automation testing makes this process an easier one. It's also able to analyze a wider test coverage than humans may be able to.

- Simplifies testing - Testing is a routine part of the operations of most SaaS and tech companies. Making it as simple as possible is key. Using automation is extremely beneficial. When automating test tools, the test scripts can be reused
- Speeds up the testing process - Machines and automated technology work faster than humans. Along with improved accuracy, this is why we use them. In turn, this shortens your software development cycles.

- Reduces human intervention - Tests can be run at any time of day, even overnight, without needing humans to oversee it. Plus, when it's conducted automatically, this can also reduce the risk of human error.

### **5.2.5 Selenium Testing**

Selenium is an open-source tool that automates web browsers. It provides a single interface that lets you write test scripts in programming languages like Ruby, Java, NodeJS, PHP, Perl, Python, and C#, among others. The Selenium testing tool is used to automate tests across browsers for web applications. It's used to ensure high-quality web applications — whether they are responsive, progressive, or regular. Selenium is an open-source tool.

## Test cases for a Login Page

Project Name: Vehicle Sharing System					
Login Test Case					
Test Case ID: Fun_1			Test Designed By: Jobin Joseph		
Test Priority (Low/Medium/High): High			Test Designed Date: 18-07-2022		
Module Name: Login Screen			Test Executed By: Dr. Juby Mathew		
Test Title: Verify login with valid username and password			Test Execution Date: 18-07-2022		
Description: Test the Login Page					
Pre-Condition: User has valid username and password					
Step	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Navigation to Login Page		Login Page should be display ed	Login page displayed	Pass
2	Provide Valid username	Username: jobinjoseph2501@gmail.com	User should be able to Login	User Logged in and navigated to User Dashboad	Pass
3	Provide Valid Password	Password: 1234			
4	Click on Sign In button				
5	Provide Invalid username or password	Username: jobinjoseph25@gmail.com  Password: User1235	User should not be able to Login	Message for enter valid email id or password displayed	Pass
6	Provide Null username or Password	Username: null Password: null			
7	Click on Sign In button				

**Post-Condition:** User is validated with database and successfully login into account. The Account session details are logged in database.

### Code package

```
test.java  exam.java  login.java x
package logintest;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class login {
    public static void main(String[] args) {
        System.setProperty("webdriver.chrome.driver", "C:\\Users\\jobin\\Downloads\\chromedriver_win32\\chrome
        WebDriver driver=new ChromeDriver();

        driver.get("http://localhost/mypro/log.php");
        driver.findElement(By.id("name")).sendKeys("jobinjoseph2501@gmail.com");
        driver.findElement(By.id("pass")).sendKeys("1234");
        driver.findElement(By.id("submit1")).click();
        String actualUrl="http://localhost/mypro/owner/Home.php";
        String expectedUrl= driver.getCurrentUrl();
        if(actualUrl.equalsIgnoreCase(expectedUrl)) {
            System.out.println("Test passed");
        } else {
            System.out.println("Test failed");
        }
    }
}
```

```
<terminated> login [Java Application] C:\Users\jobin\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.3.v20220515-1416\jre\bin\javaw.exe (Jul 18
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/C:/Users/jobin/eclipse-workspace/SeleniumTest/selenium-server-4.3.0.jar!/org/slf4j/impl/Stat
SLF4J: Found binding in [jar:file:/C:/Users/jobin/Downloads/selenium-server-4.3.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.JDK14LoggerFactory]
Starting ChromeDriver 103.0.5060.53 (a1711811edd74ff1cf2150f36ffa3b0dae40b17f-refs/branch-heads/5060@{#853}) on port 51032
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
Jul 18, 2022 12:49:02 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected upstream dialect: W3C
Jul 18, 2022 12:49:02 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
INFO: Found exact CDP implementation for version 103
Test passed
```

Project Name: Vehicle Sharing System					
Registration Test Case					
Test Case ID: Registration			Test Designed By: Jobin Joseph		
Test-Priority (Low/Medium/High): High			Test Designed Date: 18-07-2022		
Module Name: Registration Screen			Test Executed By: Dr. Juby Mathew		
Test Title: User Registration Details			Test Execution Date: 18-07-2022		
Description: Register to system and Registration is completed then some error occurs, test will fail					
Pre-Condition: User has valid user name and password					
Step	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Navigation to Register Page		Register Page Should be displayed	Registration page displayed	Pass
2	Provide Valid Registration details	User Name: Jobin Joseph	User should be able to Register	User registration Completed after go to the login page	Pass
3					
4	Click on Register button				
5	Provide profile details	Input profile details	User will be redirected to Login page	User will be redirected to Login page	Pass
7	Click on register button				
8	Provide invalid information	Input invalid profile details.	User will be stay in register page	User will be stay on that page showing error message	Pass
9	Click on register button				
Post-Condition: User is validated with database and successfully login into account. The Account session details are logged in database.					

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class registration {
    public static void main(String[] args) {
        System.setProperty("webdriver.chrome.driver", "C:\\Users\\jobin\\Downloads\\chromedriver_win32\\chromedriver.exe");
        WebDriver driver=new ChromeDriver();

        driver.get("http://localhost/mypro/reg.php");
        driver.findElement(By.id("fname")).sendKeys("Jobin Joseph");
        driver.findElement(By.id("uname")).sendKeys("jobin@456");
        driver.findElement(By.id("usertype")).sendKeys("Vehicle Owner");
        driver.findElement(By.id("mail")).sendKeys("jobin@gmail.com");
        driver.findElement(By.id("phone")).sendKeys("7865423695");
        driver.findElement(By.id("dob")).sendKeys("25-02-1999");
        driver.findElement(By.id("place")).sendKeys("Mundakayam");
        driver.findElement(By.id("state")).sendKeys("kerala");
        driver.findElement(By.id("place")).sendKeys("Mundakayam");
        driver.findElement(By.id("country")).sendKeys("India");
        driver.findElement(By.id("district")).sendKeys("Kottayam");
        driver.findElement(By.id("gender")).sendKeys("Male");
        driver.findElement(By.id("aadhar")).sendKeys("7896321456987542");
        driver.findElement(By.id("licence")).sendKeys("44578789624");
        driver.findElement(By.id("passwd")).sendKeys("45678");
        driver.findElement(By.id("repasswd")).sendKeys("45678");
        driver.findElement(By.id("reginsert")).click();
        String actualUrl="http://localhost/mypro/owner/Home.php";
        String expectedUrl= driver.getCurrentUrl();
        if(actualUrl.equalsIgnoreCase(expectedUrl)) {
            System.out.println("Test passed");
        } else {
            System.out.println("Test failed");
        }
    }
}
```



```

login.java  registration.java  X
1 import org.openqa.selenium.By;
4
5 public class registration {
6     public static void main(String[] args) {
7         System.setProperty("webdriver.chrome.driver", "C:\\Users\\jobin\\Downloads\\chromedriver_win32\\chromedriver.exe");
8         WebDriver driver=new ChromeDriver();
9
10        driver.get("http://localhost/mypro/reg.php");
11        driver.findElement(By.id("fname")).sendKeys("Jobin Joseph");
12        driver.findElement(By.id("uname")).sendKeys("jobin@456");
13        driver.findElement(By.id("usertype")).sendKeys("Vehicle Owner");
14        driver.findElement(By.id("mail")).sendKeys("jobin@gmail.com");
15        driver.findElement(By.id("phone")).sendKeys("7865423695");
16        driver.findElement(By.id("dob")).sendKeys("25-02-1999");
17        driver.findElement(By.id("place")).sendKeys("Mundakayam");
18        driver.findElement(By.id("state")).sendKeys("kerala");
19        driver.findElement(By.id("place")).sendKeys("Mundakayam");
20        driver.findElement(By.id("country")).sendKeys("India");
21        driver.findElement(By.id("district")).sendKeys("Kottayam");
22        driver.findElement(By.id("gender")).sendKeys("Male");
23        driver.findElement(By.id("aadhar")).sendKeys("7896321456987542");
24        driver.findElement(By.id("licence")).sendKeys("44578789624");
25        driver.findElement(By.id("passwd")).sendKeys("45678");
26        driver.findElement(By.id("repasswd")).sendKeys("45678");
27        driver.findElement(By.id("reginsert")).click();
28        String actualUrl="http://localhost/mypro/owner/Home.php";
29        String expectedUrl= driver.getCurrentUrl();
30        if(actualUrl.equalsIgnoreCase(expectedUrl)) {
31            System.out.println("Test passed");
32        } else {
33            System.out.println("Test failed");
34        }
35    }
36 }

```

```

<terminated> login [Java Application] C:\Users\jobin\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_17.0.3.v20220515-1416\jre\bin\javaw.exe (Jul 18, 2022 12:49:02 PM)
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/C:/Users/jobin/eclipse-workspace/SeleniumTest/selenium-server-4.3.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/C:/Users/jobin/Downloads/selenium-server-4.3.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.JDK14LoggerFactory]
Starting ChromeDriver 103.0.5060.53 (a1711811edd74ff1cf2150f36ffa3b0dae40b17f-refs/branch-heads/5060@{#853}) on port 51032
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
Jul 18, 2022 12:49:02 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected upstream dialect: W3C
Jul 18, 2022 12:49:02 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
INFO: Found exact CDP implementation for version 103
Test passed

```

## **CHAPTER 6**

### **IMPLEMENTATION**

## 6.1 INTRODUCTION

The project's implementation phase is where the conceptual design is transformed into a functional system. It can be regarded as the most important stage in creating a successful new system since it gives users assurance that the system will operate as intended and be reliable and accurate. User documentation and training are its main concerns. Usually, conversion happens either during or after the user's training. Implementation is the process of turning a newly revised system design into an operational one, and it simply refers to placing a new system design into operation.

The user department now bears the most of the workload, faces the most disruption, and has the biggest influence on the current system. If the implementation is not well thought out or managed, confusion and mayhem may result.

Implementation encompasses all of the steps used to switch from the old system to the new one. The new system could be entirely different, take the place of an existing manual or automated system, or it could be modified to work better. A reliable system that satisfies organizational needs must be implemented properly. System implementation refers to the process of actually using the built system. This comprises all the processes involved in switching from the old to the new system. Only after extensive testing and if it is determined that the system is operating in accordance with the standards can it be put into use. The system personnel assess the system's viability. The system analysis and design work needed to implement the three key components of education and training, system testing, and changeover will increase in complexity as a system is implemented.

The implementation state involves the following tasks:

- Careful planning.
- Investigation of system and constraints. Design of methods to achieve the changeover.

## 6.2 IMPLEMENTATION PROCEDURES

Software implementation refers to the complete installation of the package in its intended environment, as well as to the system's functionality and satisfaction of its intended applications. The software development project is frequently commissioned by someone who will not be using it. People first have doubts regarding the software, but we must

Make sure that there isn't an increase in resistance because one must make sure that:

The new system's advantages must be known to the active user. Their faith in the software is increased. The user receives the appropriate instruction so that he feels confident using the application.

Before examining the system, the user must be aware that the server software needs to be running on the server in order to access the results. The actual process won't happen if the server object is not active and functioning on the server. User Training

The purpose of user training is to get the user ready to test and modify the system. The people who will be involved must have faith in their ability to contribute to the goal and benefits anticipated from the computer-based system. Training is more necessary as systems get more complicated. The user learns how to enter data, handle error warnings, query the database, call up routines that will generate reports, and execute other important tasks through user training. Training on the Application Software

After providing the necessary basic training on computer awareness the user will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the date entered. It should then cover information needed by the specific user/ group to use the system or part of the system while imparting the training of the program on the application. This training may be different across different user groups and across different levels of hierarchy

### **6.2.2 Training on the Application Software**

After providing the necessary basic training on computer awareness the user will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the date entered. It should then cover information needed by the specific user/ group to use the system or part of the system while imparting the training of the program on the application.

This training may be different across different user groups and across different levels of hierarchy

### **6.2.3 System Maintenance**

The mystery of system development is maintenance. When a software product is in the maintenance stage of its lifecycle, it is actively working. A system should be properly maintained after it has been effectively implemented. An essential part of the software development life cycle is system maintenance. In order for a system to be flexible to changes in the system environment, maintenance is required. Of course, software maintenance involves much more than just "Finding Mistakes."

### **6.2.4 Hosting**

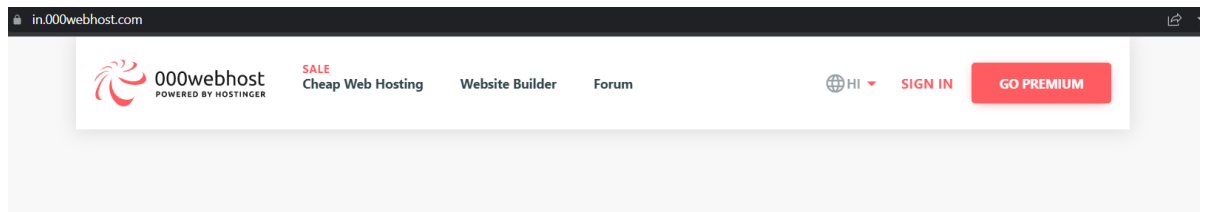
When a facilitating supplier distributes space on a web server for an online site to store its files, they are facilitating an online site. Web facilitating makes the records that include web site (code, pictures, etc.) accessible for seeing online. Every website you've ever gone to is facilitated on a server. The sum of space apportioned on a server to website depends on the sort of facilitating. The most sorts of facilitating are shared, committed, VPS and affiliate. They are separated by the kind of innovation utilized for the server, the level of administration given and the extra administrations on offer.

### **000Webhost**

Free web facilitating tends to be so constrained in capabilities and highlights that client must pay to induce what they need. However, 000webhost may be a free website facilitating arrangement that gives an cluster of important highlights, counting web site builder, WordPress back, and no advertisements. Clients can overhaul to a paid arrange to induce indeed more highlights and back, but based on our surveys, 000webhost is the most excellent free web facilitating arrangement for those who are genuinely on a tight budget.

## Setting up website in 000webhost

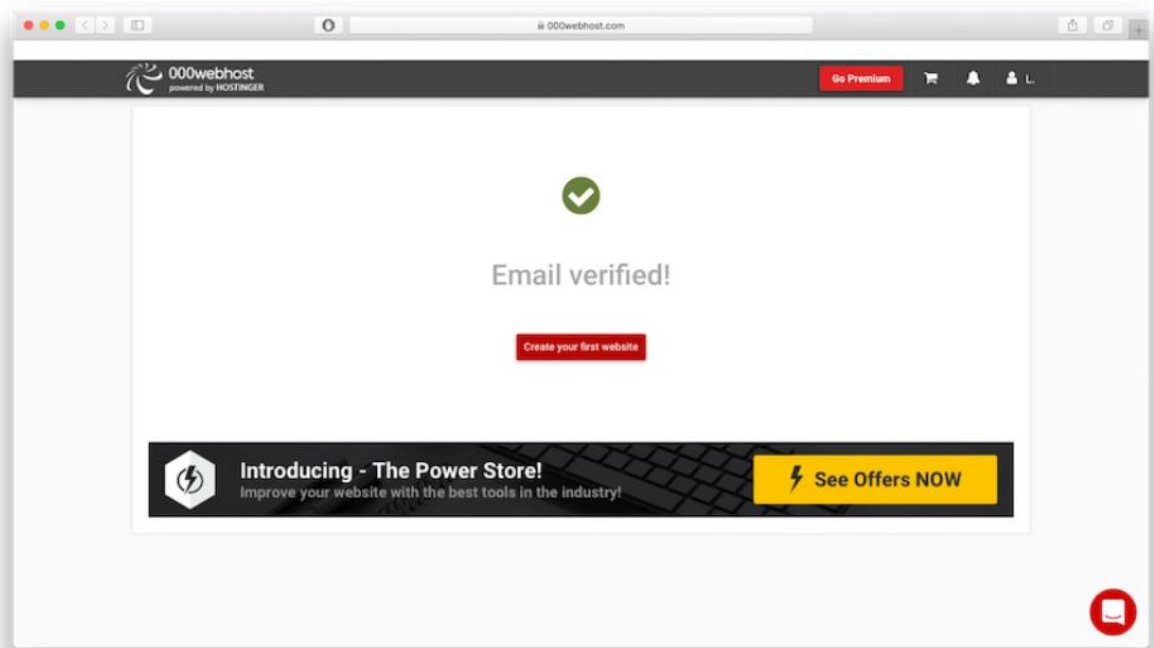
**Step 1:** Open 000webhost.com



**Step 2 :** Sign up using valid email

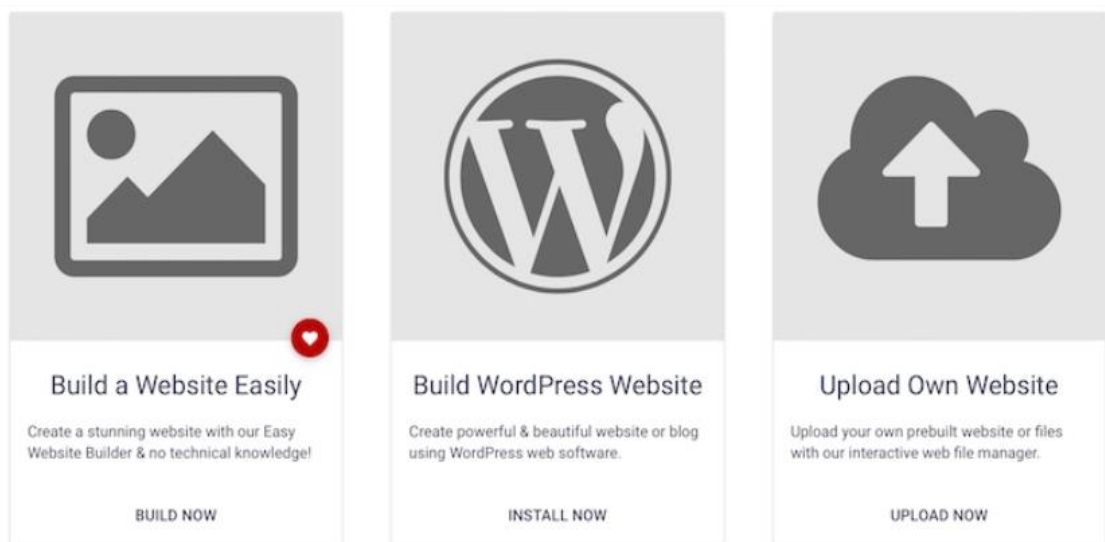
A screenshot of the 000webhost 'Sign Up' page. At the top is the 000webhost logo. Below it, the text 'Sign Up' is displayed in a large, bold, dark blue font. The form consists of three input fields: 'Email' containing 'jobinjoseph2501@gmail.com', 'Password' with masked characters '.....', and 'Repeat Password' also with masked characters '.....'. Below these fields is a large red button with the text 'SIGN UP' in white. At the bottom, there is a horizontal line with the word 'OR' in the center, indicating an alternative sign-up method.

**Step 3:** Verify mail id by clicking on the link send by 000webhost.com



**Step 4 :** search for a domain name and confirm (Provided by 000webhost)

**Step 5:** Upload website files



**Step 6:** Upload database

**Step 7:** update connection file with details provided by the system


## My Databases

[+ New Database](#)

1/2

DB Name	DB User	DB Host	
id19133352_vechilepo...	id19133352_jobin123	localhost	<a href="#">Manage ▾</a>

## Step 8 : Hosted website

**Vshare** Home Add Track Bookings Profile About Logout 

### Add Vehicle Details

Model name\*  Model Number\*


Vehicle Number\*  Build Year\*


Fuel Type\*  Vehicle Type\*  Picture\*

[Add](#)

#### Added Vehicles

Show  entries

Model Name	Model Nuber	vehicle Number	Build year	Fuel type	Vehicle Type	Picture	
BMW2	478989	kl27	2014	Diesel	4Wheeler		<a href="#">Edit</a>

Powered by  000webhost



## **CHAPTER 7**

### **CONCLUSION AND FUTURE SCOPE**

## **7.1 CONCLUSION**

This system is an effort to reduce consumption of fuel, our most important non-renewable resource and traffic congestion on roads by encouraging people to use Vehicle sharing. So, it is an environment-friendly social application and also helps people to reduce their journey time. It Also helps to meet new peoples. It Also helps in reducing stress from driving.

## **7.2 FUTURE SCOPE**

- The proposed system is designed in such a way that the payment should be done in online mode as well as cash on delivery.
- Users can able to do advanced search options
- Vehicle users can able to add complaints and feedbacks etc.
- Data security can be enhanced.
- Reduce the fuel consumption.

## **CHAPTER 8**

## **BIBLIOGRAPHY**

**REFERENCES:**

Gary B. Shelly, Harry J. Rosenblatt, “*System Analysis and Design*”, 2009.

Roger S Pressman, “*Software Engineering*”, 1994.

PankajJalote, “*Software engineering: a precise approach*”, 2006.

James lee and Brent ware Addison, “Open source web development with LAMP”, 2003

IEEE Std 1016 Recommended Practice for Software Design Descriptions.

**WEBSITES:**

- <https://getbootstrap.com/>
- [www.jquery.com](http://www.jquery.com)
- <https://api.jquery.com/jquery.ajax/>
- <https://app.diagrams.net>
- [www.agilemodeling.com/artifacts/useCaseDiagram.html](http://www.agilemodeling.com/artifacts/useCaseDiagram.html)

## **CHAPTER 9**

## **APPENDIX**

## 9.1 Sample Code

### Login.php

```
<?php
session_start();
?>

<html>
<head>

<script      src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-beta1/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
pprn3073KE6tl6bjs2QrFaJGz5/SUsLqktiwsUTF55Jfv3qYSDhgCecCxMW52nD2"
crossorigin="anonymous"></script>

    <!-- CSS only -->
<link          href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-beta1/dist/css/bootstrap.min.css"
rel="stylesheet"                                integrity="sha384-
0evHe/X+R7YkIZDRvuzKMRqM+OrBnVFBL6DOitfPri4tjfHxaWutUpFmBp4vmVor"
crossorigin="anonymous">
    <style>
        .gradient-custom-2 {
/* fallback for old browsers */
background: #fccb90;

/* Chrome 10-25, Safari 5.1-6 */
background: -webkit-linear-gradient(to right, #ee7724, #d8363a, #dd3675, #b44593);

/* W3C, IE 10+/ Edge, Firefox 16+, Chrome 26+, Opera 12+, Safari 7+ */
background: linear-gradient(to right, #ee7724, #d8363a, #dd3675, #b44593);
}

    @media (min-width: 768px) {
.gradient-form {
height: 100vh !important;
}
}
    @media (min-width: 769px) {
.gradient-custom-2 {
border-top-right-radius: .3rem;
border-bottom-right-radius: .3rem;
}
}

    </style>
</head>
<body>
<section class="h-100 gradient-form" style="background-color: #eee;">
<div class="container py-5 h-100">
<div class="row d-flex justify-content-center align-items-center h-100">
<div class="col-xl-10">
<div class="card rounded-3 text-black">
<div class="row g-0">
```

```

<div class="col-lg-6">
<div class="card-body p-md-5 mx-md-4">

<div class="text-center">

<h4 class="mt-1 mb-5 pb-1">Log in</h4>
</div>

<p>Please login to your account</p>
<form action="log.php" method="POST" enctype="multipart/form-data">
<div class="form-outline mb-4">
<input type="email" id="form2Example11" class="form-control"
placeholder="enter email address" name="email" />
<label class="form-label" for="form2Example11">Username</label>
</div>

<div class="form-outline mb-4">
<input type="password" id="form2Example22" class="form-control" name="passwd" />
<label class="form-label" for="form2Example22">Password</label>
</div>
<div class="text-center pt-1 mb-5 pb-1">
<button class="btn btn-primary btn-block fa-lg gradient-custom-2 mb-3" type="submit"
name="btnlogin">Log
in</button>
</form>
<a class="text-muted" href="#">Forgot password?</a>
</div>
<div class="d-flex align-items-center justify-content-center pb-4">
<p class="mb-0 me-2">Don't have an account?</p>
<a href="reg.php"> <button type="button" class="btn btn-outline-danger">Create
new</button></a>
</div>

</div>
</div>
<div class="col-lg-6 d-flex align-items-center gradient-custom-2">
<div class="text-white px-3 py-4 p-md-5 mx-md-4">
<h4 class="mb-4">We are more than just a company</h4>
<p class="small mb-0">The vehicle Sharing system is a web-based application to provide us with a
simple riding platform between the vehicle owner and a user. The vehicle owner can post a travel
destination prior to the travel date and if anyone is interested to travel in the same location, he/she
can contact the vehicle owner and book the ride with an amount specified by the owner. Personal ride
booking and sharing services allow customers to arrange transportation quickly and at a reasonable
amount. This application typically matches a customer's location with the nearest available vehicle.
This project consists of a website and its mobile application.
This website keeps the data in a centralized way which is available to all the users at the same time.
This system is an effort to reduce consumption of fuel, our most important non-renewable resource
and traffic congestion
on roads by encouraging people to use Vehicle sharing. So it is an environment-friendly social
application and also helps people to reduce their journey time.</p>
</p>
</div>
</div>
</div>

```

```
</div>
</div>
</div>
</div>
</section></body>
</html>
<?php
include("connect.php");

if(isset($_POST["btnlogin"])){

    $uname=$_POST["email"];
    $pass=$_POST["passwd"];

    $sql="select * from registration where email='$uname' && password='$pass'";

    $result=mysqli_query($con,$sql);

    if ($result->num_rows > 0) {
    while($row = $result->fetch_assoc()) {
        $_SESSION['uid']= $row["id"];
        $_SESSION['nnm']= $row["name"];
        $_SESSION['prof']= $row["path"];

        $typ=$row["usertype"];

    }}

    $count=mysqli_num_rows($result);

    if($count>0)
    {
        if($typ=="Vehicle_User")
        {

            echo "<script>window.location.href='Home.php';</script>";

        }
        else if($typ=="")
        {

            header("location:Home.php");

        }

    }
    else
    {
        ?>
        <script>
        alert("invalid username or password");
```



```

    </script>
    <?php
    }
  }
?>

```

### Registraion.php

```

<html>
<head>
<link      href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-beta1/dist/css/bootstrap.min.css"
rel="stylesheet"                                integrity="sha384-
0evHe/X+R7YkIZDRvuzKMRqM+OrBnVFBL6DOitfPri4tjfHxaWutUpFmBp4vmVor"
crossorigin="anonymous">
<script    src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-beta1/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
pprn3073KE6tl6bjs2QrFaJGz5/SUsLqktiwsUTF55Jfv3qYSDhgCecCxMW52nD2"
crossorigin="anonymous"></script>

<script type = "text/javascript">

function validate() {
var password1=document.myForm.passwd.value;
var password2=document.myForm.repasswd.value;

if (document.myForm.fname.value=="")
{
document.getElementById('fn').innerHTML="**Enter fullname";
return false;
}

if (document.myForm.uname.value=="")
{
document.getElementById('unam').innerHTML="**Enter Username";
return false;
}
var filter = /^[a-zA-Z0-9_\.\\-]+\@((([a-zA-Z0-9\\-])+\.)+([a-zA-Z0-9]{2,4})+)$/;
if (document.myForm.mail.value==" " || !filter.test(document.myForm.mail.value))
{

document.getElementById('em').innerHTML="**Enter a valid Email";
return false;
}
if (document.myForm.phone.value=="")
{
document.getElementById('pn').innerHTML="**Enter Phone";
return false;
}
var val = document.myForm.phone.value;
if (/^\d{10}$/.test(val)==false) {
// value is ok, use it
document.getElementById('pn').innerHTML="**No must be 10 digit";

return false;
}
if (document.myForm.place.value=="")

```

---

```
{
document.getElementById('pl').innerHTML="**Enter Place";
return false;
}

if (document.myForm.state.value=="")
{
document.getElementById('st').innerHTML="**Enter State";
return false;
}

var valpin = document.myForm.pin.value;
if (/^\d{6}$/.test(valpin)==false) {
// value is ok, use it
document.getElementById('pi').innerHTML="**Pin must be 6 digit";

return false;
}

if (document.myForm.licence.value=="")
{
document.getElementById('ln').innerHTML="**Enter Dob";
return false;
}
if (document.myForm.aadhar.value=="")
{
document.getElementById('ad').innerHTML="**Enter Adhar";
return false;
}
var valadhar = document.myForm.aadhar.value;
if (/^\d{18}$/.test(valadhar)==false) {
// value is ok, use it
document.getElementById('ad').innerHTML="**Adhar no must be 18 digit";

return false;
}
if (document.myForm.passwd.value=="")
{
document.getElementById('pa').innerHTML="**Enter Password";
return false;
}
if (document.myForm.repasswd.value=="")
{
document.getElementById('cpa').innerHTML="**Enter Confirm password";
return false;
}
if (password1 != password2) {
alert ("\nPassword did not match: Please try again....");
return false;
}

}
```

---

---

```

</script>
</head>
<body class="p-3 mb-2 bg-dark text-white">
<br>
<center><h1>Sign Up</h1></center>
<form name="myForm" method="POST" enctype="multipart/form-data" onsubmit =
"return(validate());">

<!-- action="view.php" -->
<div class="container">

<div class="row">

<div class="col">

<div class="mb-3">

<label class="form-label">Full Name <span id="fn" style="color:red;"></span></label><label
style="color:red;"> *</label>

<input type="text" value="" class="form-control" id="fname" name="fname">

</div>

</div>

<div class="col">

<div class="mb-3">

<label class="form-label">User Name <span id="unam" style="color:red;"></span></label><label
style="color:red;"> *</label>

<input type="text" value="" class="form-control" id="uname" name="uname">

</div>

</div>

<div class="col">

<div class="mb-3">

<label class="form-label">User Type</label><label style="color:red;"> *</label>

<select class="form-select col-4" aria-label="Default select example" id="usertype"
name="usertype">
<option value="Vehicle_Owner">Vehicle Owner</option>
<option value="Vehicle_User">Vehicle User</option>

</select>
</div>

</div>

</div>

```

---

---

```
<div class="row">

<div class="col">

<div class="mb-3">

<label class="form-label">Email address<span id="em" style="color:red;"></span></label><label
style="color:red;"> *</label>

<input type="text" class="form-control" id="mail" name="mail" value="" >

<div id="emailHelp" class="form-text">We'll never share your email with anyone else.</div>

</div>

</div>

<div class="col">

<div class="mb-3">

<label class="form-label">phone<span id="ph" style="color:red;"></span></label><label
style="color:red;"> *</label>

<input type="text" class="form-control" id="phone" name="phone" value="">

</div>

</div>

<div class="col">

<div class="mb-3">

<label class="form-label">Date Of Birth</label><label style="color:red;"> *</label>

<input type="date" class="form-control" id="dob" name="dob" value="" max="<?= date('Y-m-d');
?>" />

</div>

</div>

</div>

<div class="row">

<div class="col">

<div class="mb-3">

<label class="form-label">Place<span id="pl" style="color:red;"></span></label><label
style="color:red;"> *</label>

<input type="text" class="form-control" id="place" name="place" value="">

</div>
```

---

```
</div>

<div class="col">

<div class="mb-3">

<label class="form-label">State</label><label style="color:red;"> *</label>

<div class="container">

<div class="row">

<select class="form-select col-4" aria-label="Default select example" id="state" name="state">
<option value="Kerala">Kerala</option>
<option value="Andhra Pradesh">Andhra Pradesh</option>
<option value="Andaman and Nicobar Islands">Andaman and Nicobar Islands</option>
<option value="Arunachal Pradesh">Arunachal Pradesh</option>
<option value="Assam">Assam</option>
<option value="Bihar">Bihar</option>
<option value="Chandigarh">Chandigarh</option>
<option value="Chhattisgarh">Chhattisgarh</option>
<option value="Dadar and Nagar Haveli">Dadar and Nagar Haveli</option>
<option value="Daman and Diu">Daman and Diu</option>
<option value="Delhi">Delhi</option>
<option value="Lakshadweep">Lakshadweep</option>
<option value="Puducherry">Puducherry</option>
<option value="Goa">Goa</option>
<option value="Gujarat">Gujarat</option>
<option value="Haryana">Haryana</option>
<option value="Himachal Pradesh">Himachal Pradesh</option>
<option value="Jammu and Kashmir">Jammu and Kashmir</option>
<option value="Jharkhand">Jharkhand</option>
<option value="Karnataka">Karnataka</option>

<option value="Madhya Pradesh">Madhya Pradesh</option>
<option value="Maharashtra">Maharashtra</option>
<option value="Manipur">Manipur</option>
<option value="Meghalaya">Meghalaya</option>
<option value="Mizoram">Mizoram</option>
<option value="Nagaland">Nagaland</option>
<option value="Odisha">Odisha</option>
<option value="Punjab">Punjab</option>
<option value="Rajasthan">Rajasthan</option>
<option value="Sikkim">Sikkim</option>
<option value="Tamil Nadu">Tamil Nadu</option>
<option value="Telangana">Telangana</option>
<option value="Tripura">Tripura</option>
<option value="Uttar Pradesh">Uttar Pradesh</option>
<option value="Uttarakhand">Uttarakhand</option>
<option value="West Bengal">West Bengal</option>

</select>

</div>
```

```
</div>

</div>

</div>

<div class="col">

<div class="mb-3">

<label class="form-label">Country</label><label style="color:red;"> *</label>

<div class="container">

<div class="row">

<select class="form-select col-4" aria-label="Default select example" id="country"
name="country">
<option value="India">India</option>
</select>

</div>

</div>

</div>

</div>

</div>

<div class="col">

<div class="mb-3">

<label class="form-label">District</label><label style="color:red;"> *</label>

<input type="text" class="form-control" id="district" name="district" value="">
</div>

</div>

<div class="col">

<div class="mb-3">

<label class="form-label">Gender</label><label style="color:red;"> *</label>
<select class="form-select col-4" aria-label="Default select example" id="gender" name="gender">
<option value="MALE">MALE</option>
<option value="FEMALE">FEMALE</option>
<option value="FEMALE">OTHER</option>
```

---

```

</select>
</div>

</div>

<div class="col">

<div class="mb-3">

<label class="form-label">Aadhaar No<span id="ad" style="color:red;"></span></label><label
style="color:red;"> *</label>

<input type="text" class="form-control" id="aadhar" name="aadhar" value="">
</div>

</div>
</div>

<div class="row">

<div class="col">

<div class="mb-3">

<label class="form-label">Licence Id<span id="ln" style="color:red;"></span></label><label
style="color:red;"> *</label>

<input type="text" class="form-control" id="licence" name="licence" value="">
</div>
</div>
<div class="col">

<div class="mb-3">

<label class="form-label">Password<span id="pa" style="color:red;"></span></label><label
style="color:red;"> *</label>

<input type="text" class="form-control" id="passwd" name="passwd" value="">
</div>
</div>
<div class="col">

<div class="mb-3">

<label class="form-label">Re Enter Password<span id="cpa"
style="color:red;"></span></label><label style="color:red;"> *</label>

<input type="text" class="form-control" id="repasswd" name="repasswd" value="">
</div>
</div>

</div>

```

---

---

```

<div class="row">
<div class="col">

<div class="mb-3">

<img id="uploadPreview" style="width: 100px; height: 100px;" />
<input      id="uploadImage"      type="file"      accept="image/*"      name="myPhoto"
onchange="PreviewImage();" />
<script type="text/javascript">

    function PreviewImage() {
        var oFReader = new FileReader();
        oFReader.readAsDataURL(document.getElementById("uploadImage").files[0]);

        oFReader.onload = function (oFREvent) {
            document.getElementById("uploadPreview").src = oFREvent.target.result;
        };
    };

</script>
</div>
</div>
<center><div class="col-12">
<button type="submit" name="reginsert" class="btn btn-primary col-2">Sign up</button></div>
</center>
</div>

</body>
</html>
<?php
include("connect.php");

if(isset($_POST["reginsert"]))
{

    $fname=$_POST["fname"];
    $Username=$_POST["uname"];
    $usertype=$_POST["usertype"];
    $email=$_POST["mail"];
    $phone=$_POST["phone"];
    $dob=$_POST["dob"];
    $place=$_POST["place"];
    $state=$_POST["state"];
    $country=$_POST["country"];
    $gender=$_POST["gender"];

    $aadhar=$_POST["aadhar"];

    $licence=$_POST["licence"];
    $passwd=$_POST["repasswd"];

    $msg="";
    $imgpath=$_FILES["myPhoto"]["name"];

    // check if the user has clicked the button "UPLOAD"

```

---



---

```
$filename = $_FILES["myPhoto"]["name"];

$tempname = $_FILES["myPhoto"]["tmp_name"];

$folder = "propic/".$filename;
if (move_uploaded_file($tempname, $folder)) {

    $msg = "Image uploaded successfully";

} else {

    $imgpath="user.png";
}
$sql="INSERT INTO `registration`(`name`, `username`, `usertype`, `email`, `phone`, `dob`,
`place`, `state`, `country`, `gender`,
`aadhar`, `Licence_id`, `password`, `path`)VALUES
('$fname','$Username','$usertype','$email','$phone','$dob','$place','$state',
'$country','$gender','$aadhar','$licence','$passwd','$imgpath')";

if(mysqli_query($con,$sql))
{
    ?>
<script>
    alert("inserted");
</script>
<?php
}
else
{
    echo "error";
}

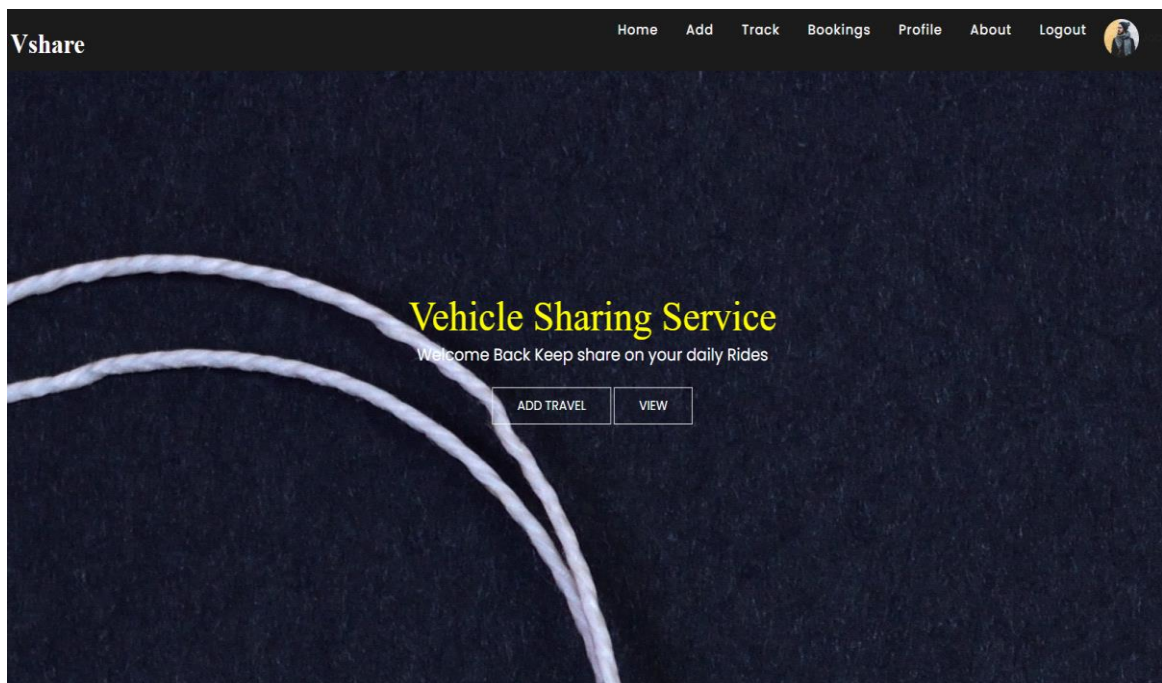
}
?>
```

## 9.3 Screen Shots

### Home page



### Vehicle owner home page



## Add Vehicle

Vshare
Home Add Track Bookings Profile About Logout



### Add Vehicle Details

Model name\*
Model Number\*
Vehicle Number\*
Build Year\*
Fuel Type\*
Vehicle Type\*
Picture\*

Add

#### Added Vehicles

Show 10 entries
Search:

Model Name	Model Nuber	vehicle Number	Build year	Fuel type	Vehicle Type	Picture
BMW2	478989	kl27	2014	Diesel	4Wheeler	 Edit Delete
swift	4587	kl151234	2016	Diesel	4Wheeler	 Edit Delete

## Add Travel


Vshare
Home Add Track Bookings Profile About Logout

### Add Travel Details

From\*
To\*
Rout map\*
Date\*
Time\*
No of Seats Available\*
Rate /seat\*
Luggage space Avilable\*
Select\_Vehicle\*

Add Travel

## Manage Travel Requests

**Vshare**[Home](#)[Add](#)[Track](#)[Bookings](#)[Profile](#)[About](#)[Logout](#)

### My Travel Status

Show  entries


Search:

From	↑↓ Routs	↑↓ To	↑↓ Date	↑↓ Time	↑↓ no_of_seats	↑↓ Rate	↑↓
kottayam	kply kanjirapally ponkunnam	pala	2022-07-11	11:4	2	150	<a href="#">View</a>
mundakayam	kanjirappally ponkunnam	pala	2022-07-12	10:4	3	100	<a href="#">View</a>


Showing 1 to 2 of 2 entries

[Previous](#) [1](#) [Next](#)

## Edit profile

**Vshare**[Home](#)[Add](#)[Track](#)[Bookings](#)[Profile](#)[About](#)[Logout](#)

Profile Picture



JPG or PNG no larger than 5 MB

[Upload new image](#)

Edit Profile Details

Fullname (how your name will appear to other users on the site)

Username

Email

Phone

Place

State

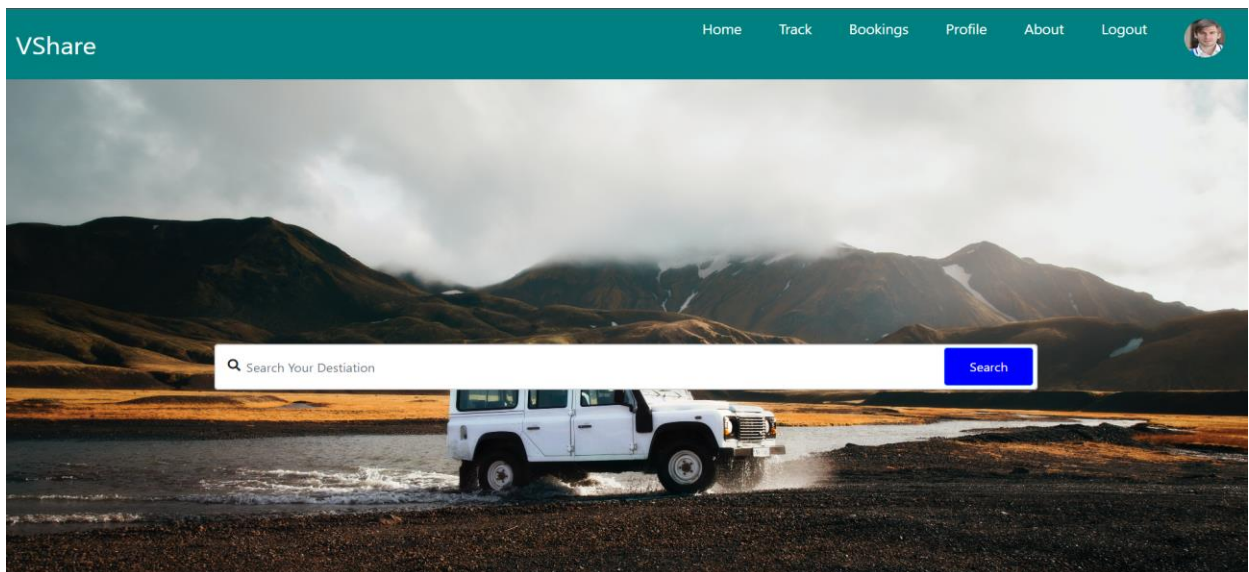
country

District

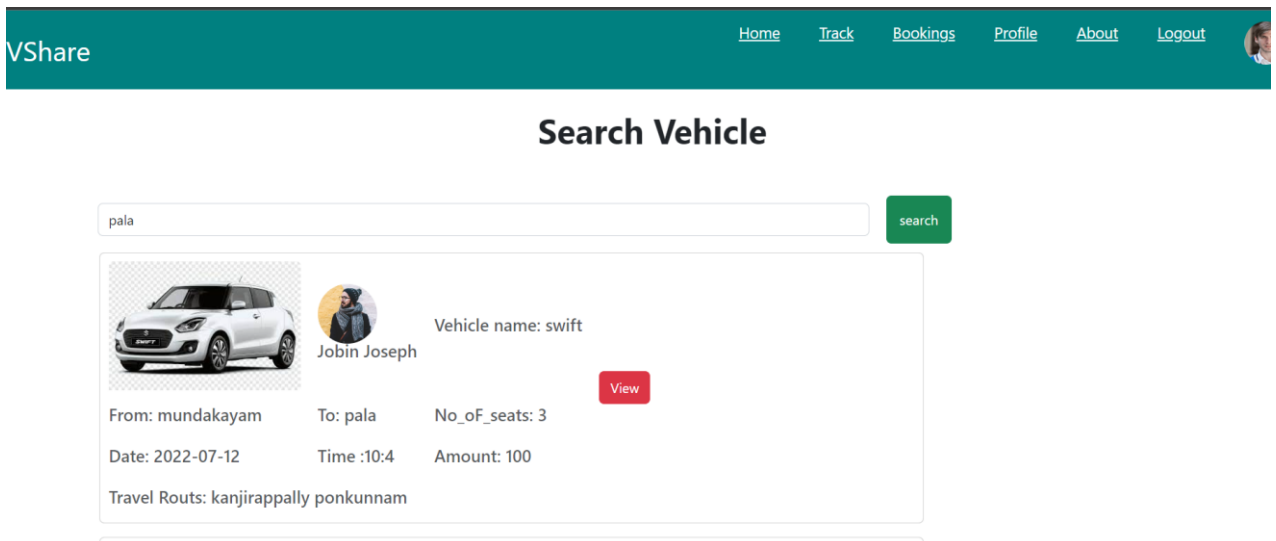
[Save changes](#)

## VEHICLE USER'S PAGES

### Users home page



### Search vehicle



## Request status

VShare [Home](#) [Track](#) [Bookings](#) [Profile](#) [About](#) [Logout](#)

### My Travel Details

Show  entries Search:

From	Routes	To	Date	Time	no_of_seats	Total_seats	amount	Status
kottayam	kply kanjirapally ponkunnam	pala	2022-07-11	11:40	1	2	150	Approved <a href="#">View</a>
mundakayam	kanjirapally ponkunnam	pala	2022-07-12	10:40	2	1	100	Approved <a href="#">View</a>

Showing 1 to 2 of 2 entries Previous **1** Next

VShare [Home](#) [Track](#) [Bookings](#) [Profile](#) [About](#) [Logout](#)

### Your Booking Status

From	Routes	To	Date	Time	Totalseats	seatsneeded	Rate	Payment Status
kottayam	kply kanjirapally ponkunnam	pala	2022-07-11	11:40	1	2	150	unpaid

Status: Approved [Cancel](#) [Pay Online](#) [Print](#)

## Payment

V-Share  
Travel Amount  
₹ 300  
English

Country: +91 Phone Number: 4586932147

Email: kmabijith@gmail.com

This payment is secured by Razorpay.

PROCEED

V-Share  
Travel Amount  
₹ 300  
English

+914586932147 | kmabijith@gmail.c...

PREFERRED PAYMENT METHODS

UPI - PhonePe

UPI - PayTM

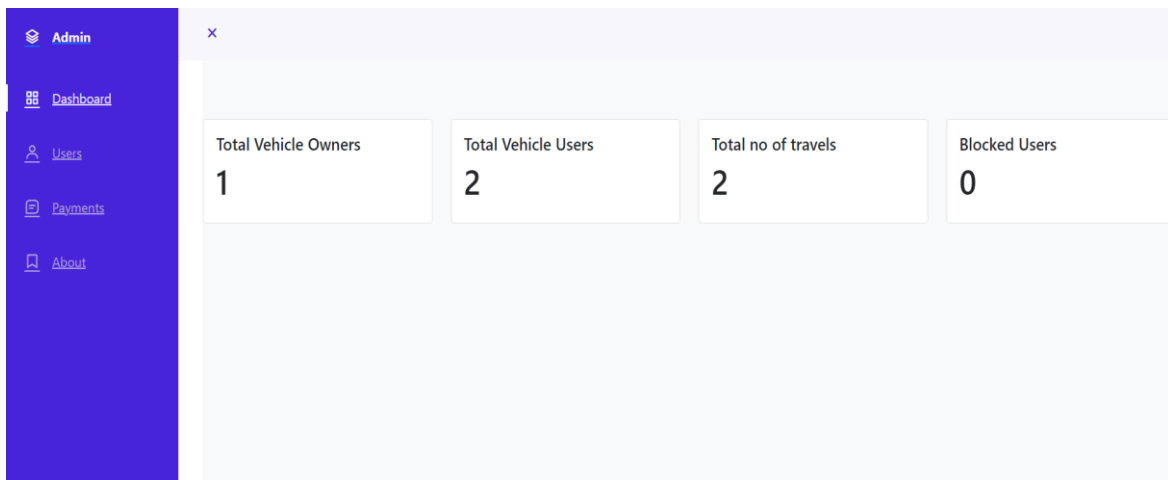
CARDS, UPI & MORE

Card  
Visa, MasterCard, RuPay, and Maestro

UPI / QR

## ADMIN PAGES

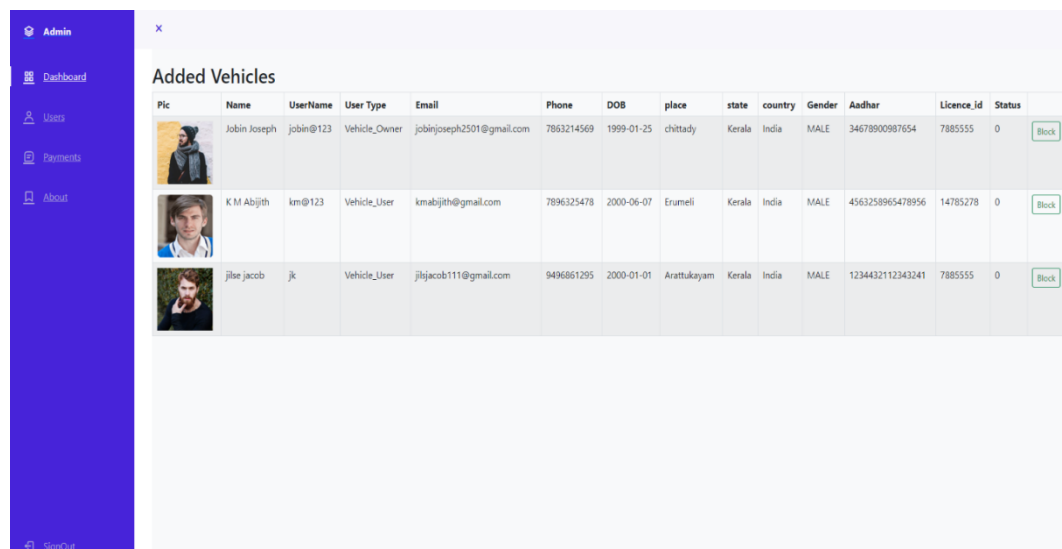
### Admin Home






The Admin Home dashboard features a blue sidebar with navigation links: Admin, Dashboard, Users, Payments, and About. The main content area displays four summary cards:

- Total Vehicle Owners: 1
- Total Vehicle Users: 2
- Total no of travels: 2
- Blocked Users: 0

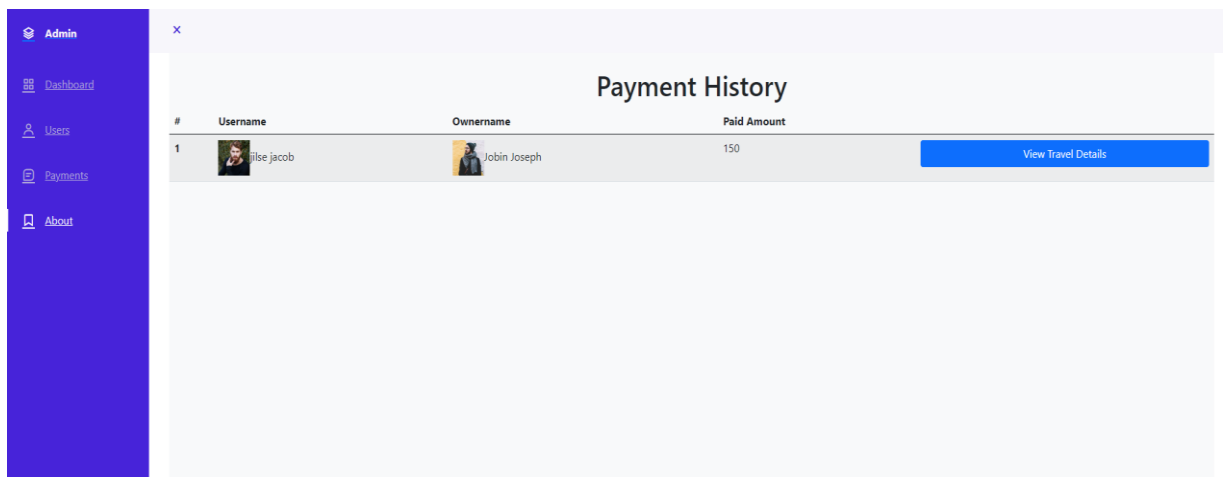
### Manage Users and Owners





The Manage Users and Owners page displays a table titled "Added Vehicles" with the following data:

Pic	Name	UserName	User Type	Email	Phone	DOB	place	state	country	Gender	Aadhar	Licence_id	Status	
	Jobin Joseph	jobin@123	Vehicle_Owner	jobinjoseph2501@gmail.com	7863214569	1999-01-25	chittady	Kerala	India	MALE	34678900987654	7885555	0	<button>Block</button>
	K M Abijith	km@123	Vehicle_User	kmabijith@gmail.com	7896325478	2000-06-07	Erumeli	Kerala	India	MALE	4563258965478956	14785278	0	<button>Block</button>
	jilse jacob	jk	Vehicle_User	jilsejacob111@gmail.com	9496861295	2000-01-01	Arattukayam	Kerala	India	MALE	1234432112343241	7885555	0	<button>Block</button>

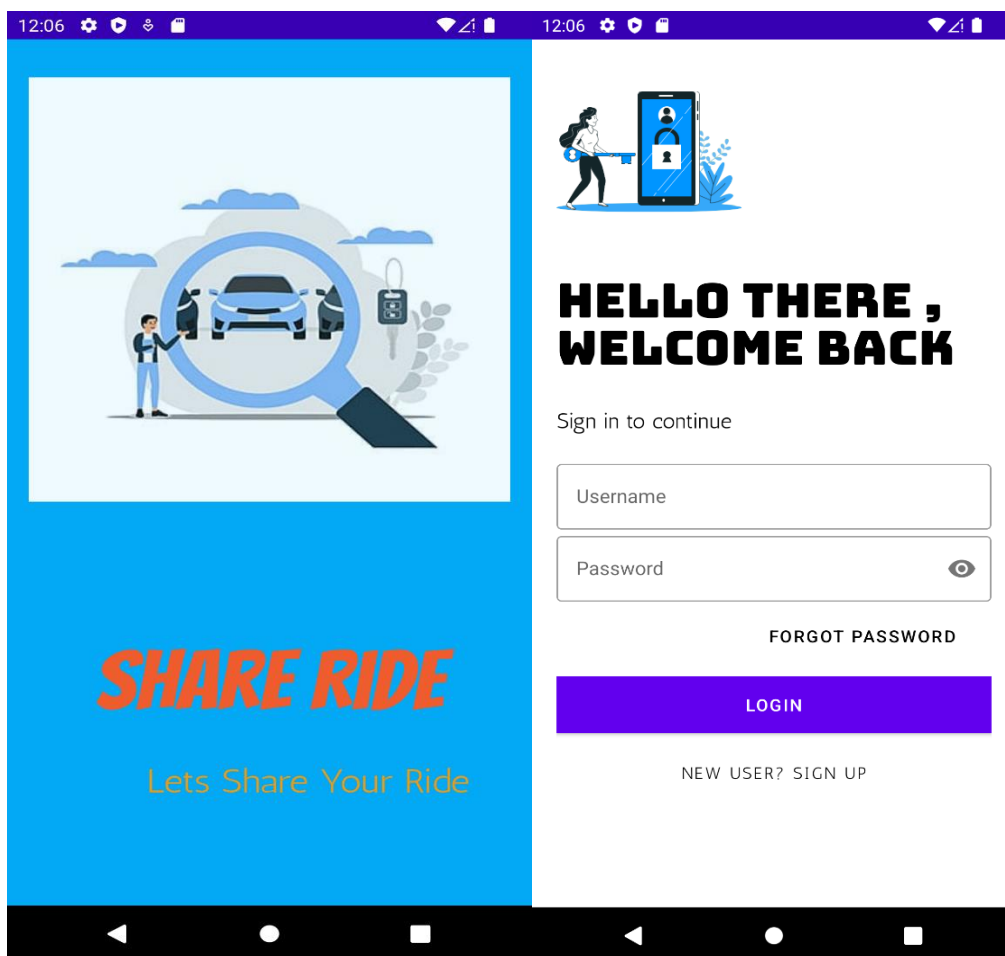
## Payment History



#	Username	Ownername	Paid Amount
1	 ilse jacob	 Jobin Joseph	150 <a href="#">View Travel Details</a>

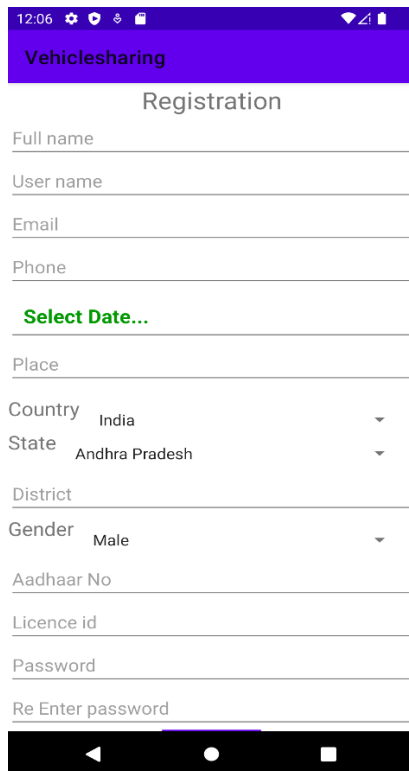
## Android App

### Splash screen and Login page



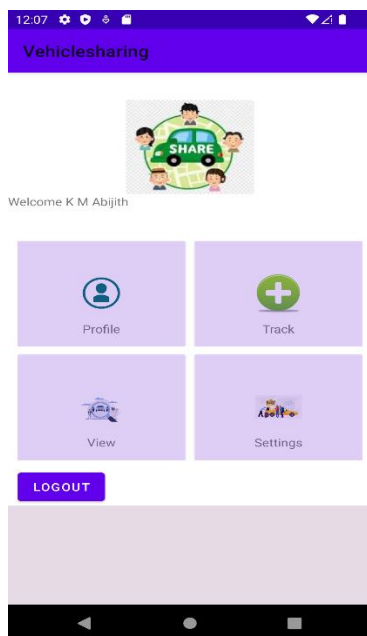


## Registration





The screenshot shows the Registration screen of the Vehiclesharing app. The status bar at the top indicates the time is 12:06. The app's title bar is purple and says "Vehiclesharing". The screen has a white background with the title "Registration" at the top. Below the title are several input fields: "Full name", "User name", "Email", "Phone", "Select Date..." (in green), "Place", "Country" (with a dropdown menu showing "India"), "State" (with a dropdown menu showing "Andhra Pradesh"), "District", "Gender" (with a dropdown menu showing "Male"), "Aadhaar No", "Licence id", "Password", and "Re Enter password". At the bottom, there is a black navigation bar with three white icons: a back arrow, a circle, and a square.

## Home page



## Booking page

12:07  Vehiclesharing



swift

**mundakayam To pala**


**Routs :** kanjirappally ponkunnam

**Date :** 2022-07-12 **Time :** 10:4

**FuelType :** Diesel **Build Year :** 2016


**Total Seats :** 3 **Amount :** 100


**Vehicle Owner Details**



**Name :** jilse jacob **Username :** jk  
**Email :** jilsjacob111@gmail.com  
**Phone :** 9496861295  
**Place :** Arattukayam

**BOOKNOW**

12:08  Vehiclesharing





swift

**mundakayam To pala**

**Routs :** kanjirappally ponkunnam

**Date :** 2022-07-12 **Time :** 10:4

**Total Seats :** 3 **Amount :** 100

select Seats :  1 

**BOOKNOW**