

VECHILE SHARING MANAGEMENT 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
(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]

2021-2022

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

Rev.Fr.Dr.Rubin Thottupurathu Jose

Coordinator

Rev.Fr.Dr.Rubin Thottupurathu Jose
Head of the Department

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 (MCA) from APJ Abdul Kalam Technological University, during the academic year 2021-2022.

Date:

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. The online 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.

CONTENT

Sl. No	Topic	Page No
1	INTRODUCTION	1
1.1	PROJECT OVERVIEW	2
1.2	PROJECT SPECIFICATION	2
2	SYSTEM STUDY	4
2.1	INTRODUCTION	5
2.2	EXISTING SYSTEM	6
2.3	DRAWBACKS OF EXISTING SYSTEM	6
2.4	PROPOSED SYSTEM	6
2.5	ADVANTAGES OF PROPOSED SYSTEM	6
3	REQUIREMENT ANALYSIS	8
3.1	FEASIBILITY STUDY	9
3.1.1	ECONOMICAL FEASIBILITY	9
3.1.2	TECHNICAL FEASIBILITY	9
3.1.3	BEHAVIORAL FEASIBILITY	9
3.2	SYSTEM SPECIFICATION	10
3.2.1	HARDWARE SPECIFICATION	10
3.2.2	SOFTWARE SPECIFICATION	10
3.3	SOFTWARE DESCRIPTION	10
3.3.1	PHP	10
3.3.2	MYSQL	11
4	SYSTEM DESIGN	13
4.1	INTRODUCTION	14
4.2	UML DIAGRAM	14
4.2.1	USE CASE DIAGRAM	15
4.2.2	SEQUENCE DIAGRAM	18
4.2.3	STATE CHART DIAGRAM	20
4.2.4	ACTIVITY DIAGRAM	22
4.2.5	CLASS DIAGRAM	22
4.2.6	OBJECT DIAGRAM	24
4.2.7	COMPONENT DIAGRAM	25
4.2.8	DEPLOYMENT DIAGRAM	26

4.3	USER INTERFACE DESIGN	27
4.4	DATA BASE DESIGN	33
5	SYSTEM TESTING	39
5.1	INTRODUCTION	40
5.2	TEST PLAN	41
5.2.1	UNIT TESTING	41
5.2.2	INTEGRATION TESTING	42
5.2.3	VALIDATION TESTING	42
5.2.4	USER ACCEPTANCE TESTING	43
6	IMPLEMENTATION	44
6.1	INTRODUCTION	45
6.2	IMPLEMENTATION PROCEDURE	45
6.2.1	USER TRAINING	46
6.2.2	TRAINING ON APPLICATION SOFTWARE	46
6.2.3	SYSTEM MAINTENANCE	46
7	CONCLUSION & FUTURE SCOPE	47
7.1	CONCLUSION	48
7.2	FUTURE SCOPE	48
8	BIBLIOGRAPHY	49
9	APPENDIX	51
9.1	SAMPLE CODE	52
9.2	SCREEN SHOTS	79

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 aim of this project is to make a Vehicle Sharing Service Application. The online 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 app.

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. The ride fare is fixed by the vehicle owner within the range of kilometers travelled and will even take care of all administrative issues. It is simple, fast, and cost – effective.

The system includes 3 modules. They are:

1. Admin Module

Admin must have a login into this system. He has the overall control of the system. Admin can add or block vehicle owners as well as users. Admin can View all the registered users and also manage all his data.

2. Vehicle Owner Module

Vehicle owners can register and they can add their vehicle. Also, they can add travel details.

3. User Module

Users can search for travel and can book a vehicle with an amount specified by the owner.

CHAPTER 2

SYSTEM STUDY

2.1 INTRODUCTION

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information to recommend improvements on the system. It is a problem-solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is studied to the minute's detail and analyzed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The system is viewed as a whole and the input to the system are identified. The outputs from the organizations are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analyzing and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action.

A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal.

Preliminary study is the process of gathering and interpreting facts, using the information for further studies on the system. Preliminary study is problem solving activity that requires intensive communication between the system users and system developers. It does various feasibility studies. In these studies, a rough figure of the system activities can be obtained, from which the decision about the strategies to be followed for effective system study and analysis can be taken.

2.2 EXISTING SYSTEM

In the existing system there is no good communication between administration and the car drivers. The older system is not user friendly. Administrator not able to view the requests of the users. Security is less compared to existing system. Details of the routes are not given online.

2.3 DRAWBACKS OF EXISTING SYSTEM

- Require to search vehicle manually.
- Less secure and take lot of time.

2.4 PROPOSED SYSTEM

The proposed system is user friendly. Good communication is maintained between admin and driver. All the user's requests can be viewed by the administrator 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.

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:

➤ **Better security: -**

For data to remain secure measures must be taken to prevent unauthorized access. Security means that data are protected from various forms of destruction. The system security problem can be divided into four related issues: security, integrity, privacy and confidentiality. Username and password requirement to sign in ensures security. It will also provide data security as we are using the secured databases for maintaining the documents.

➤ **Ensure data accuracy: -**

The proposed system eliminates the manual errors while entering the details of the users during the registration.

➤ **Better service: -**

The product will avoid the burden of hard copy storage. We can also conserve the time and human resources for doing the same task. The data can be maintained for longer period with no loss of data.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are:

3.1.1 Economical Feasibility

This study is carried out to check the economic impact will have on the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customised products have to be purchased.

3.1.2 Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes for the implementing this system.

3.1.3 Behavioral Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

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
Backend	-	MYSQL
Client on PC	-	Windows 7 and above.
Technologies used	-	JS, HTML5, PHP, CSS

3.3 SOFTWARE DESCRIPTION

3.3.1 PHP

PHP is a server side scripting language designed for web development but also used as a general purpose programming language. PHP is now installed on more than 244 million websites and 2.1 million web servers. Originally created by Rasmus Lerdorf in 1995, the reference implementation of PHP is now produced by the PHP group. While PHP originally stood for personal Home page ,it now stands for PHP:HypertextPreprocessor, a recursive acronym.PHP code is interpreted by a web server with a PHP processor module which generates the resulting web page.PHP commands can be embedded directly into a HTML source document rather than calling an external file to process data. It has also evolved to include a command-line interface capability and can be used in standalone incompatible with the GNU General Public License (GPL) due to restrictions on the usage of the term PHP.PHP can be deployed on most web servers and also as a standalone shell on almost every operating system and platform, free of charge.

3.3.2 MySQL

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation. The MySQL Web site provides the latest information about MySQL software.

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

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

- **MySQL databases are relational.**

A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one-to-one, one-to-many, unique, required or optional, and “pointers” between different tables. The database enforces these rules, so that with a well-designed database, your application never sees inconsistent, duplicate, orphan, out-of-date, or missing data. The SQL part of “MySQL” stands for “Structured Query Language”. SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax. SQL is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. In this manual, “SQL92” refers to the standard released in 1992, “SQL: 1999” refers to the standard released in 1999, and “SQL: 2003” refers to the current version of the standard. We use the phrase “the SQL standard” to mean the current version of the SQL Standard at any time.

- **MySQL software is Open Source.**

Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License), to define what you may and may not do with the software in different situations. If you feel uncomfortable with the GPL or need to embed MySQL code into a commercial application, you can buy a commercially licensed version from us. See the MySQL Licensing Overview for more information.

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

If that is what you are looking for, you should give it a try. MySQL Server can run comfortably on a desktop or laptop, alongside your other applications, web servers, and so on, requiring little or no attention. If you dedicate an entire machine to MySQL, you can adjust the settings to take advantage of all the memory, CPU power, and I/O capacity available.

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

The MySQL Database Software is a client/server system that consists of a multi-threaded SQL server that supports different backends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs). We also provide MySQL Server as an embedded multi-threaded library that you can link into your application to get a smaller, faster, easier-to-manage standalone product.

CHAPTER 4

SYSTEM DESIGN

4.1 INTRODUCTION

Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. The term “design” is defined as “the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization”. It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used. The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The design phase is a transition from a user oriented document to a document to the programmers or database personnel. System design goes through two phases of development: Logical and Physical Design.

4.2 UML DIAGRAM

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML was created by the Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997.

UML stands for **Unified Modeling Language**. UML is different from the other common programming languages such as C++, Java, COBOL, etc. UML is a pictorial language used to make software blueprints. UML can be described as a general-purpose visual modeling language to visualize, specify, construct, and document software system. Although UML is generally used to model software systems, it is not limited within this boundary. It is also used to model non-software systems as well. For example, the process flow in a manufacturing unit, etc. UML is not a programming language but tools can be used to generate code in various languages using UML diagrams. UML has a direct relation with object oriented analysis and design. After

some standardization, UML has become an OMG standard. All the elements, relationships are used to make a complete UML diagram and the diagram represents a system. The visual effect of the UML diagram is the most important part of the entire process. All the other elements are used to make it complete. UML includes the following nine diagrams.

- Class diagram
- Object diagram
- Use case diagram
- Sequence diagram
- Collaboration diagram
- Activity diagram
- Statechart diagram
- Deployment diagram
- Component diagram

4.2.1 USE CASE DIAGRAM

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are employed in UML (Unified Modeling Language), a standard notation for the modeling of real-world objects and systems.

System objectives can include planning overall requirements, validating a hardware design, testing and debugging a software product under development, creating an online help reference, or performing a consumer-service-oriented task. For example, use cases in a product sales environment would include item ordering, catalog updating, payment processing, and customer relations. A use case diagram contains four components.

- The boundary, which defines the system of interest in relation to the world around it.
- The actors, usually individuals involved with the system defined according to their

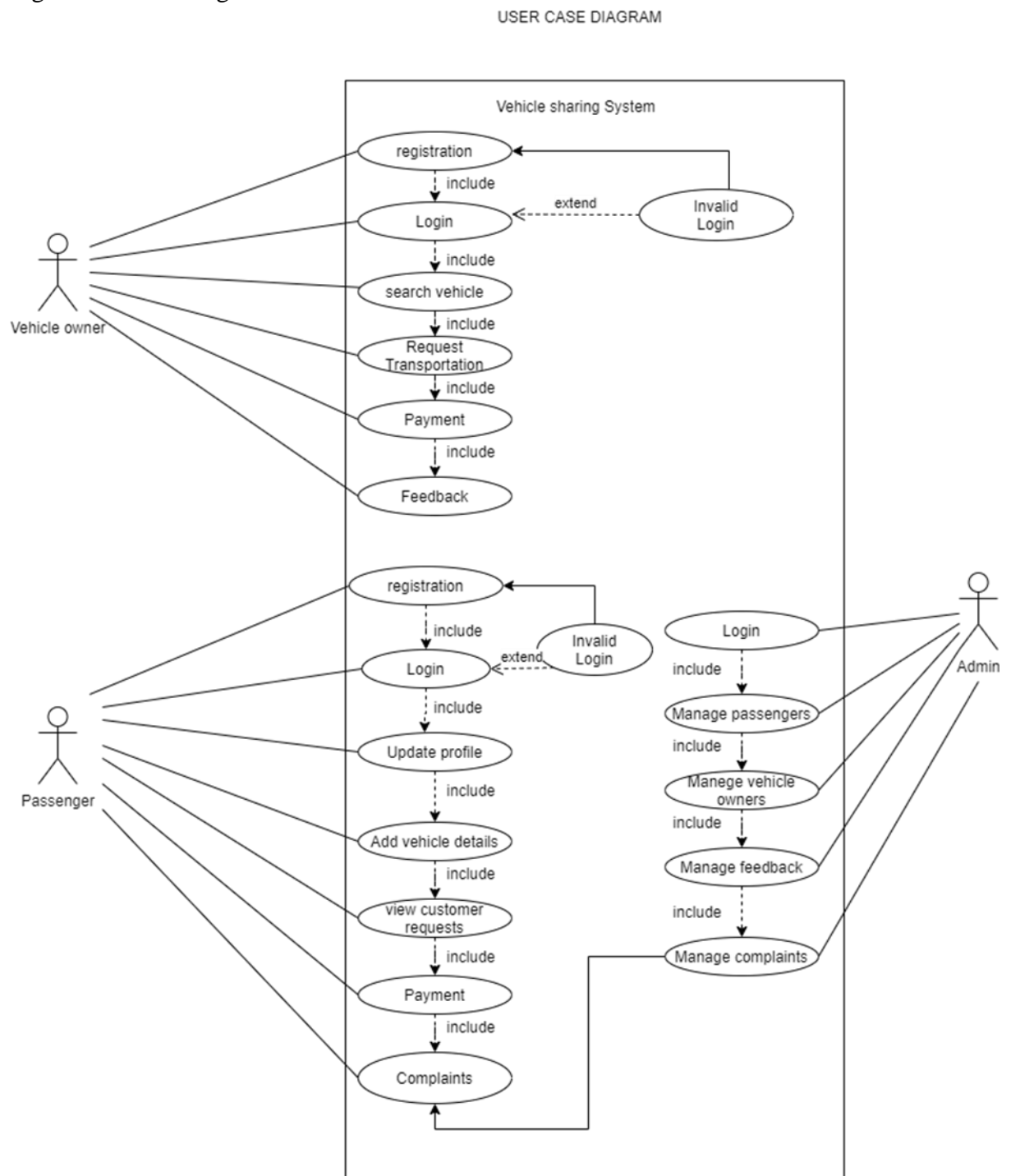
roles.

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

Use case diagrams are drawn to capture the functional requirements of a system. After identifying the above items, we have to use the following guidelines to draw an efficient use case diagram

- The name of a use case is very important. The name should be chosen in such a way so that it can identify the functionalities performed.
- Give a suitable name for actors.
- Show relationships and dependencies clearly in the diagram.
- Do not try to include all types of relationships, as the main purpose of the diagram is to identify the requirements.
- Use notes whenever required to clarify some important points.

Fig 1 : Use case diagram for E-care



4.2.2 SEQUENCE DIAGRAM

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

Sequence Diagram Notations –

- i. **Actors** – An actor in a UML diagram represents a type of role where it interacts with the system and its objects. It is important to note here that an actor is always outside the scope of the system we aim to model using the UML diagram. We use actors to depict various roles including human users and other external subjects. We represent an actor in a UML diagram using a stick person notation. We can have multiple actors in a sequence diagram.
- ii. **Lifelines** – A lifeline is a named element which depicts an individual participant in a sequence diagram. So basically each instance in a sequence diagram is represented by a lifeline. Lifeline elements are located at the top in a sequence diagram.
- iii. **Messages** – Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline. We represent messages using arrows. Lifelines and messages form the core of a sequence diagram.

Messages can be broadly classified into the following categories:

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

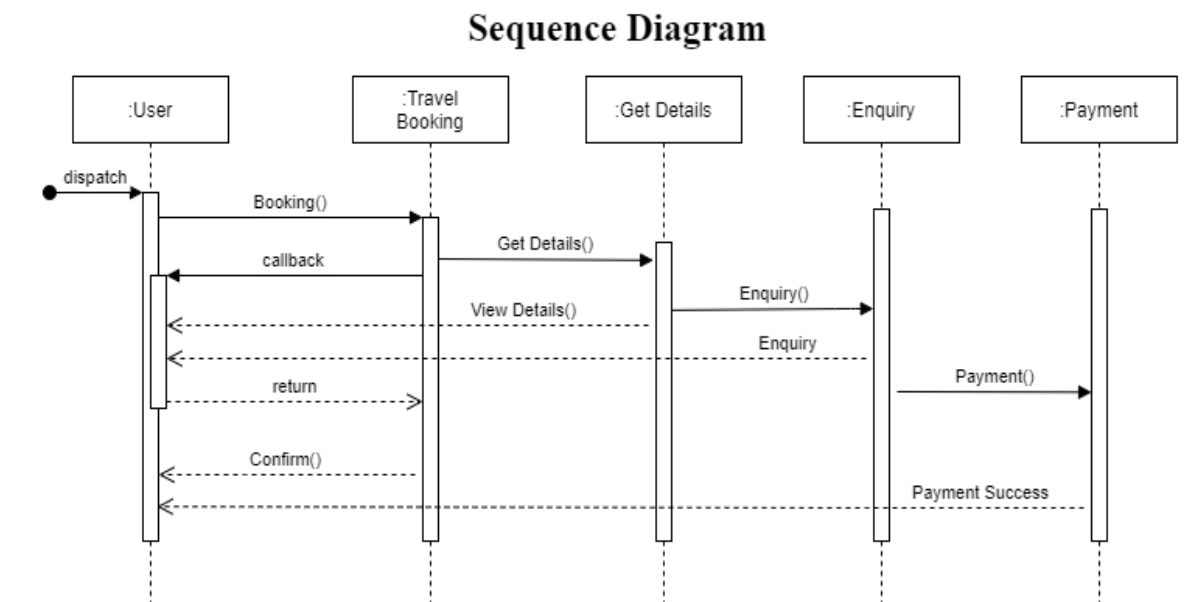
- Lost Message

- iv. **Guards** – To model conditions we use guards in UML. They are used when we need to restrict the flow of messages on the pretext of a condition being met. Guards play an important role in letting software developers know the constraints attached to a system or a particular process.

Uses of sequence diagrams –

- Used to model and visualize the logic behind a sophisticated function, operation or procedure.
- They are also used to show details of UML use case diagrams.
- Used to understand the detailed functionality of current or future systems.
- Visualise how messages and tasks move between objects or components in a system.

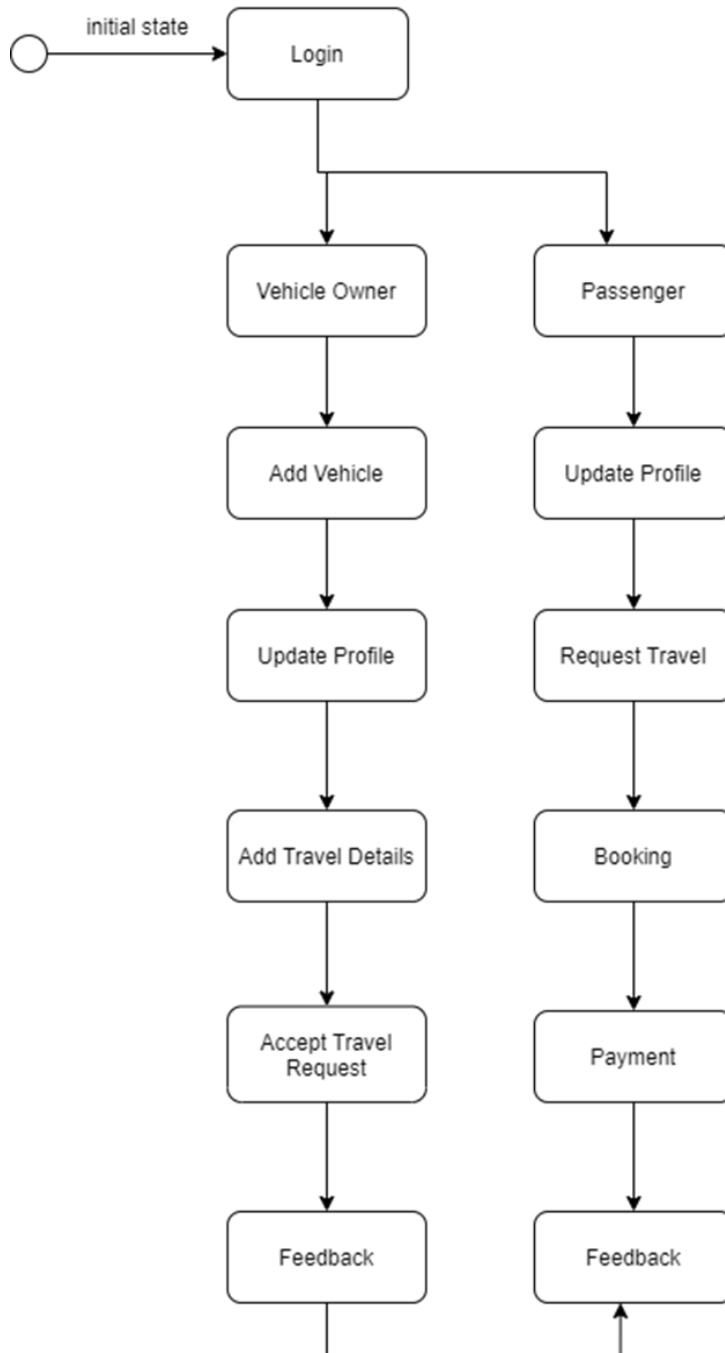
Fig 1 : Sequence diagram for E-care Construction Management System



4.2.3 State Chart diagram

State Diagram are used to capture the behavior of a software system. UML State machine diagrams can be used to model the behavior of a class, a subsystem, a package, or even an entire system. It is also called a State chart or State Transition diagram. State chart diagrams provide us an efficient way to model the interactions or communication that occur within the external entities and a system. These diagrams are used to model the event-based system. A state of an object is controlled with the help of an event. State chart diagrams are used to describe various states of an entity within the application system.

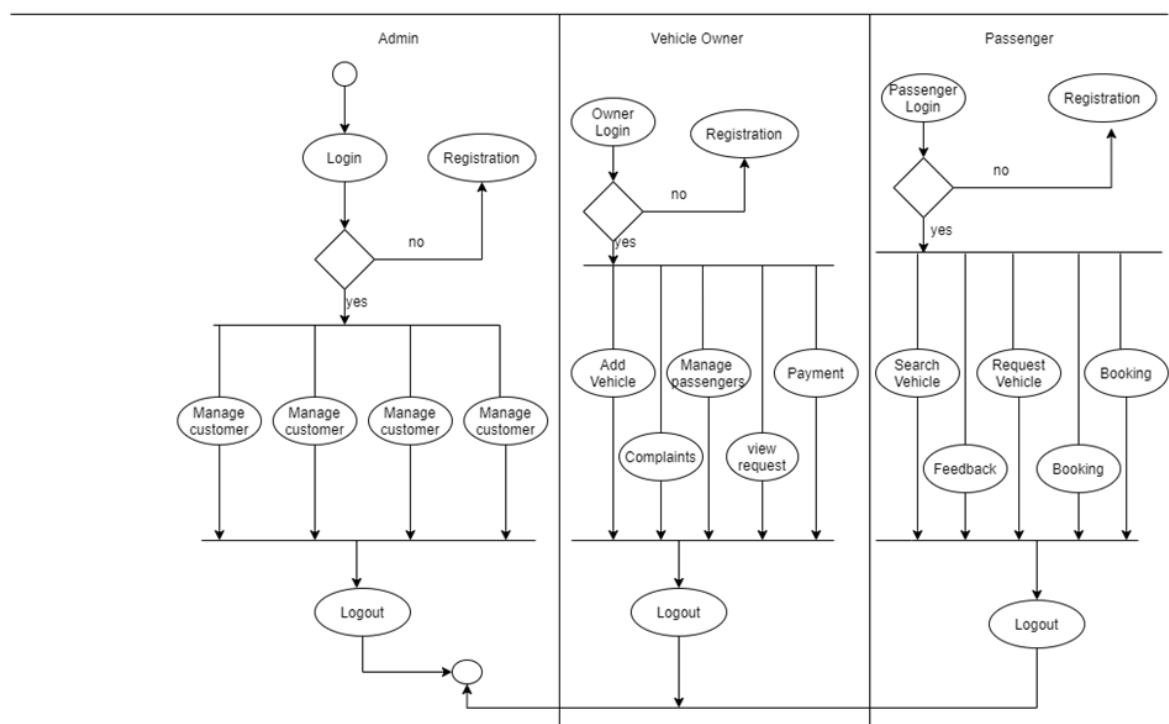
State Machine Diagram



4.2.4 Activity diagram

Activity Diagrams describe how activities are coordinated to provide a service which can be at different levels of abstraction. Typically, an event needs to be achieved by some operations, particularly where the operation is intended to achieve a number of different things that require coordination, or how the events in a single use case relate to one another, in particular, use cases where activities may overlap and require coordination. It is also suitable for modeling how a collection of use cases coordinates to represent business workflows.

Activity Diagram

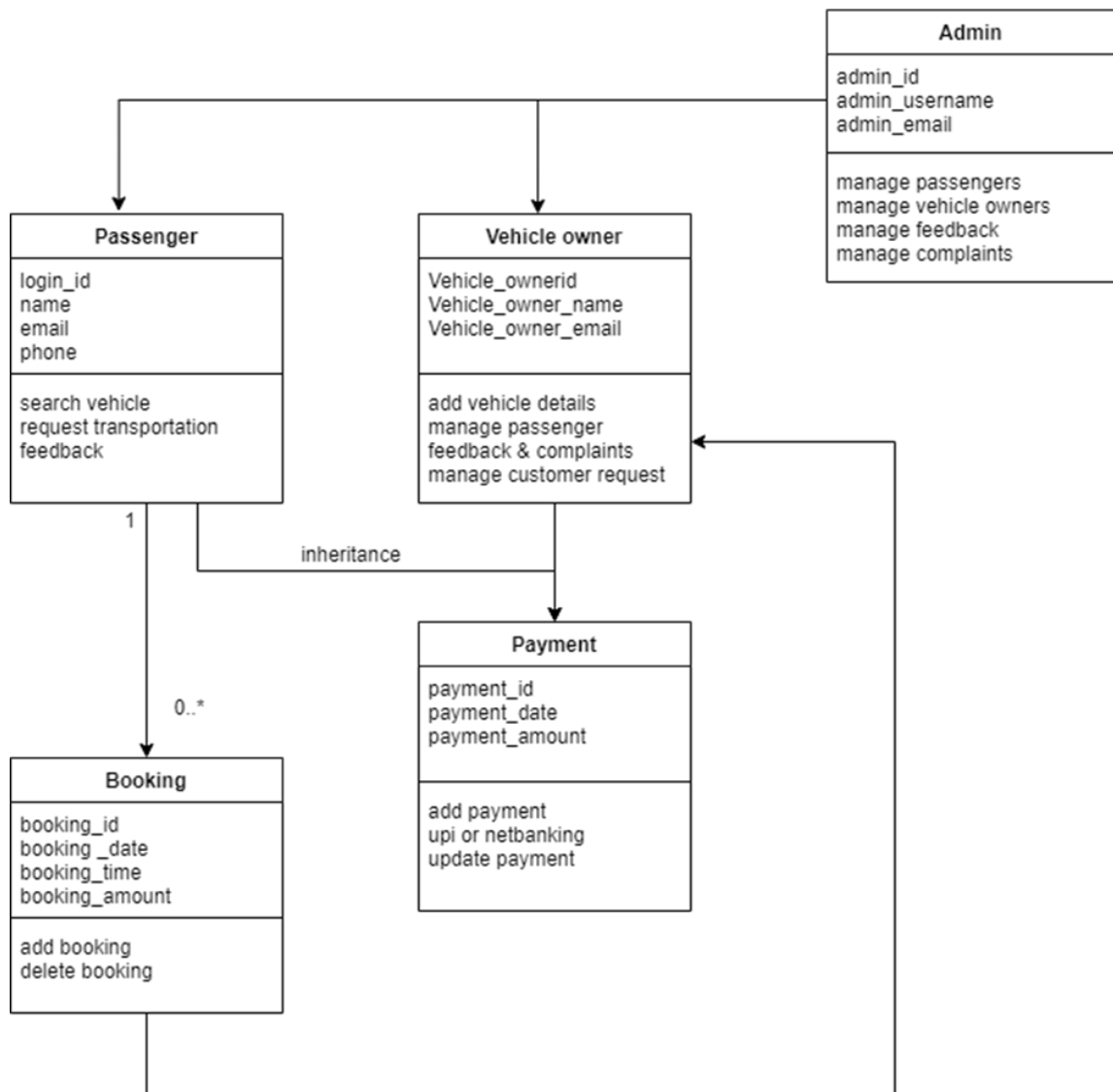


4.2.5 Class Diagram

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application. Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with

object-oriented languages. Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

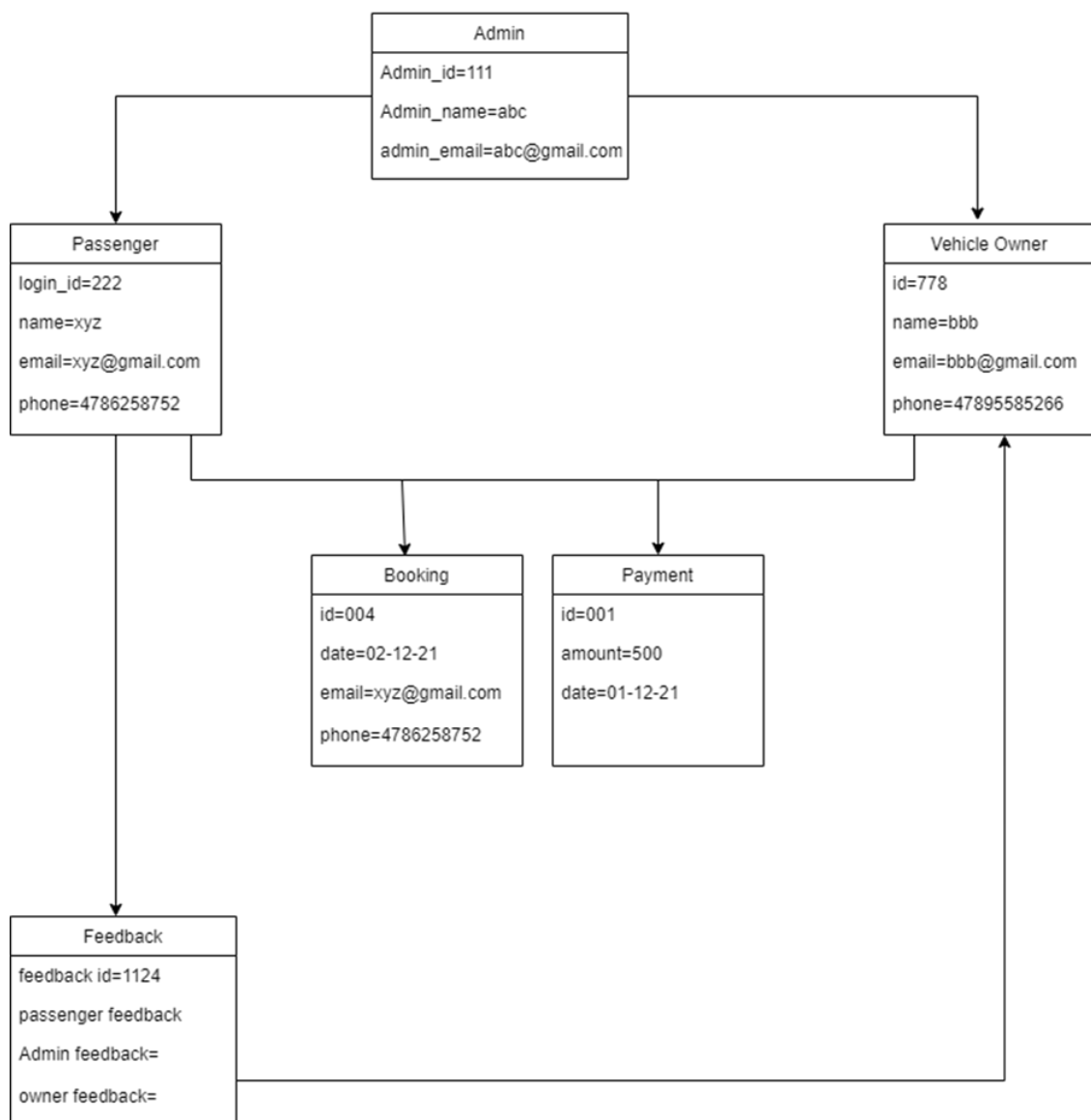
class diagram



4.2.6 Object Diagram

Object diagrams are derived from class diagrams so object diagrams are dependent upon class diagrams. Object diagrams represent an instance of a class diagram. The basic concepts are similar for class diagrams and object diagrams. Object diagrams also represent the static view of a system but this static view is a snapshot of the system at a particular moment. Object diagrams are used to render a set of objects and their relationships as an instance.

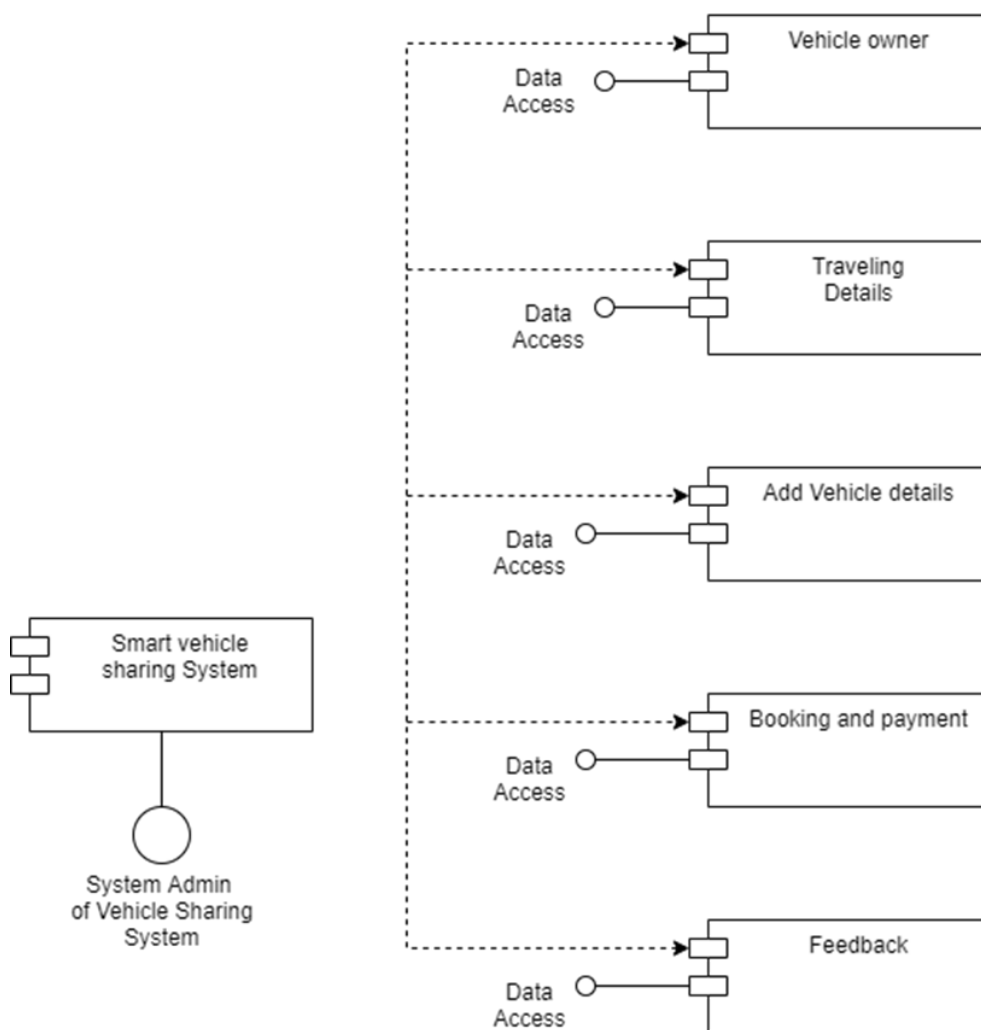
Object diagram



4.2.7 Component Diagram

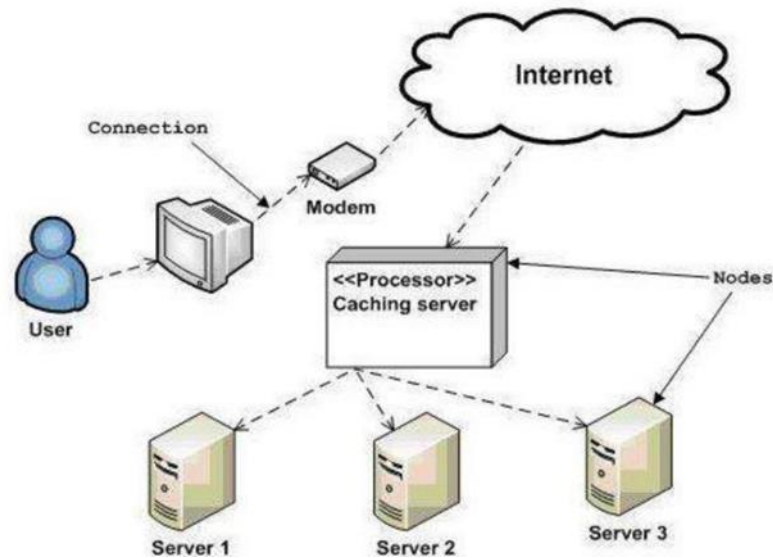
Component diagrams are different in terms of nature and behavior. Component diagrams are used to model the physical aspects of a system. Physical aspects are the elements such as executables, libraries, files, documents, etc. which reside in a node. Component diagrams are used to visualize the organization and relationships among components in a system. These diagrams are also used to make executable systems.

Component Diagram



4.2.8 Deployment Diagram

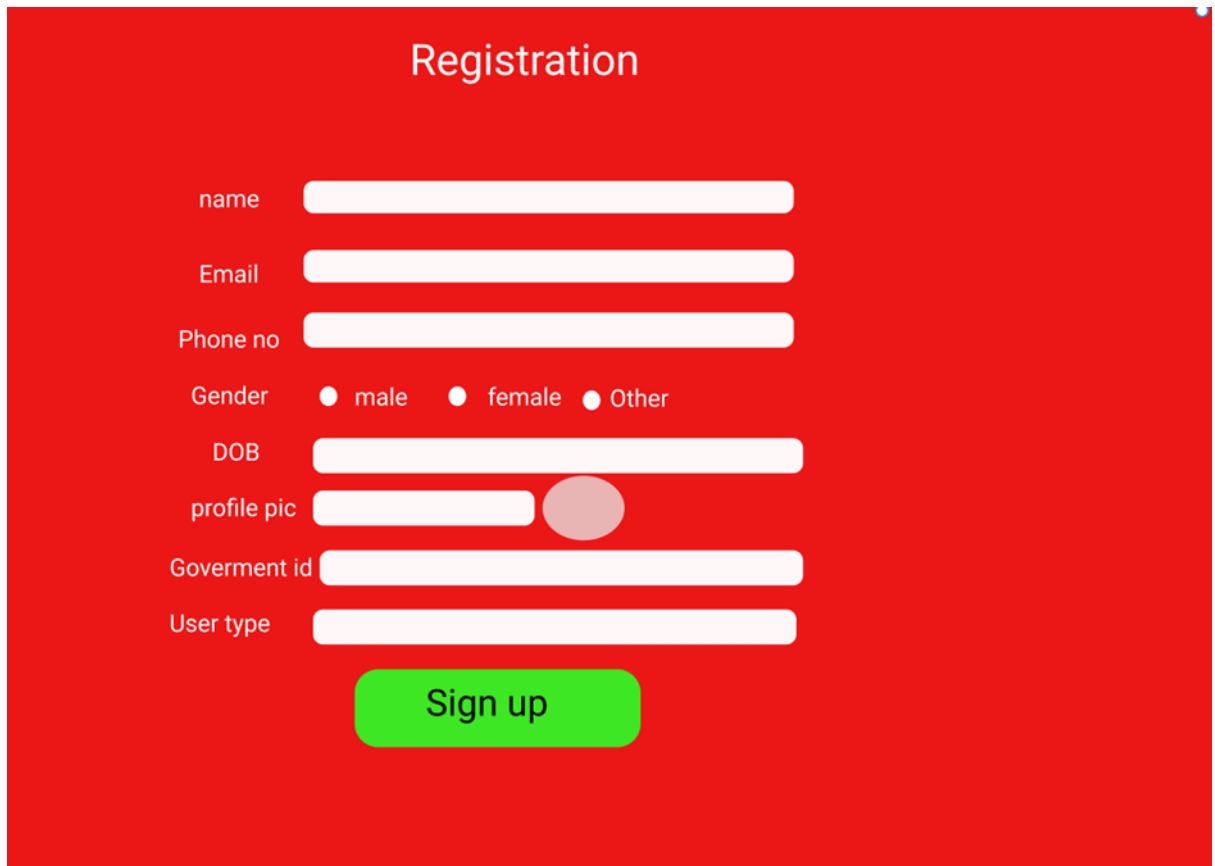
Deployment diagrams are used to visualize the topology of the physical components of a system, where the software components are deployed. Deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships.



4.3 USER INTERFACE DESIGN

4.3.1-INPUT DESIGN

Form Name : Registration



The image shows a registration form with a red background. The title 'Registration' is centered at the top in white. Below it, there are several input fields and a radio button group. The fields are labeled 'name', 'Email', 'Phone no', 'DOB', 'profile pic', 'Goverment id', and 'User type'. The 'Gender' field has three radio buttons labeled 'male', 'female', and 'Other'. A green 'Sign up' button is at the bottom.

Registration


name

Email

Phone no

Gender ☐ male ☐ female ☐ Other

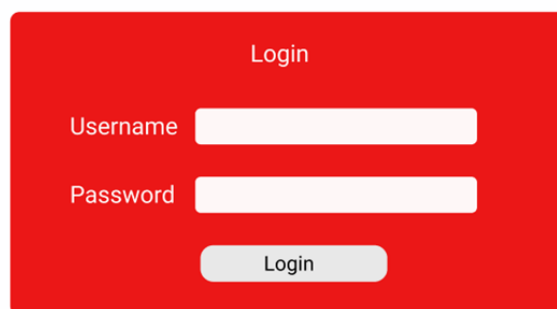
DOB

profile pic 

Goverment id

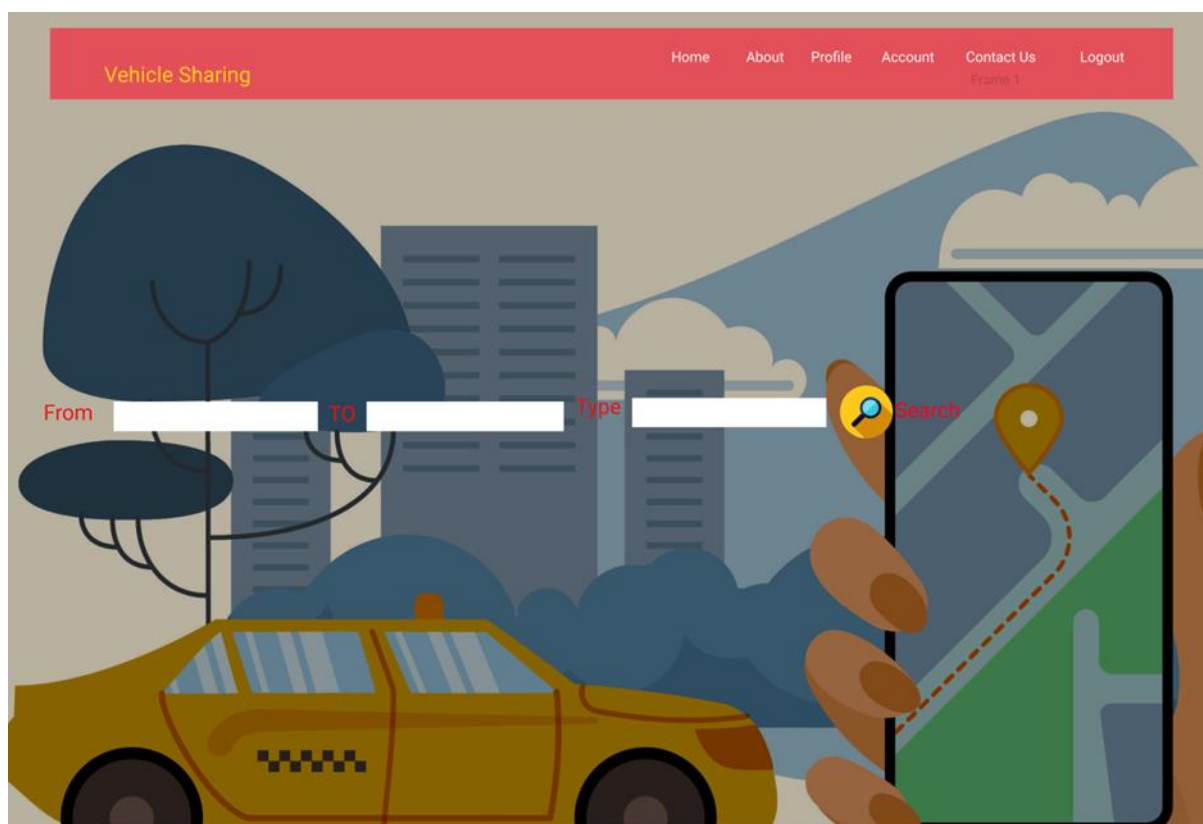
User type

Form Name : login

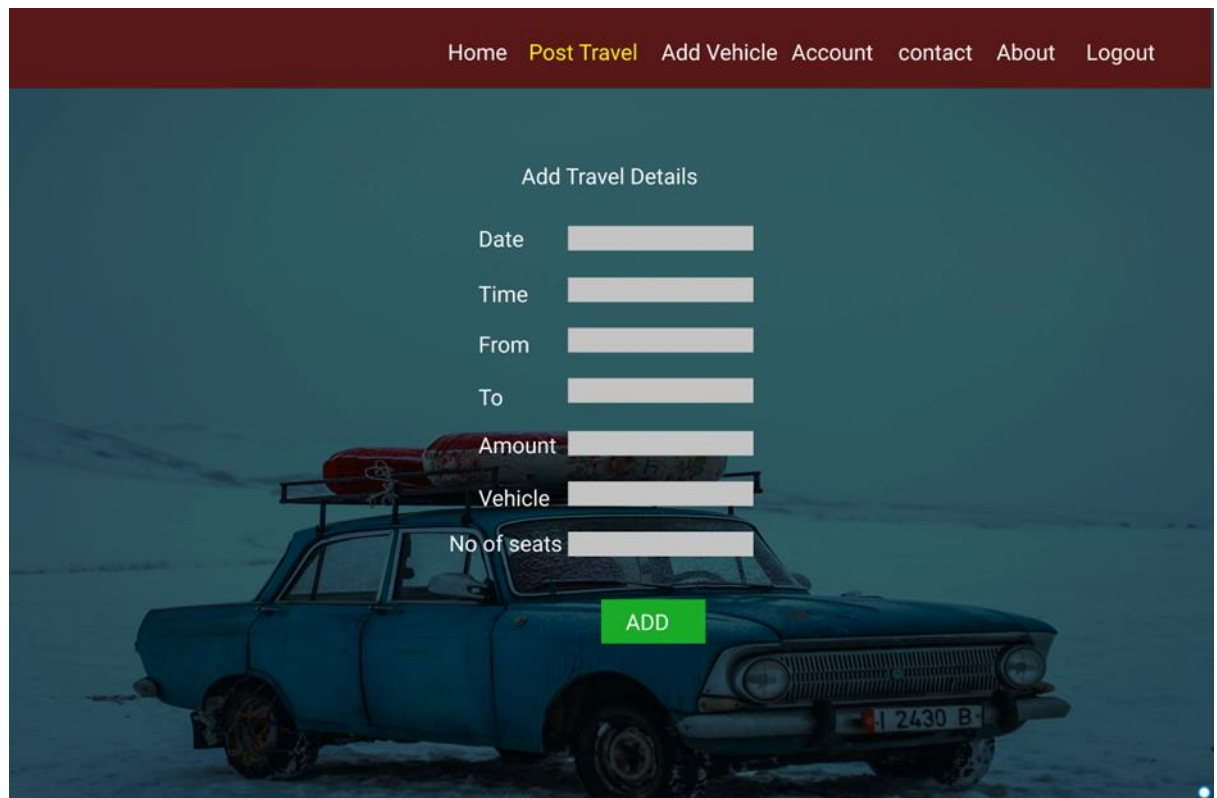


A red rectangular login form with rounded corners. At the top center is the text "Login" in white. Below it are two white input fields: the first is labeled "Username" and the second is labeled "Password". At the bottom center is a white button with the text "Login" in black.

Form Name : Users home page



Form Name : Add travel details



Home **Post Travel** Add Vehicle Account contact About Logout

Add Travel Details

Date

Time

From

To

Amount

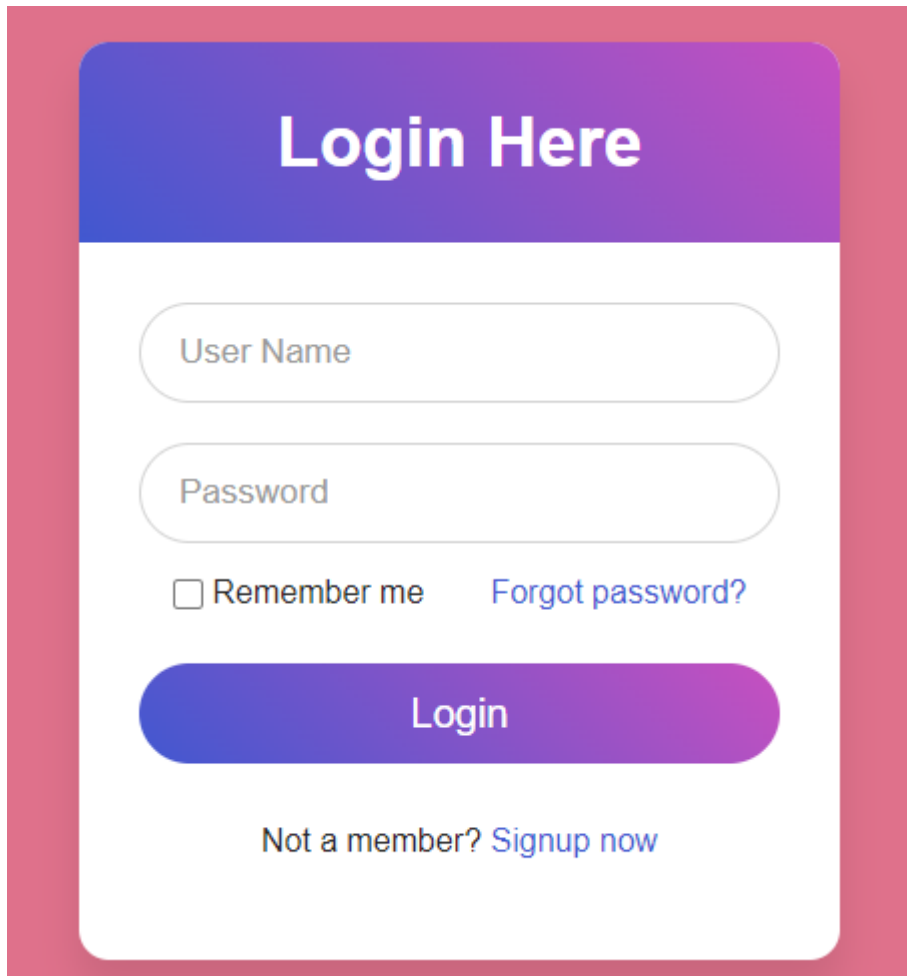
Vehicle

No of seats

ADD

4.3.2 OUTPUT DESIGN

Login page

The image shows a login page design. It features a central white card with rounded corners, set against a pink background. The card has a purple header with the text "Login Here" in white. Below the header, there are two input fields: "User Name" and "Password", both with rounded rectangular borders. Under the "Password" field, there is a checkbox labeled "Remember me" and a link "Forgot password?". Below these fields is a large, rounded purple button with the text "Login" in white. At the bottom of the card, there is a link "Not a member? Signup now".

Login Here

User Name

Password


☐ Remember me [Forgot password?](#)

Login

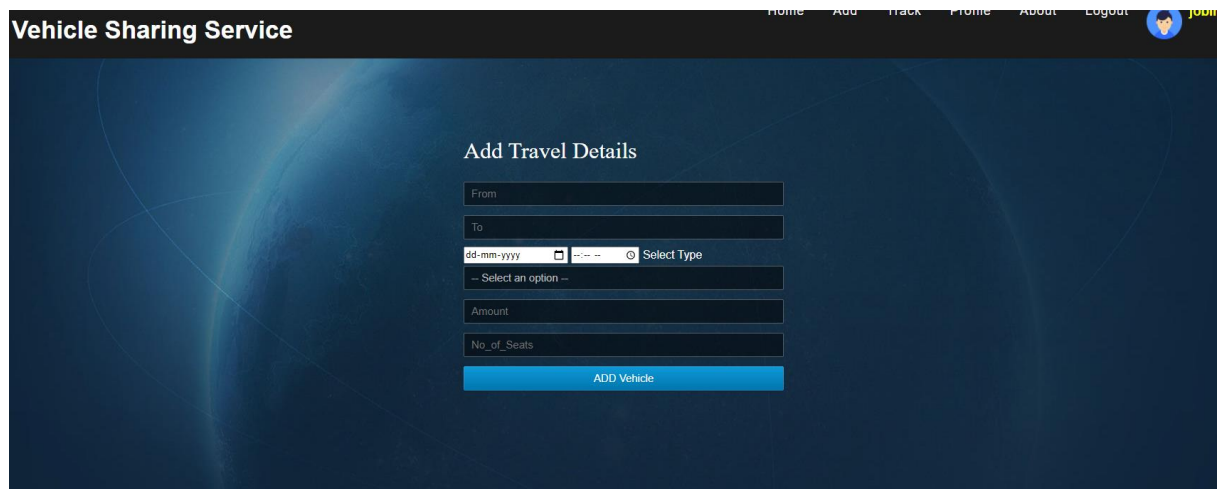
Not a member? [Signup now](#)

Registration page

Registration

Full Name	Username
<input type="text" value="Enter your name"/>	<input type="text" value="Enter your username"/>
Email	Phone Number
<input type="text" value="Enter your email"/>	<input type="text" value="Enter your number"/>
Place	State
<input type="text" value="Enter your number"/>	<input type="text" value="Enter your number"/>
pin	DOB
<input type="text" value="Enter your pin number"/>	<input type="text" value="dd-mm-yyyy"/> 
User Type	Adhar Number
<input type="text" value="Owner"/>	<input type="text" value="Enter your adhar number"/>
Password	Confirm Password
<input type="text" value="Enter your password"/>	<input type="text" value="Confirm your password"/>
Gender	
<input type="radio"/> Male <input type="radio"/> Female <input type="radio"/> Other	
profile pic <input type="button" value="Choose File"/> No file chosen	
<input type="button" value="Register"/>	
Already Login...?	

Add Travel details

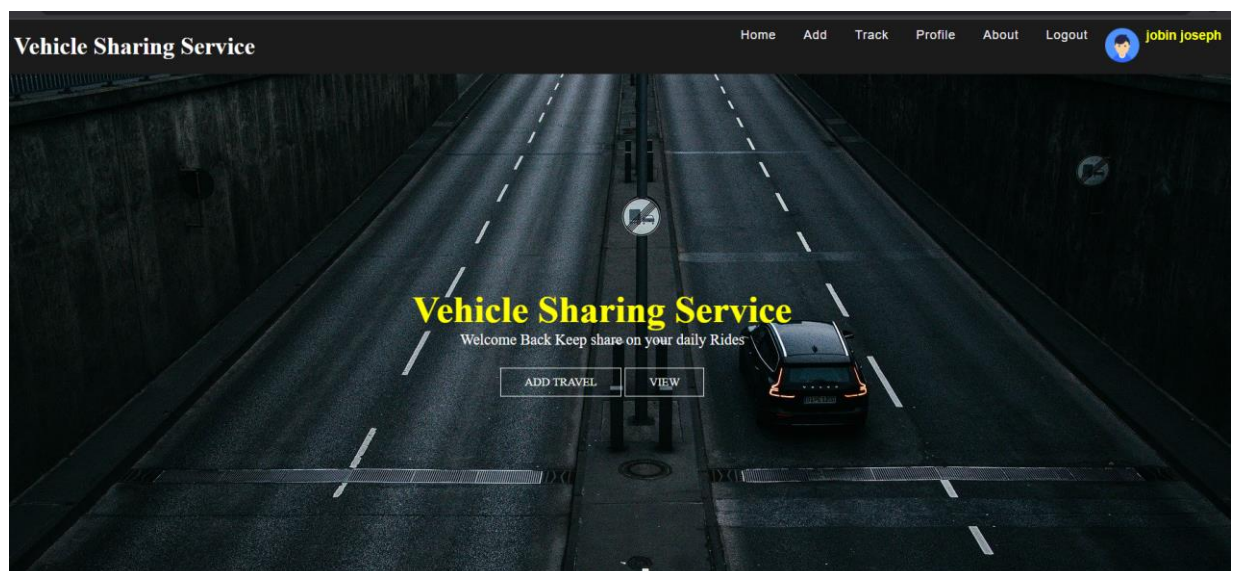


The screenshot shows the 'Add Travel Details' form within the 'Vehicle Sharing Service' application. The form is set against a dark blue background with a faint globe pattern. It includes the following fields and elements:

- From:** A text input field.
- To:** A text input field.
- dd-mm-yyyy**: A date input field with a calendar icon.
- --**: A time input field.
- Select Type**: A dropdown menu.
- Select an option --**: A placeholder text for the dropdown menu.
- Amount:** A text input field.
- No_of_Seats:** A text input field.
- ADD Vehicle**: A blue button at the bottom of the form.

The top navigation bar includes links for Home, Add, Track, Profile, About, and Logout, along with a user profile icon labeled 'jobin'.

Home page



4.4 DATABASE DESIGN

A database is an organized mechanism that has the capability of storing information through which a user can retrieve stored information in an effective and efficient manner. The data is the purpose of any database and must be protected.

The database design is a two level process. In the first step, user requirements are gathered together and a database is designed which will meet these requirements as clearly as possible. This step is called Information Level Design and it is taken independent of any individual DBMS.

In the second step, this Information level design is transferred into a design for the specific DBMS that will be used to implement the system in question. This step is called Physical Level Design, concerned with the characteristics of the specific DBMS that will be used. A database design runs parallel with the system design. The organization of the data in the database is aimed to achieve the following two major objectives.

- Data Integrity
- Data independence

4.6.1 Relational Database Management System (RDBMS)

A relational model represents the database as a collection of relations. Each relation resembles a table of values or file of records. In formal relational model terminology, a row is called a tuple, a column header is called an attribute and the table is called a relation. A relational database consists of a collection of tables, each of which is assigned a unique name. A row in a table represents a set of related values.

Relations, Domains & Attributes

A table is a relation. The rows in a table are called tuples. A tuple is an ordered set of n elements. Columns are referred to as attributes. Relationships have been set between every table in the database. This ensures both Referential and Entity Relationship Integrity. A domain D is a set of atomic values. A common method of specifying a domain is to specify a data type from which the data values forming the domain are drawn. It is also useful to specify a name for the domain to help in interpreting its values.

Every value in a relation is atomic, that is not decomposable.

Relationships

- Table relationships are established using Key. The two main keys of prime importance are Primary Key & Foreign Key. Entity Integrity and Referential Integrity Relationships can be established with these keys.
- Entity Integrity enforces that no Primary Key can have null values.
- Referential Integrity enforces that no Primary Key can have null values.
- Referential Integrity for each distinct Foreign Key value, there must exist a matching Primary Key value in the same domain. Other key are Super Key and Candidate Keys.

4.6.2 Normalization

Data are grouped together in the simplest way so that later changes can be made with minimum impact on data structures. Normalization is formal process of data structures in manners that eliminates redundancy and promotes integrity. Normalization is a technique of separating redundant fields and breaking up a large table into a smaller one. It is also used to avoid insertion, deletion, and updating anomalies. Normal form in data modelling use two concepts, keys and relationships. A key uniquely identifies a row in a table. There are two types of keys, primary key and foreign key. A primary key is an element or a combination of elements in a table whose purpose is to identify records from the same table. A foreign key is a column in a table that uniquely identifies record from a different table. All the tables have been normalized up to the third normal form.

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

The First Normal Form states that the domain of an attribute must include only atomic values and that the value of any attribute in a tuple must be a single value from the domain of that attribute. In other words 1NF disallows “relations within relations” or “relations as attribute values within tuples”. The only attribute values permitted by 1NF are single atomic or indivisible values. The first step is to put the data into First Normal Form. This can be done by moving data into separate tables where the data is of similar type in each table. Each table is given a Primary Key or Foreign Key as per requirement of the project. In this we form new relations for each non-atomic attribute or nested relation. This eliminated repeating groups of data. A relation is said to be in first normal form if only if it satisfies the constraints that contain the primary key only.

Second Normal Form

According to Second Normal Form, for relations where primary key contains multiple attributes, no non-key attribute should be functionally dependent on a part of the primary key. In this we decompose and setup a new relation for each partial key with its dependent attributes. Make sure to keep a relation with the original primary key and any attributes that are fully functionally dependent on it. This step helps in taking out data that is only dependent on a part of the key. A relation is said to be in second normal form if and only if it satisfies all the first normal form conditions for the primary key and every non-primary key attributes of the relation is fully dependent on its primary key alone.

Third Normal Form

According to Third Normal Form, Relation should not have a non-key attribute functionally determined by another non-key attribute or by a set of non-key attributes. That is, there should be no transitive dependency on the primary key. In this we decompose and set up relation that includes the non-key attributes that functionally determines other non-key attributes. This step is taken to get rid of anything that does not depend entirely on the Primary Key. A relation is said to be in third normal form if only if it is in second normal form and more over the non key attributes of the relation should not be depend on other non-key attribute.

TABLE DESIGN**Tbl_registration**

Field	Datatype	Constraint	Description
Uid	INT	Primary key	User id
Name	VARCHAR	Not null	User name
email	VARCHAR	Not null	User email
gender	VARCHAR	Not null	gender
DOB	DATE	Not null	DOB
image	VARCHAR	Not null	Image path
Gov_id_No	Varchar	Not null	Verification
Phone no	INT	Not null	Phone no
User_Type	INT	Not null	User type

Tbl_login

Field	Datatype	Constraint	Description
Login_id	INT	Primary key	Login id
Uid	INT	Foreign Key	Reference
username	Varchar	Not null	Username
Password	Varchar	Not null	password
Status	INT	Not null	User Status

Tbl_addVehicle

Field	Datatype	Constraint	Description
Vid	INT	Primary Key	Vehicle id
Login_id	INT	Foreign Key	Login id
Model	VARCHAR	Not null	Vehicle model
Type	VARCHAR	Not null	Type
Build Year	VARCHAR	Not null	Build Year
No_of_seats	INT	Not null	No_of_seats
Vehicle_no	INT	Not null	Vehicle_no
Image	Varchar	Not null	Image path

Tbl_Travel_Details

Field	Datatype	Constraint	Description
Travel_id	INT	Primary key	Travel id
Vid	INT	Foreign Key	vid
Date	Date	Not null	Date
Time	VARCHAR	Not null	Time
From	VARCHAR	Not null	From
To	VARCHAR	Not null	To
Amount	INT	Not null	Amount

Tbl_booking details

Field	Datatype	Constraints	Description
Bid	INT	Primary Key	Bid
Uid	INT	Foreign Key	Uid
Travel_id	INT	Foreign Key	travelid
Payment status	INT	Not null	Payment Status

Tbl_payment details

Field	Datatype	Constraints	Description
Payment_id	INT	Primary Key	Payment id
Login_id	INT	Foreign Key	Login id
Travel_id	INT	Foreign Key	travelid
Payment status	INT	Not null	Payment Status

Tbl_Feedback

Field	Datatype	Constraints	Description
Feedback_id	INT	Primary Key	Feedback id
Login_id	INT	Foreign Key	Log id
Feedback	Varchar	Not null	Feedback

CHAPTER 5

SYSTEM TESTING

5.1 INTRODUCTION

Software Testing is the process of executing software in a controlled manner, in order to answer the question - Does the software behave as specified? Software testing is often used in association with the terms verification and validation. Validation is the checking or testing of items, includes software, for conformance and consistency with an associated specification. Software testing is just one kind of verification, which also uses techniques such as reviews, analysis, inspections, and walkthroughs. Validation is the process of checking that what has been specified is what the user actually wanted.

Other activities which are often associated with software testing are static analysis and dynamic analysis. Static analysis investigates the source code of software, looking for problems and gathering metrics without actually executing the code. Dynamic analysis looks at the behavior of software while it is executing, to provide information such as execution traces, timing profiles, and test coverage information.

Testing is a set of activity that can be planned in advanced and conducted systematically. Testing begins at the module level and work towards the integration of entire computers-based system. Nothing is complete without testing, as it vital success of the system testing objectives, there are several rules that can serve as testing objectives. 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 testing is conducted successfully according to the objectives as stated above, it would uncover errors in the software. Also testing demonstrate that the software function appear to be working according to the specification, that performance requirement appear to have been met.

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 implies a series of desired course of action to be followed in accomplishing various testing methods. The Test Plan acts as a blue print for the action that is to be followed. The software engineers create a computer program, its documentation and related data structures. The software developers is always responsible for testing the individual units of the programs, ensuring that each performs the function for which it was designed. There is an independent test group (ITG) which is to remove the inherent problems associated with letting the builder to test the thing that has been built. The specific objectives of testing should be stated in measurable terms. So that the mean time to failure, the cost to find and fix the defects, remaining defect density or frequency of occurrence and test work-hours per regression test all should be stated within the test plan.

The levels of testing include:

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

5.2.1 Unit Testing

Unit testing focuses verification effort on the smallest unit of software design – the software component or module. Using the component level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and uncovered scope established for unit testing. The unit testing is white-box oriented, and step can be conducted in parallel for multiple components. The modular interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm's execution. Boundary conditions are tested to ensure that all statements in a module have been executed at least once. Finally, all error handling paths are tested.

Tests of data flow across a module interface are required before any other test is initiated. If data do not enter and exit properly, all other tests are moot. Selective testing of execution paths is an essential task during the unit test. Good design dictates that error conditions be anticipated and error handling paths set up to reroute or cleanly terminate processing when an error does occur. Boundary testing is the last task of unit testing step. Software often fails at its boundaries.

Unit testing was done in Sell-Soft System by treating each module as separate entity and testing each one of them with a wide spectrum of test inputs. Some flaws in the internal logic of the modules were found and were rectified. After coding each module is tested and run individually. All unnecessary code were removed and ensured that all modules are working, and gives the expected result.

5.2.2 Integration Testing

Integration testing is systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design. The entire program is tested as whole. Correction is difficult because isolation of causes is complicated by vast expanse of entire program. Once these errors are corrected, new ones appear and the process continues in a seemingly endless loop. After performing unit testing in the System all the modules were integrated to test for any inconsistencies in the interfaces. Moreover differences in program structures were removed and a unique program structure was evolved.

5.2.3 Validation Testing or System Testing

This is the final step in testing. In this the entire system was tested as a whole with all forms, code, modules and class modules. This form of testing is popularly known as Black Box testing or System tests.

Black Box testing method focuses on the functional requirements of the software. That is, Black Box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program.

Black Box testing attempts to find errors in the following categories; incorrect or missing functions, interface errors, errors in data structures or external data access, performance errors and initialization errors and termination errors.

5.2.4 Output Testing or User Acceptance Testing

The system considered is tested for user acceptance; here it should satisfy the firm's need.

The software should keep in touch with perspective system; user at the time of developing and making changes whenever required. This done with respect to the following points:

- Input Screen Designs,
- Output Screen Designs,

The above testing is done taking various kinds of test data. Preparation of test data plays a vital role in the system testing. After preparing the test data, the system under study is tested using that test data. While testing the system by which test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

CHAPTER 6

IMPLEMENTATION

6.1 INTRODUCTION

Implementation is the stage of the project where the theoretical design is turned into a working system. It can be considered to be the most crucial stage in achieving a successful new system gaining the users confidence that the new system will work and will be effective and accurate. It is primarily concerned with user training and documentation. Conversion usually takes place about the same time the user is being trained or later. Implementation simply means convening a new system design into operation, which is the process of converting a new revised system design into an operational one.

At this stage the main work load, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned or controlled, it can create chaos and confusion.

Implementation includes all those activities that take place to convert from the existing system to the new system. The new system may be a totally new, replacing an existing manual or automated system or it may be a modification to an existing system. Proper implementation is essential to provide a reliable system to meet organization requirements. The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from the old system to the new system. The system can be implemented only after through testing is done and if it is found to be working according to the specifications. The system personnel check the feasibility of the system. The more complex the system being implemented, the more involved will be the system analysis and design effort required to implement the three main aspects: education and training, system testing and changeover.

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

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended uses and the operation of the system. In many organizations someone who will not be operating it, will commission the software development project. In the initial stage people doubt about the software but we have to

ensure that the resistance does not build up, as one has to make sure that:

The active user must be aware of the benefits of using the new system. Their confidence in the software is built up.

Proper guidance is imparted to the user so that he is comfortable in using the application.

Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not up running on the server, the actual process won't take place.

6.2.1 User Training

User training is designed to prepare the user for testing and converting the system. To achieve the objective and benefits expected from computer-based system, it is essential for the people who will be involved to be confident of their role in the new system. As system becomes more complex, the need for training is more important. By user training the user comes to know how to enter data, respond to error messages, interrogate the database and call up routine that will produce reports and perform other necessary functions.

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

Maintenance is the enigma of system development. The maintenance phase of the software cycle is the time in which a software product performs useful work. After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle. The need for system maintenance is for it to make adaptable to the changes in the system environment. Software maintenance is of course, far more than "Finding Mistakes".

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

The current system working technology is old fashioned and there is no usage of commonly used technologies like internet, digital money. 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.

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:

- www.w3schools.com
- www.jquery.com
- <https://www.lovelycoding.org/construction-work-management-system/>
- <https://www.slideshare.net/chiragbarasiya/construction-management-system-final-year-report>
- <https://app.diagrams.net>
- <http://homepages.dcc.ufmg.br/~rodolfo/es-1-03/IEEE-Std-830-1998.pdf>
- www.agilemodeling.com/artifacts/useCaseDiagram.html

CHAPTER 9

APPENDIX

9.1 Sample Code

Login.php

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

<html lang="en" dir="ltr">
<head>
<meta charset="utf-8">
<!----<title>Login Form Design | CodeLab</title>---->
<link rel="stylesheet" href="stylelog.css">

</head>
<body>

<div class="wrapper">
<div class="title">Login Here</div>
<form name = "myForm" action="log.php" method="POST" enctype="multipart/form-data">
<div class="field">
<input type="text" name="username" required >
<label>User Name</label>
</div>
<div class="field">
<input type="password" name="password" required>
<label>Password</label>
</div>
<div class="content">
<div class="checkbox">
<input type="checkbox" id="remember-me">
<label for="remember-me">Remember me</label>
```

```
</div>

<div class="pass-link"><a href="#">Forgot password?</a></div>

</div>

<div class="field">

<input type="submit" value="Login" name="Click1">

</div>

<div class="signup-link">Not a member? <a href="reg.php">Signup now</a></div>

</form>

</div>

</body>

</html>
```

```
<?php
include("connect.php");

if(isset($_POST["Click1"])){

$username=$_POST["username"];
$password=$_POST["password"];

$sql="select * from registration where username='$username' && password='$password'";

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

if ($result->num_rows > 0) {
while($row = $result->fetch_assoc()) {
$_SESSION['uid']= $row["userid"];
$_SESSION['nnm']= $row["fullname"];
$_SESSION['prof']= $row["propath"];
$typ=$row["usertype"];
$st=$row["status"];
```

```
}}
```

```
$count=mysqli_num_rows($result);
```

```
if($count>0)
```

```
{
```

```
if($typ==1)
```

```
{
```

```
if($st==1){
```

```
?>
```

```
<script>
```

```
alert("You are blocked by admin");
```

```
</script>
```

```
<?php
```

```
}
```

```
else{
```

```
header("location:ho.php");
```

```
}
```

```
}
```

```
else if($typ==2)
```

```
{
```

```
if($st==1)
```

```
{
```

```
?>
```

```
<script>
```

```
alert("You are blocked by admin");
```

```
</script>
```

```
<?php
```

```
}
```

```
else{
```

```
header("location:home.php");
```

```
}
```

```
}

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

}

mysqli_close($con);

}
?>
```

Registraion.php

```
<?php
include 'connect.php';
//onsubmit = "return(validate());"
?>

<html lang="en" dir="ltr">
<head>
<meta charset="UTF-8">
<!--<title> Responsive Registration Form | CodingLab </title>-->
<link rel="stylesheet" href="style.css">
<meta name="viewport" content="width=device-width, initial-scale=1.0">

<script type = "text/javascript">

function validate() {
var password1=document.myForm.pass.value;
var password2=document.myForm.conpass.value;
```

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

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

if (document.myForm.pin.value=="")
{
document.getElementById('pi').innerHTML="**Enter Pin";
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.dob.value=="")
{
document.getElementById('do').innerHTML="**Enter Dob";
return false;
}
if (document.myForm.adhar.value=="")
{
document.getElementById('ad').innerHTML="**Enter Adhar";
return false;
}
var valadhar = document.myForm.adhar.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.pass.value=="")
{
document.getElementById('pa').innerHTML="**Enter Password";
return false;
}
if (document.myForm.conpass.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>

<div class="container">
<div class="title">Registration</div>
<div class="content">
<form name = "myForm" action="reg.php" method="POST" onsubmit = "return(validate());"
enctype="multipart/form-data">
<div class="user-details">
<div class="input-box">
<span class="details">Full Name <span id="fn" style="color:red;"></span></span>
<input type="text" placeholder="Enter your name" name="fullname">
</div>
<div class="input-box">
<span class="details">Username <span id="un" style="color:red;"></span></span>
<input type="text" name="uname" placeholder="Enter your username" >
</div>

```

```

<div class="input-box">
<span class="details">Email <span id="em" style="color:red;"></span>
<input type="text" placeholder="Enter your email" name="email">
</div>
<div class="input-box">
<span class="details">Phone Number <span id="pn" style="color:red;"></span>
<input type="text" placeholder="Enter your number" name="phone">
</div>
<div class="input-box">
<span class="details">Place <span id="pl" style="color:red;"></span>
<input type="text" placeholder="Enter your number" name="place" >
</div>
<div class="input-box">
<span class="details">State <span id="st" style="color:red;"></span>
<input type="text" placeholder="Enter your number" name="state">
</div>
<div class="input-box">
<span class="details">pin <span id="pi" style="color:red;"></span>
<input type="text" placeholder="Enter your pin number" name="pin">
</div>
<div class="input-box">
<span class="details">DOB <span id="do" style="color:red;"></span>
<input type="date" placeholder="Enter your DOB" name="dob" max="<?= date('Y-m-d'); ?>">

</div>
<div class="input-box">
<span class="details">User Type</span>

<select class="input-box" name="usertype">
<option value="1" >Owner</option>
<option value="2" >User</option></select>

</div>
<div class="input-box">
<span class="details">Adhar Number <span id="ad" style="color:red;"></span>
<input type="text" placeholder="Enter your adhar number" name="adhar">
</div>
<div class="input-box">
<span class="details">Password <span id="pa" style="color:red;"></span>
<input type="password" placeholder="Enter your password" name="pass">
</div>
<div class="input-box">
<span class="details">Confirm Password <span id="cpa" style="color:red;"></span>
<input type="password" placeholder="Confirm your password" name="conpass">
</div>
</div>
<div class="gender-details">
<input type="radio" name="gender" id="dot-1" value="male">
<input type="radio" name="gender" id="dot-2" value="female">
<input type="radio" name="gender" id="dot-3" value="other">
<span class="gender-title">Gender</span>
<div class="category">
<label for="dot-1">
<span class="dot one"></span>
<span class="gender">Male</span>
</label>

```

```

<label for="dot-2">
<span class="dot two"></span>
<span class="gender">Female</span>
</label>
<label for="dot-3">
<span class="dot three"></span>
<span class="gender">Other</span>
</label>
</div>
</div>
<div><label>profile pic </label>
<input type="file" name="imgpath" accept="image/*" value="" required /></div>
<div class="button">
<input type="submit" value="Register" name="Click1">
</div>
<div>
<center> <h2 class="linker" ><a href="log.php" style="text-decoration: none;">Already
Login...?</a></h2></center>
</div>

</form>
</div>
</div>

</body>
</html>

```

```

<?php
include("connect.php");

if(isset($_POST["Click1"]))
{
$fullname=$_POST["fullname"];
$username=$_POST["uname"];
$email=$_POST["email"];
$phone=$_POST["phone"];
$place=$_POST["place"];
$state=$_POST["state"];
$pin=$_POST["pin"];
$DOB=$_POST["dob"];
$usertype=$_POST["usertype"];
$adhar=$_POST["adhar"];
$password=$_POST["pass"];
$gender=$_POST["gender"];

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

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

$filename = $_FILES["imgpath"]["name"];

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

$folder = "propic/".$filename;

```

```

if (move_uploaded_file($tempname, $folder)) {

$msg = "Image uploaded successfully";

}else{

$imgpath="user.png";
}
$sql="insert into
registration(fullname,username,email,phonno,place,state,pin,dob,usertype,adhar,password,gender,propa
th)
values('$fullname','$username','$email',
$phone','$place','$state','$pin','$DOB','$usertype','$adhar','$password','$gender','$imgpath')";
if(mysqli_query($con,$sql))
{
?>
<script>
alert("inserted");
</script>
<?php
}
else
{
echo "error";
}

mysqli_close($con);

}
?>

```

Addtravel.php

```

<?php
include 'nav.php';
?>
<html lang="en" dir="ltr">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<!--<title> Registration or Sign Up form in HTML CSS | CodingLab </title>-->

<link
href="//db.onlinewebfonts.com/c/a4e256ed67403c6ad5d43937ed48a77b?family=Core+Sans+N+
W01+35+Light" rel="stylesheet" type="text/css"/>
<style type="text/css">
*{
margin:0;
padding:0;
}

```

```
h1 {  
  font-size: 2em;  
  font-family: "Core Sans N W01 35 Light";  
  font-weight: normal;  
  margin: .67em 0;  
  display: block;  
}
```

```
#registered {  
  margin-top: 50px;  
}
```

```
#registered img {  
  margin-bottom: 0px;  
  width: 100px;  
  height: 100px;  
}
```

```
#registered span {  
  clear: both;  
  display: block;  
}
```

```
img {  
  margin-bottom: 20px;  
}
```

```
.avatar {  
  margin: 10px 0 20px 0;  
}
```

```
.module{  
  position:relative;  
  top:10%;  
  height:65%;  
  width:450px;  
  margin-left:auto;  
  margin-right:auto;  
}
```

```
.user {  
  color: #66d8fc;  
  font-weight: bold;  
}
```

```
.userlist {  
  float:left;  
  padding: 30px;  
}
```

```
.userlist span {  
  color: #0590fc;  
}
```

```
.welcome{
```

```
position:relative;
top:30%;
height:65%;
width:900px;
margin-left:auto;
margin-right:auto;
margin-top: 50px;
}

::-moz-selection {
background: #19547c;
}
::selection {
background: #19547c;
}
input::-moz-selection {
background: #037db6;
}
input::selection {
background: #037db6;
}

body{
color: #fff;
background-color:#f0f0f0;
font-family:helvetica;
background:url('http://clevertechie.com/img/bnet-bg.jpg') #0f2439 no-repeat center top;
}

.body-content{
position:relative;
top:20px;
height:700px;
width:800px;
margin-left:auto;
margin-right:auto;
background: transparent;
}

select,
textarea,
input[type="text"],
input[type="password"],
input[type="email"]
{
height:30px;
width:100%;;
display: inline-block;
vertical-align: middle;
height: 34px;
padding: 0 10px;
margin-top: 3px;
margin-bottom: 10px;
font-size: 15px;
line-height: 20px;
border: 1px solid rgba(255, 255, 255, 0.3);
```

```
background-color: rgba(0, 0, 0, 0.5);
color: rgba(255, 255, 255, 0.7);
-moz-box-sizing: border-box;
box-sizing: border-box;
border-radius: 2px;
}

select,
textarea,
input[type="text"],
input[type="password"],
input[type="email"] {
-mwebkit-appearance: none;
-moz-appearance: none;
-ms-appearance: none;
appearance: none;
-mwebkit-transition: background-position 0.2s, background-color 0.2s, border-color 0.2s, box-shadow 0.2s;
transition: background-position 0.2s, background-color 0.2s, border-color 0.2s, box-shadow 0.2s;
}
select:hover,
textarea:hover,
input[type="text"]:hover,
input[type="password"]:hover,
input[type="email"]:hover {
border-color: rgba(255, 255, 255, 0.5);
background-color: rgba(0, 0, 0, 0.5);
color: rgba(255, 255, 255, 0.7);
}
select:focus,
textarea:focus,
input[type="text"]:focus,
input[type="password"]:focus,
input[type="email"]:focus {
border: 2px solid;
border-color: #1e5f99;
background-color: rgba(0, 0, 0, 0.5);
color: #ffffff;
}
.btn {
text-overflow: ellipsis;
white-space: nowrap;
overflow: hidden;
margin: 3px 0;
padding: 6px 20px;
font-size: 15px;
line-height: 20px;
height: 34px;
background-color: rgba(0, 0, 0, 0.15);
color: #00aeef;
border: 1px solid rgba(255, 255, 255, 0.15);
box-shadow: 0 0 rgba(0, 0, 0, 0);
border-radius: 2px;
-mwebkit-transition: background-color 0.2s, box-shadow 0.2s, background-color 0.2s, border-color 0.2s, color 0.2s;
transition: background-color 0.2s, box-shadow 0.2s, background-color 0.2s, border-color 0.2s,
```

```
color 0.2s;
}
.btn.active,
.btn:active {
padding: 7px 19px 5px 21px;
}
.btn.disabled:active,
.btn[disabled]:active,
.btn.disabled.active,
.btn[disabled].active {
padding: 6px 20px !important;
}
.btn:hover,
.btn:focus {
background-color: rgba(0, 0, 0, 0.25);
color: #ffffff;
border-color: rgba(255, 255, 255, 0.3);
box-shadow: 0 0 rgba(0, 0, 0, 0);
}
.btn.active,
.btn.active {
background-color: rgba(0, 0, 0, 0.15);
color: rgba(255, 255, 255, 0.8);
border-color: rgba(255, 255, 255, 0.07);
box-shadow: inset 1.5px 1.5px 3px rgba(0, 0, 0, 0.5);
}
.btn-primary {
background-color: #098cc8;
color: #ffffff;
border: 1px solid transparent;
box-shadow: 0 0 rgba(0, 0, 0, 0);
border-radius: 2px;
-webkit-transition: background-color 0.2s, box-shadow 0.2s, background-color 0.2s, border-color 0.2s, color 0.2s;
transition: background-color 0.2s, box-shadow 0.2s, background-color 0.2s, border-color 0.2s, color 0.2s;
background-image: -webkit-linear-gradient(top, #0f9ada, #0076ad);
background-image: linear-gradient(to bottom, #0f9ada, #0076ad);
border: 0;
box-shadow: 0 1px 1px rgba(0, 0, 0, 0.3), 0 0 0 1px rgba(255, 255, 255, 0.15) inset;
}
.btn-primary:hover,
.btn-primary:focus {
background-color: #21b0f1;
color: #ffffff;
border-color: transparent;
box-shadow: 0 0 rgba(0, 0, 0, 0);
}
.btn-primary:active,
.btn-primary.active {
background-color: #006899;
color: rgba(255, 255, 255, 0.7);
border-color: transparent;
box-shadow: inset 1.5px 1.5px 3px rgba(0, 0, 0, 0.5);
}
.btn-primary:hover,
```

```

.btn-primary:focus {
background-image: -webkit-linear-gradient(top, #37c0ff, #0097dd);
background-image: linear-gradient(to bottom, #37c0ff, #0097dd);
}
.btn-primary:active,
.btn-primary.active {
background-image: -webkit-linear-gradient(top, #006ea1, #00608d);
background-image: linear-gradient(to bottom, #006ea1, #00608d);
box-shadow: 1px 1px 2px rgba(0, 0, 0, 0.6) inset, 0 0 0 1px rgba(255, 255, 255, 0.07) inset;
}
.btn-block {
display: block;
width: 100%;
padding-left: 0;
padding-right: 0;
}

.alert {
-moz-box-sizing: border-box;
box-sizing: border-box;
padding: 4px 20px 4px 20px;
font-size: 13px;
line-height: 20px;
margin-bottom: 20px;
text-shadow: none;
position: relative;
background-color: #272e3b;
color: rgba(255, 255, 255, 0.7);
border: 1px solid #000;
box-shadow: 0 0 0 1px #363d49 inset, 0 5px 10px rgba(0, 0, 0, 0.75);
}
.alert-error {
color: #f00;
background-color: #360e10;
box-shadow: 0 0 0 1px #551e21 inset, 0 5px 10px rgba(0, 0, 0, 0.75);
}
.alert-empty{
display: none;
}
.alert-success {
color: #21ec0c;
background-color: #15360e;
box-shadow: 0 0 0 1px #2a551e inset, 0 5px 10px rgba(0, 0, 0, 0.75);
}
</style>
</head>

```

```

<div class="body-content">
<div class="module">
<h1>Add Vehicle Details</h1>
<form class="form" action="addvehicle.php" method="POST" enctype="multipart/form-data"
autocomplete="off">
<div class="alert alert-error"></div>
<input type="text" placeholder="Model Name" name="modelname" required />

```

```

<input type="text" placeholder="Model NO" name="modelno" required />
<input type="text" placeholder="Build Year" name="buildyear" required />
<label>Type</label>
<select name="vehicletype">
<option value="Two Wheeler" >Two Wheeler</option>
<option value="Four Wheeler">Four Wheeler</option></select>

<div class="avatar"><label>Image: </label>
<input type="file" name="imgpath" accept="image/*" value="" required /></div>
<input type="submit" value="ADD Vehicle" name="Click04" class="btn btn-block btn-primary"
/>
</form>
</div>
</div>

</html>

<?php
include("connect.php");

if(isset($_POST["Click04"]))
{

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

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

$filename = $_FILES["imgpath"]["name"];

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

$folder = "image/".$filename;

$userid=$_SESSION['uid'];
$modelname=$_POST["modelname"];
$modelno=$_POST["modelno"];
$buildyear=$_POST["buildyear"];
$vehicletype=$_POST["vehicletype"];

$sql="INSERT INTO tbladdvehicle(usrid, model_name, model_no, Build_year, Type, imgpath)
VALUES (' $userid','$modelname','$modelno', '$buildyear','$vehicletype','$imgpath')";
if(mysqli_query($con,$sql))
{
?>
<script>
alert("Succesfully inserted");

```

```
</script>
<?php
}
else
{
echo "error";
}

mysqli_close($con);

}
// Add the image to the "image" folder"

if (move_uploaded_file($tempname, $folder)) {

$msg = "Image uploaded successfully";

}else{

$msg = "Failed to upload image";

}

?>
```

Addtravel.php

```
<?php
include 'nav.php';
include("connect.php");
?>
<html lang="en" dir="ltr">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<!--<title> Registration or Sign Up form in HTML CSS | CodingLab </title>-->

<link
href="//db.onlinewebfonts.com/c/a4e256ed67403c6ad5d43937ed48a77b?family=Core+Sans+N+W01+
35+Light" rel="stylesheet" type="text/css"/>
<style type="text/css">
*{
margin:0;
padding:0;
}

h1 {
font-size: 2em;
font-family: "Core Sans N W01 35 Light";
font-weight: normal;
```

```
margin: .67em 0;
display: block;
}

#registered {
margin-top: 50px;
}

#registered img {
margin-bottom: 0px;
width: 100px;
height: 100px;
}

#registered span {
clear: both;
display: block;
}

img {
margin-bottom: 20px;
}

.avatar {
margin: 10px 0 20px 0;
}

.module{
position:relative;
top:10%;
height:65%;
width:450px;
margin-left:auto;
margin-right:auto;
}

.user {
color: #66d8fc;
font-weight: bold;
}

.userlist {
float:left;
padding: 30px;
}

.userlist span {
color: #0590fc;
}

.welcome{
position:relative;
top:30%;
height:65%;
width:900px;
margin-left:auto;
```

```
margin-right:auto;
margin-top: 50px;
}

::-moz-selection {
background: #19547c;
}
::selection {
background: #19547c;
}
input::-moz-selection {
background: #037db6;
}
input::selection {
background: #037db6;
}

body{
color: #fff;
background-color:#f0f0f0;
font-family:helvetica;
background:url('http://clevertechie.com/img/bnet-bg.jpg') #0f2439 no-repeat center top;
}

.body-content{
position:relative;
top:20px;
height:700px;
width:800px;
margin-left:auto;
margin-right:auto;
background: transparent;
}

select,
textarea,
input[type="text"],
input[type="password"],
input[type="email"]
{
height:30px;
width:100%;;
display: inline-block;
vertical-align: middle;
height: 34px;
padding: 0 10px;
margin-top: 3px;
margin-bottom: 10px;
font-size: 15px;
line-height: 20px;
border: 1px solid rgba(255, 255, 255, 0.3);
background-color: rgba(0, 0, 0, 0.5);
color: rgba(255, 255, 255, 0.7);
-moz-box-sizing: border-box;
box-sizing: border-box;
border-radius: 2px;
```

```
}

select,
textarea,
input[type="text"],
input[type="password"],
input[type="email"] {
  -webkit-appearance: none;
  -moz-appearance: none;
  -ms-appearance: none;
  appearance: none;
  -webkit-transition: background-position 0.2s, background-color 0.2s, border-color 0.2s, box-shadow 0.2s;
  transition: background-position 0.2s, background-color 0.2s, border-color 0.2s, box-shadow 0.2s;
}
select:hover,
textarea:hover,
input[type="text"]:hover,
input[type="password"]:hover,
input[type="email"]:hover {
  border-color: rgba(255, 255, 255, 0.5);
  background-color: rgba(0, 0, 0, 0.5);
  color: rgba(255, 255, 255, 0.7);
}
select:focus,
textarea:focus,
input[type="text"]:focus,
input[type="password"]:focus,
input[type="email"]:focus {
  border: 2px solid;
  border-color: #1e5f99;
  background-color: rgba(0, 0, 0, 0.5);
  color: #ffffff;
}
.btn {
  text-overflow: ellipsis;
  white-space: nowrap;
  overflow: hidden;
  margin: 3px 0;
  padding: 6px 20px;
  font-size: 15px;
  line-height: 20px;
  height: 34px;
  background-color: rgba(0, 0, 0, 0.15);
  color: #00aeef;
  border: 1px solid rgba(255, 255, 255, 0.15);
  box-shadow: 0 0 rgba(0, 0, 0, 0);
  border-radius: 2px;
  -webkit-transition: background-color 0.2s, box-shadow 0.2s, background-color 0.2s, border-color 0.2s, color 0.2s;
  transition: background-color 0.2s, box-shadow 0.2s, background-color 0.2s, border-color 0.2s, color 0.2s;
}
.btn.active,
.btn:active {
  padding: 7px 19px 5px 21px;
```

```

}
.btn.disabled:active,
.btn[disabled]:active,
.btn.disabled.active,
.btn[disabled].active {
padding: 6px 20px !important;
}
.btn:hover,
.btn:focus {
background-color: rgba(0, 0, 0, 0.25);
color: #ffffff;
border-color: rgba(255, 255, 255, 0.3);
box-shadow: 0 0 rgba(0, 0, 0, 0);
}
.btn:active,
.btn.active {
background-color: rgba(0, 0, 0, 0.15);
color: rgba(255, 255, 255, 0.8);
border-color: rgba(255, 255, 255, 0.07);
box-shadow: inset 1.5px 1.5px 3px rgba(0, 0, 0, 0.5);
}
.btn-primary {
background-color: #098cc8;
color: #ffffff;
border: 1px solid transparent;
box-shadow: 0 0 rgba(0, 0, 0, 0);
border-radius: 2px;
-webkit-transition: background-color 0.2s, box-shadow 0.2s, background-color 0.2s, border-color 0.2s, color 0.2s;
transition: background-color 0.2s, box-shadow 0.2s, background-color 0.2s, border-color 0.2s, color 0.2s;
background-image: -webkit-linear-gradient(top, #0f9ada, #0076ad);
background-image: linear-gradient(to bottom, #0f9ada, #0076ad);
border: 0;
box-shadow: 0 1px 1px rgba(0, 0, 0, 0.3), 0 0 1px rgba(255, 255, 255, 0.15) inset;
}
.btn-primary:hover,
.btn-primary:focus {
background-color: #21b0f1;
color: #ffffff;
border-color: transparent;
box-shadow: 0 0 rgba(0, 0, 0, 0);
}
.btn-primary:active,
.btn-primary.active {
background-color: #006899;
color: rgba(255, 255, 255, 0.7);
border-color: transparent;
box-shadow: inset 1.5px 1.5px 3px rgba(0, 0, 0, 0.5);
}
.btn-primary:hover,
.btn-primary:focus {
background-image: -webkit-linear-gradient(top, #37c0ff, #0097dd);
background-image: linear-gradient(to bottom, #37c0ff, #0097dd);
}
.btn-primary:active,

```

```

.btn-primary.active {
background-image: -webkit-linear-gradient(top, #006ea1, #00608d);
background-image: linear-gradient(to bottom, #006ea1, #00608d);
box-shadow: 1px 1px 2px rgba(0, 0, 0, 0.6) inset, 0 0 0 1px rgba(255, 255, 255, 0.07) inset;
}
.btn-block {
display: block;
width: 100%;
padding-left: 0;
padding-right: 0;
}

.alert {
-moz-box-sizing: border-box;
box-sizing: border-box;
padding: 4px 20px 4px 20px;
font-size: 13px;
line-height: 20px;
margin-bottom: 20px;
text-shadow: none;
position: relative;
background-color: #272e3b;
color: rgba(255, 255, 255, 0.7);
border: 1px solid #000;
box-shadow: 0 0 0 1px #363d49 inset, 0 5px 10px rgba(0, 0, 0, 0.75);
}
.alert-error {
color: #f00;
background-color: #360e10;
box-shadow: 0 0 0 1px #551e21 inset, 0 5px 10px rgba(0, 0, 0, 0.75);
}
.alert-empty{
display: none;
}
.alert-success {
color: #21ec0c;
background-color: #15360e;
box-shadow: 0 0 0 1px #2a551e inset, 0 5px 10px rgba(0, 0, 0, 0.75);
}
</style>
</head>

```

```

<div class="body-content">
<div class="module">
<h1>Add Travel Details</h1>
<form class="form" action="addtravel.php" method="POST" enctype="multipart/form-data"
autocomplete="off">
<div class="alert alert-error"></div>
<input type="text" placeholder="From" name="From" required />
<input type="text" placeholder="To" name="To" required />
<input type="Date" placeholder="Date" name="Date" required />
<input type="time" placeholder="time" name="time" required>
<label>Select Type</label>
<select name= "vehicletype" >

```

```

<option selected="selected" value="">-- Select an option --</option>
<?php
include("connect.php");
$sql = "SELECT * From tbladdvehicle";
$result=mysqli_query($con,$sql);

while($row=mysqli_fetch_array($result))

echo "<option value=" . $row['model_name'] . ">" . $row['vid'] . $row['model_name'] . "</option>";
?>
</select>
<input type="text" placeholder="Amount" name="amount" required />
<input type="text" placeholder="No_of_Seats" name="noofseats" required />

<input type="submit" value="ADD Vehicle" name="Click05" class="btn btn-block btn-primary" />
</form>
</div>
</div>

</html>

<?php
include("connect.php");

if(isset($_POST["Click05"]))
{
    $userid=$_SESSION['uid'];
    $From1=$_POST["From"];
    $To1=$_POST["To"];
    $Date=$_POST["Date"];
    $time=$_POST["time"];
    $amount=$_POST["amount"];
    $No_of_seat=$_POST["noofseats"];

    $v="SELECT vid FROM tbladdvehicle WHERE usrid=$userid";
    $res=mysqli_query($con,$v);
    while ($row=mysqli_fetch_array($res)) {
        $vvid=$row['vid'];
    }

    $carname=$_POST['vehicletype'];

    $sql="INSERT INTO tbltraveldetails(uid, vid, from1, To1, Date1, Time1, Carname, Amount,
    No_of_seats)
    VALUES ('$userid','$vvid','$From1','$To1','$Date','$time','$carname','$amount','$No_of_seat')";

    if(mysqli_query($con,$sql))
    {
        ?>
        <script>
        alert("Succesfully inserted");
        </script>

```

```
<?php
}
else
{
echo "error";
}
mysqli_close($con);

}
?>
```

Profile.php

```
<?php
include 'nav.php';
?>
<?php

include("connect.php");
$uuid= $_SESSION['uid'];

$query="select * from registration where userid='$uuid'";
$result = $con->query($query);
if ($result->num_rows > 0) {
while($row = $result->fetch_assoc()) {
$fullname=$row["fullname"];
$username=$row["username"];
$email=$row["email"];
$phone=$row["phonno"];
$place=$row["place"];
$state=$row["state"];
$pin=$row["pin"];
$DOB=$row["dob"];
$usertype=$row["usertype"];
$adhar=$row["adhar"];
$password=$row["password"];
$gender=$row["gender"];

}

} else {
echo "0 results";
}
?>
<html>
<head>
<script type = "text/javascript">

function validate() {
var password1=document.myForm.pass.value;
var password2=document.myForm.conpass.value;

if (document.myForm.fullname.value=="")
```

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

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

if (document.myForm.pin.value=="")
{
document.getElementById('pi').innerHTML="**Enter Pin";
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.dob.value=="")
```

```

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

return false;
}

}

</script>
</head>
<style type="text/css">

input[type=text] {
border: 3px inset #FFA5A5;
outline:0;
height:25px;
width: 275px;
}
input[type=Date] {
border: 3px inset #FFA5A5;
outline:0;
height:25px;
width: 275px;
}
.button {
background-color: #f44336; /* Green */
border: none;
color: white;
padding: 15px 32px;
text-align: center;
text-decoration: none;
display: inline-block;
font-size: 16px;
margin: 4px 2px;
cursor: pointer;

}

}

</style>
<body background="http://clevertechie.com/img/bnet-bg.jpg">
<center> <h1 style="color: white;">Profile</h1>

<form name = "myForm" action="profile.php" method="POST" onsubmit = "return(validate());"

```

```

enctype="multipart/form-data">
<table cellpadding="25">
<tr>
<th>
<span class="details" style="color: white;">Full Name <span id="fn"
style="color:red;"></span></span>
</th><th>
<input type="text" placeholder="Enter your name" name="fullname" value="<?php echo
$fullname;?>">
</th>
</tr>
<tr>
<th><span class="details" style="color: white;">Username <span id="un"
style="color:red;"></span></span> </th>
<th><input type="text" name="uname" placeholder="Enter your username" value="<?php echo
$username;?>"></th>
</tr>
<tr>
<th><span class="details" style="color: white;">Email <span id="em"
style="color:red;"></span></span></th>
<th><input type="text" placeholder="Enter your email" name="email" value="<?php echo
$email;?>"></th></tr>

<tr> <th> <span class="details" style="color: white;">Phone Number <span id="pn"
style="color:red;"></span></span></th>
<th><input type="text" placeholder="Enter your number" name="phone" value="<?php echo
$phone;?>"></th></tr>

<tr><th> <span class="details" style="color: white;">Place <span id="pl"
style="color:red;"></span></span></th>
<th><input type="text" placeholder="Enter your number" name="place" value="<?php echo
$place;?>"></th></tr>

<tr><th> <span class="details" style="color: white;">State <span id="st"
style="color:red;"></span></span></th>
<th><input type="text" placeholder="Enter your number" name="state" value="<?php echo
$state;?>"></th></tr>

<tr><th><span class="details" style="color: white;">pin <span id="pi"
style="color:red;"></span></span></th>
<th><input type="text" placeholder="Enter your pin number" name="pin" value="<?php echo
$pin;?>"></th></tr>

<tr><th><span class="details" style="color: white;">DOB <span id="do"
style="color:red;"></span></span></th>
<th><input type="date" placeholder="Enter your DOB" name="dob" max="<?= date('Y-m-d'); ?>"
value="<?php echo $DOB;?>"></th></tr>

<tr><th> <span class="details" style="color: white;">Adhar Number <span id="ad"
style="color:red;"></span></span></th>
<th><input type="text" placeholder="Enter your adhar number" name="adhar" value="<?php echo
$adhar;?>"></th></tr>

<tr><th></th>

<th>

```

```
<input class="button" type="submit" value="Update" name="Click3">
</th></tr>
</table>
</form>
</center> </body>
</html>

<?php
include("connect.php");

if(isset($_POST["Click3"])){
$uuid= $_SESSION['uid'];

$fullname=$_POST["fullname"];
$username=$_POST["uname"];
$email=$_POST["email"];
$phone=$_POST["phone"];
$place=$_POST["place"];
$state=$_POST["state"];
$pin=$_POST["pin"];
$DOB=$_POST["dob"];

$adhar=$_POST["adhar"];

$sql="update registration set
fullname='$fullname',username='$username',email='$email',phonno='$phone',place='$place',state='$state',pin='$pin',dob='$DOB',adhar='$adhar' where userid='$uuid'";

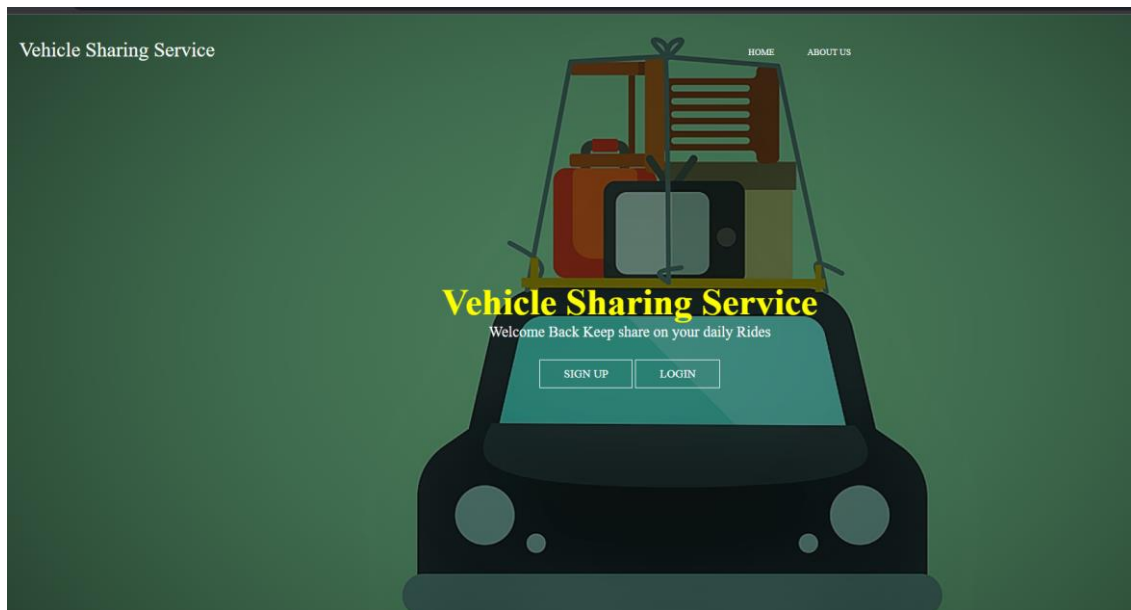
if(mysqli_query($con,$sql))
{
?>
<script>
alert("Profile updated Succesfully");
</script>
<?php
}
else
{
echo "error";
}

mysqli_close($con);

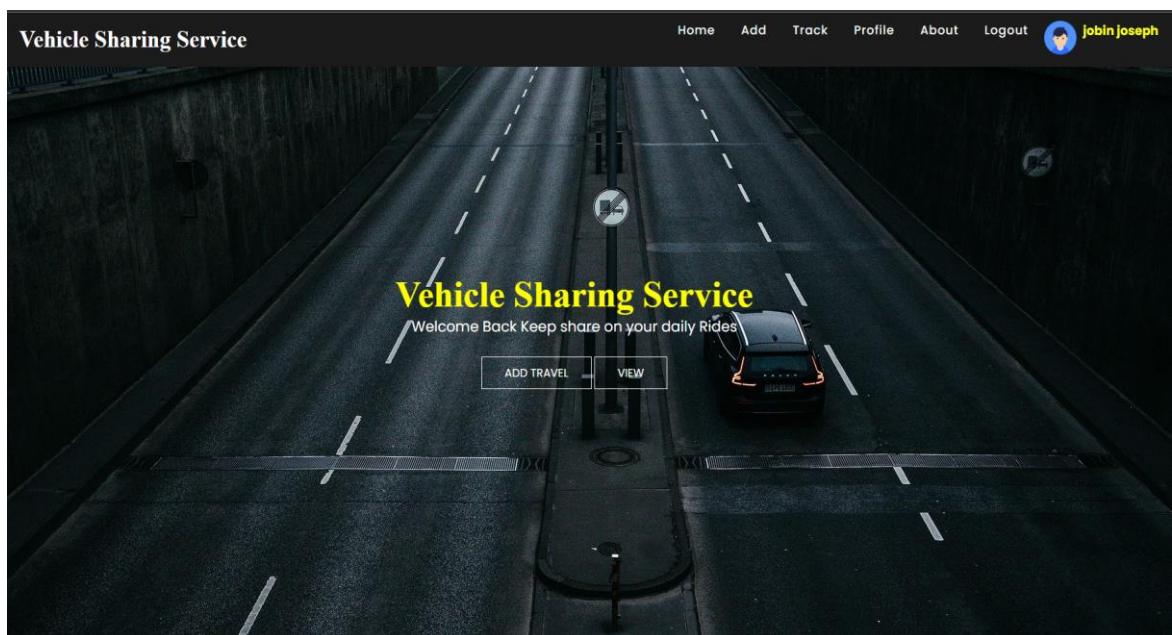
}
?>
```

9.2 Screen Shots

Home page



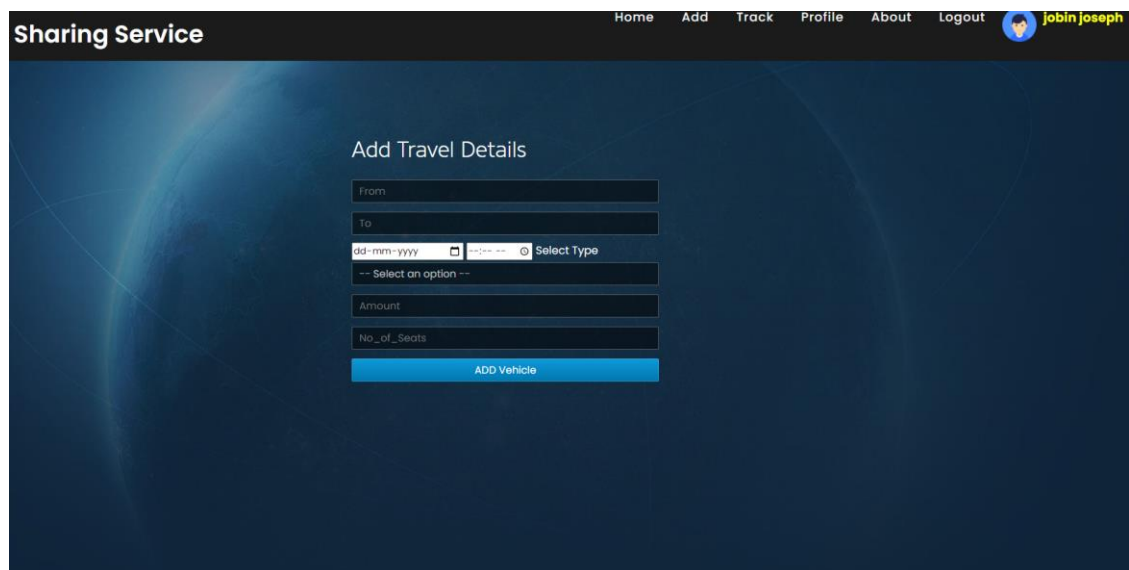
Vehicle owner home page



Addvehicle

The screenshot displays the 'Add Vehicle Details' form within the 'Vehicle Sharing Service' application. The form is set against a dark blue background with a subtle globe pattern. It includes input fields for 'Model Name', 'Model NO', and 'Build Year'. A 'Type' dropdown menu is currently set to 'Two Wheeler'. Below these fields is an 'Image' section with a 'Choose File' button and the text 'No file chosen'. At the bottom of the form is a blue 'ADD Vehicle' button. The top navigation bar is consistent with the previous screenshot, showing the user 'Jobin Joseph' is logged in.

Add Travel



The screenshot shows a web application titled "Sharing Service" with a dark blue header. The header contains navigation links: Home, Add, Track, Profile, About, and Logout. A user profile icon and the name "jobin joseph" are visible on the right. The main content area has a dark blue background with a faint globe pattern. In the center, there is a form titled "Add Travel Details". The form includes the following fields: "From" (text input), "To" (text input), a date field with a calendar icon and placeholder "dd-mm-yyyy", a "Select Type" dropdown menu with "-- Select an option --", "Amount" (text input), and "No. of Seats" (text input). At the bottom of the form is a blue button labeled "ADD Vehicle".


Manage Travel Requests

Vehicle Sharing Service

HomeAddTrackProfileAboutLogoutjobin joseph

Travel Requests


List



Your Name: Albin JoseOwner Name: jobin josephFrom: mdkmTo: palaNo_of_seats: 1

Vechile name: dlxTime: 09:56Date: 2022-02-24Type: four WheelerAmount: 200

ApproveReject



Your Name: Albin JoseOwner Name: jobin josephFrom: palaTo: ktmNo_of_seats: 2

Vechile name: maruthiTime: 06:21Date: 2022-02-09Type: Four WheelerAmount: 500

ApproveReject

Edit profile

Vehicle Sharing Service

HomeAddTrackProfileAboutLogoutjobin joseph

Profile

Full Name

jobin joseph

Username

jobin124

Email

jobinjoseph45@gmail.com

Phone Number

2147483647

Place

kottayam

State

Kerala

pin

407892

DOB

15-01-1999

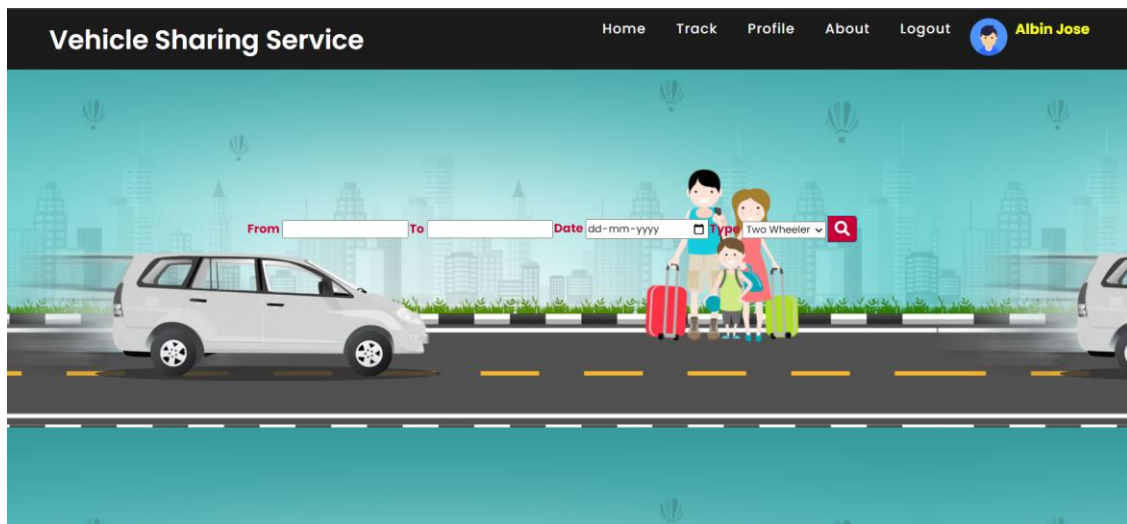
Adhar Number

2147483647

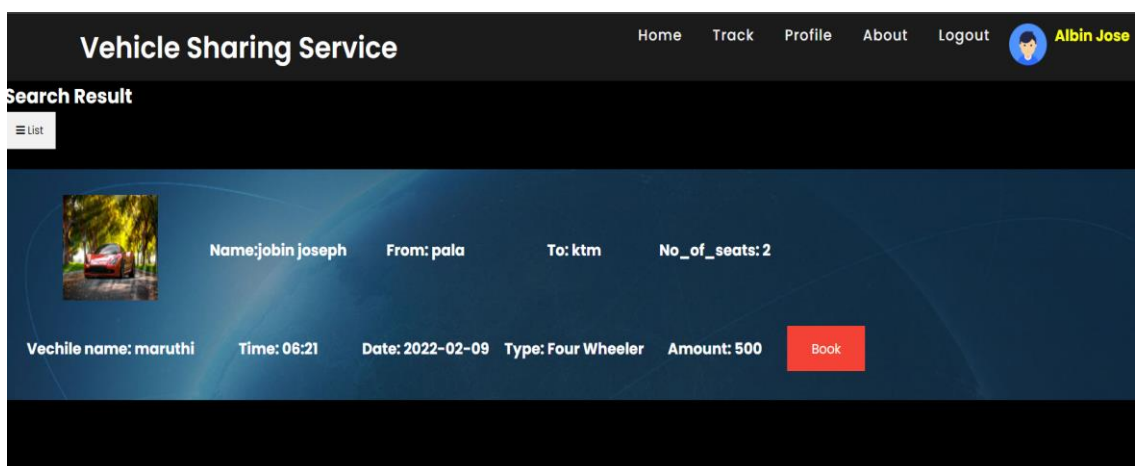
Update

VEHICLE USERS PAGES


Users home page



Search vehicle




Request status

Vehicle Sharing Service[Home](#)[Track](#)[Profile](#)[About](#)[Logout](#) **Albin Jose**


Search Result


ListGrid



Your Name: Albin Jose Owner Name: jobin joseph From: mdkm To: pala No_of_seats: 1


Vechile name: dlx Time: 09:56 Date: 2022-02-24 Type: four Wheeler Amount: 200





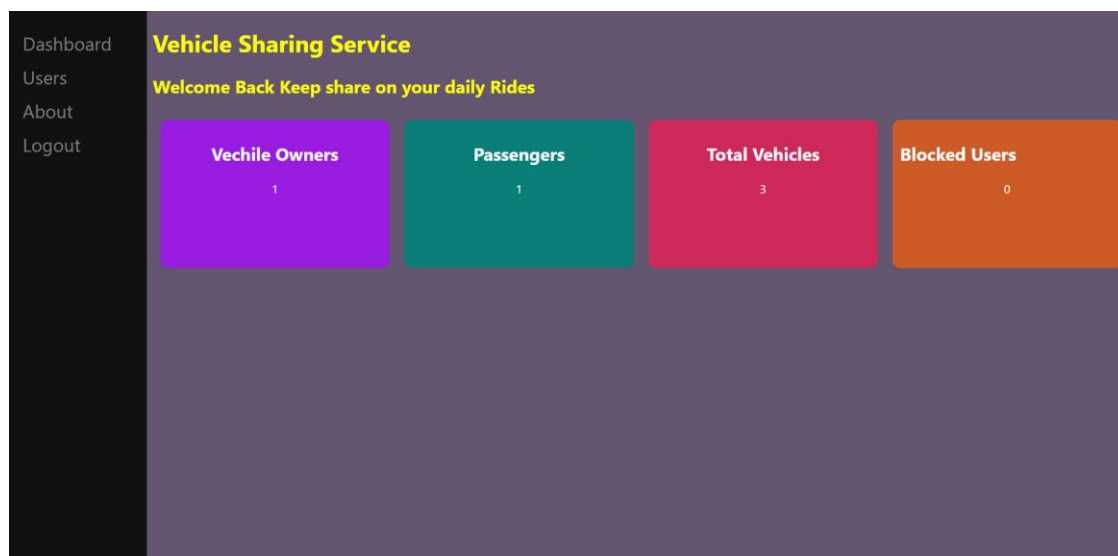
Your Name: Albin Jose Owner Name: jobin joseph From: pala To: ktm No_of_seats: 2

Vechile name: maruthi Time: 06:21 Date: 2022-02-09 Type: Four Wheeler Amount: 500



ADMIN PAGES

Admin Home



Manage Users and Owners

The Manage Users and Owners page displays a table of registered users. The sidebar is identical to the Admin Home page. The table has 14 columns: userid, fullname, username, email, phonno, place, state, pin, dob, usertype, adhar, gender, Status, and an action column. Two users are listed: jobin joseph (userid 21, status 0) and Albin Jose (userid 22, status 1). The action column contains "Block" and "Unblock" buttons respectively.

userid	fullname	username	email	phonno	place	state	pin	dob	usertype	adhar	gender	Status	
21	jobin joseph	jobin124	jobinjoseph2501@gmail.com	2147483647	Chittady	Kerala	457892	2022-01-13	1	2147483647	male	0	Block
22	Albin Jose	albin123	albin@gmail.com	2147483647	mundakayam	Kerala	147895	2019-02-03	2	2147483647	male	1	Unblock