# ABSTRACT

The Car Reservation System is being developed for customers so that they can book their vehicles from any part of the world. This application takes information from the customers through filling their details. A customer being registered in the website has the facility to book a vehicle which he requires. It is an online system through which customers can view available cars, register and book car. We developed this project to book a car on rent at the fare charges. In present system all booking work done manually and it takes very hard work to maintain the information of booking and cars. if you want to find which vehicle is available for booking then it takes a lot of time. It only makes the process more difficult and hard. This aim of the project is to automate the work performed in the car rental management system like records of cab, cabs available for booking, rental charges for cars, store records of the customer. Cars is a car booking software that provides a complete solution to all your day-to-day car booking office running needs. This system helps you to keep the information of customer online. You can check your customer information any time by using this system. Online car rental management system is a unique and innovative product. Based on this information you can take decision regarding your business development.

# ACKNOWLEDGEMENT

This project is done as a semester project. We wish to express my sincere thanks to the HOD **Dr. Chandrika J.** for giving this opportunity. We are thankful to our course faculty advisor, **Mrs. Sindhu K.S** Assistant Professor, Department of ISE, for her valuable guidance and assistance, without which the accomplishment of the task would have never been possible. And we are thankful to our **Department faculty staff** who have given a lot of suggestions in carrying out our project.

In our present project we have chosen the topic- "Car Reservation System". This report contains the complete computerized management of Supermarket. This system reflects standard structure so that any inventory management system can implements this system easily in the existing system. This system works to reduce the human efforts. Due to totally computerized occurrence of error is less. This system works smoothly when used.

We are also thankful to the persons who are indirectly involved in the project completion. We are also thankful to friends who have supported us in the successful completion of this project.

# CONTENTS

# Chapter 1

# INTRODUCTION

## 1.1 Introduction to the Car reservation system

A database management system (DBMS) refers to the technology for creating and managing databases. DBMS is a software tool to organize (create, retrieve, update and manage) data in a database. The main aim of a DBMS is to supply a way to store up and retrieve database information that is both convenient and efficient. By data, we mean known facts that can be recorded and that have embedded meaning. Normally people use software such as DBASE IV or V, Microsoft ACCESS, or EXCEL to store data in the form of a database.

Database systems are meant to handle a large collection of information. Management of data involves both defining structures for the storage of information and providing mechanisms that can do the manipulation that stored information. Moreover, the database system must ensure the safety of the information stored, despite system crashes or attempts at unauthorized access.

## 1.2 Potential of the problem

### 1.2.1 PROBLEM STATEMENT

In real world, not every person can afford their own personal car . A car rental is a vehicle that can be used temporarily for a fee during a specified period. Getting a rental car helps people get around despite the fact they do not have access to their own personal vehicle or don't own a vehicle at all. The individual who needs a car must contact a rental car company and contract out for a vehicle. This system increases customer retention and simplify vehicle and staff management.

### 1.2.2 EXISTING SYSTEM

The existing car reservation system can vary depending on the specific company and location, but they generally follow a similar structure. Here’s an overview of a typical car reservation system:

- ✓ Online reservation
- ✓ Inspection and contract signing
- ✓ Reservation period
- ✓ Follow-up

## 1.2.3PROPOSED SYSTEM

The idea is to design a management application that will:

- Reclaim cars
- Reclaim customers
- Show available cars to customers
- Inform customers of the status of their reservation(cancel,pending,confirm)
- Take stock of car location per month

**Features:**

- Login/Register System
- Easy Bookings
- Create/Manage Brands
- Post/Manage Vehicle
- Manage Bookings, Testimonials, Contact Queries, Pages.

# 1.3 Objectives of the present work

## 1.3.1OBJECTIVES AND SCOPE

- To produce a web-based system that allow customer to register and reserve car online and for the company to effectively manage their car rental business.
- As all the system is computerized, there is no need to fill any application form for renting purpose. So, the paperwork will be very less.
- To make sure a user gets his desire car as early as possible. The car rental system will provide a faster response to complete the process.

# 1.4 Platform and tools used

- Tools : Xampp
- External Library: PHP
- DBMS : MYSQL
- Browser : HTML5, CSS3, Apache webserver

# Chapter 2

# REQUIREMENT ANALYSIS AND SPECIFICATION

## 2.1 Functional Requirements Car Reservation System

These are statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations. In some cases, the functional requirements may also explicitly state what the system should not do. The functional requirements for a system describe what the system should do. These requirements depend on the type of software being developed, the expected users of. user requirements, the requirements are usually described abstractly. However, functional system requirements describe the system function in detail, its inputs and outputs, exceptions, and so on. Functional requirements for a software system may be expressed in several ways. **The functional requirements of CAR RESERVATION SYSTEM is as follow:**

**Register Module:**

- The user needs to provide their first name, last name, email, license number, phone number, password, confirm password, gender for registration.
- These details will be stored in database.

**Login Module:**

- For login user will input their email and password .
- Admin will provide their admin id and password which will compared with a database content.

**Booking Module:**

- User can view the list of cars. The booking details of cars are provided by the admin.
- User can select their preferred car and book for the same.

**Logout Module:**

- The system should allow user to logout .
- The system should also allow admin to logout.

**Hardware Requirements:**

- Processor : Intel i3/i5/1.8GHz machine or above

- Primary memory : 4 GB RAM or above
- Hard disk drive : 1 TB or greater

**Software Requirements**

- Operating System : Windows 7 or higher
- Front End : HTML5,CSS3,JavaScript
- Back End : PHP, SQL
- Frame work : Bootstrap
- Software : Visual Studio Code, XAMPP

# 2.2 Non-Functional Requirements of Car Reservation System

Non-functional requirements are requirements that are not directly concerned with the specific functions delivered by the system. They may relate to emergent system properties such as reliability, response time and store occupancy. Alternatively, they may define constraints on the system such as the capabilities of I/O devices and the data representations used in system interfaces. The plan for implementing functional requirements is detailed in the system design. The plan for implementing non-functional requirements is detailed in the system architecture. Non-functional requirements are often called qualities of a system. Other terms for non-functional requirements are "constraints", "quality attributes", "quality goals", "quality of service requirements" and "non-behavioral requirements". Qualities, that are non-functional requirements, can be divided into two main categories: Execution qualities, such as security and usability, which are observable at run time.

**Security:**

- The system should provide a high level of security and integrity of the data held by the system , only authorized personnel of the company can gain access to the company's secured page on the system.

**Performance:**

- The system should have high performance rate when executing user's input and should be able to provide feedback or response within a short time span usually 50 seconds for highly complicated task and 20 to 25 seconds for less complicated task.

**Reliability:**

- It is the probability and percentage of the system performing without any failure for a specific number of uses or amount of time.

**Consistency:**

- The car rental system provides consistency services, by retaining the data present in the database.

# Chapter 3

# SYSTEM DESIGN

System Design process partitions the system into subsystems based on the requirements. It establishes overall system architecture and is concerned with identifying various components, specifying relationships among components, specifying software structure, maintaining a record of design decisions and providing a blue print for the implementation phase.

## 3.1 ENTITY RELATION DIAGRAM

ER Diagram stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases. ER diagrams are created based on three basic concepts: entities, attributes and relationships.ER Diagrams contain different symbols that use rectangles to represent entities, ovals to define attributes and diamond shapes to represent relationships.

In the Figure 3.1, there are total 6 entities namely Admin, User, Car, Booking, Payment and Feedback Details. We took a relation APPROVE as a relationship between Admin and Booking entity with 1:N cardinality ratio because One admin can approve many booking. User entity has relationship DOES with Booking entity with N:M cardinality ratio since many users can does many bookings. The relation User has M:N relationship named GIVES with Feedback because Many user can give many feedbacks. Car has N:M relationship with Booking entity as RENTS. Since car can have N bookings. Booking Details has 1:1 relationship between Payment .In our ER diagram the relation Booking and Payment is total participation and relation admin and booking, relation user and booking, relation user and feedback, relation car and booking are partial participation.

Our ER-model consists of as follows in below:

- Entity, Entity set:
  - ✓ Strong entity
  - ✓ Weak entity
- Attributes.
- Relationship Type and Relationship Set.
- Cardinality.
- Participation Constraint.

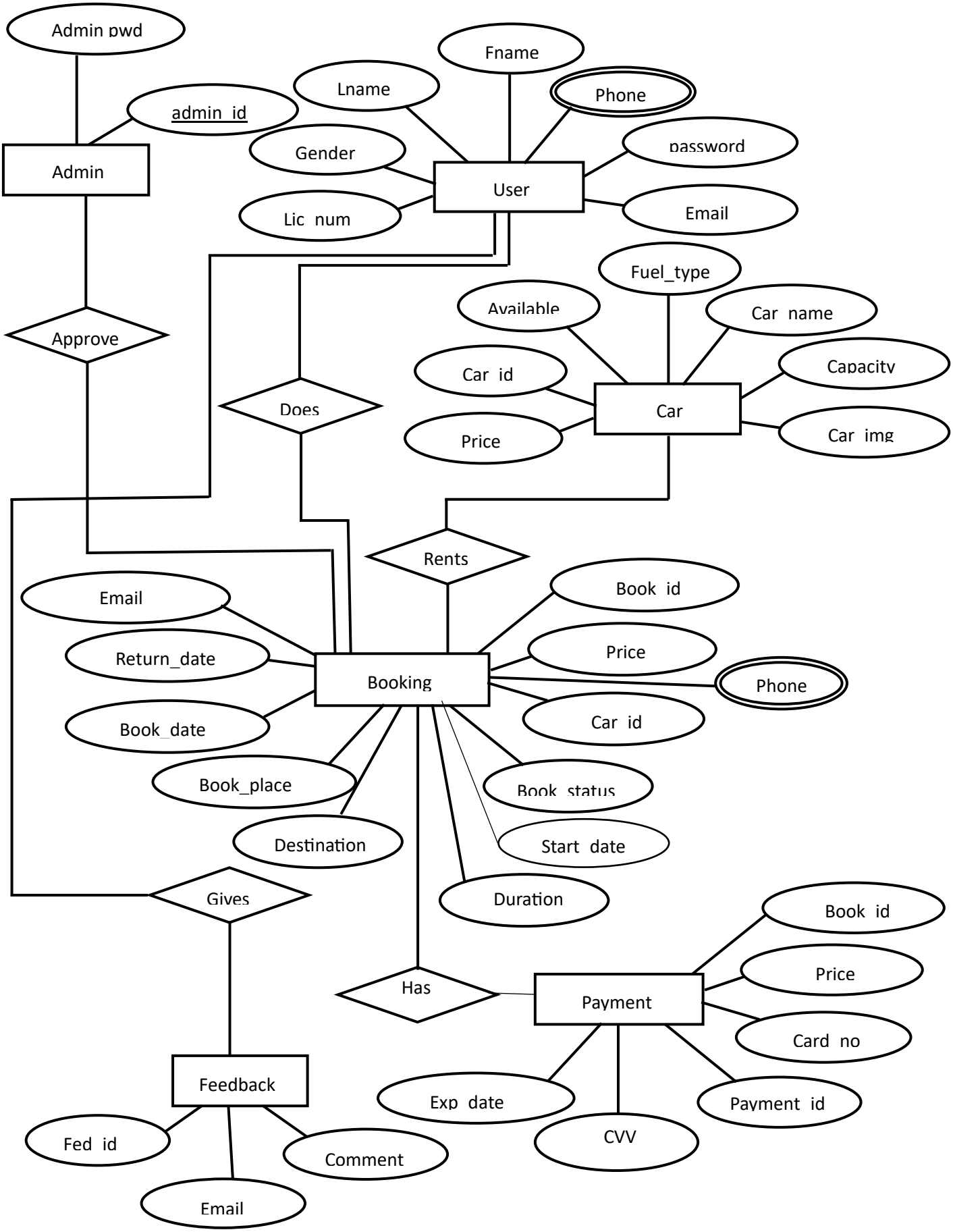



**Fig 3.1 : ER DIAGRAM**

# 3.2 SCHEMA DIAGRAM

The design of the database is called a schema. This tells us about the structural view of the database. It gives us an overall description of the database. A database schema defines how the data is organized using the schema diagram. A schema diagram is a diagram which contains entities and the attributes that will define that schema. A schema diagram only shows us the database design. It does not show the actual data of the database. Schema can be a single table or it can have more than one table which is related. The schema represents the relationship between these tables.

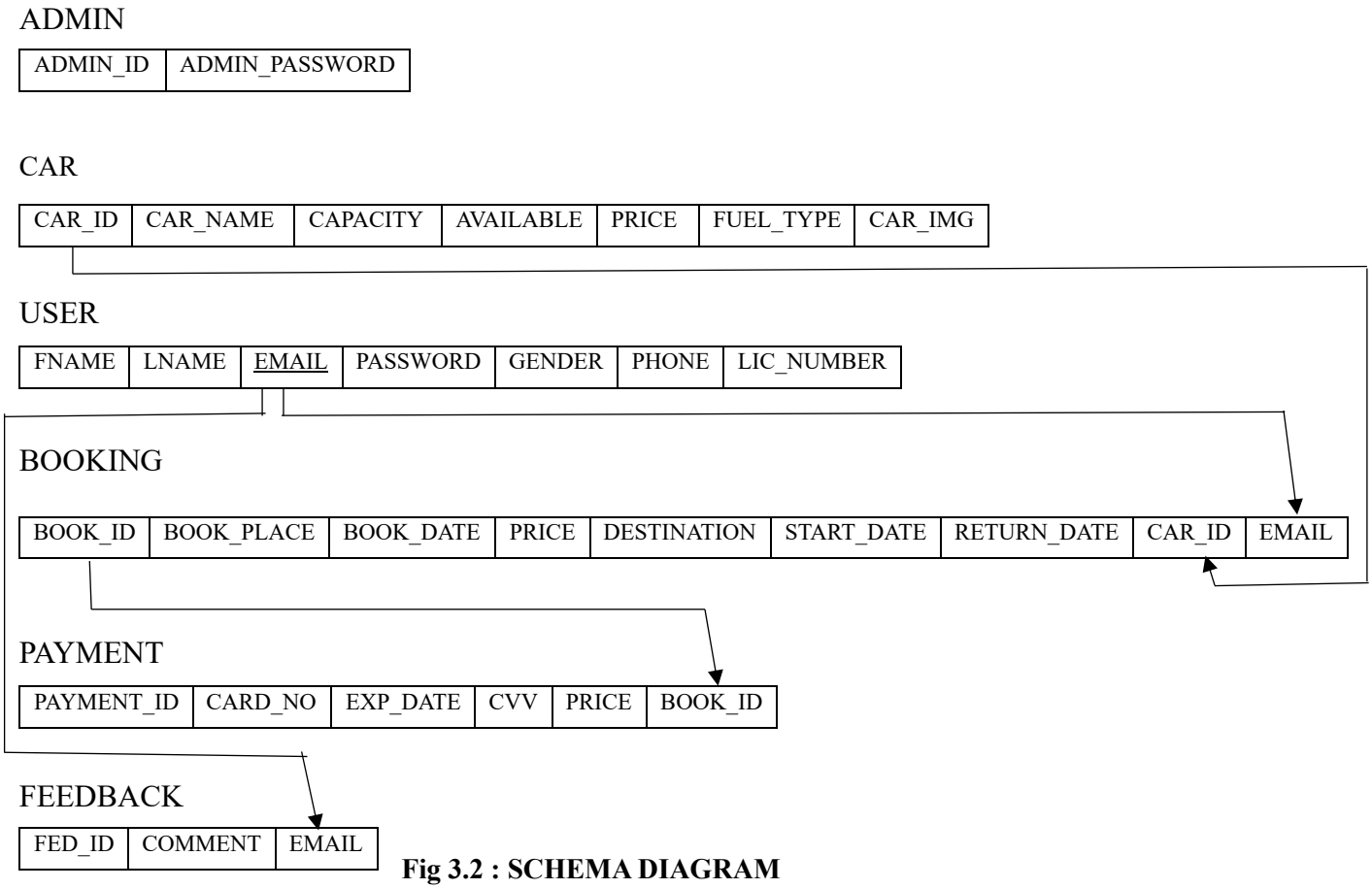



**Fig 3.2 : SCHEMA DIAGRAM**

### 3.2.1 DESCRIPTION OF TABLES

The database consists of six tables:

1. ADMIN: It stores the details of admin.

- ADMIN_ID : User name of the admin.
- ADMIN_PASSWORD: Password of the Admin.

2. CAR: It gives the details about the car.

- CAR_ID : Id given to car done by auto increment.
- CAR_NAME: Name of the car.
- CAPACITY: Seat capacity.

- AVAILABLE: Availability of car.
- PRICE: Price of the car.
- FUEL_TYPE: Car fuel type.
- CAR_IMG: Image of car.

3. USER: It stores the details of user.

- FNAME: first name of user.
- LNAME: Last name of user.
- EMAIL: Email of user.
- PASSWORD: Password of user.
- GENDER: Gender of user.
- PHONE: Phone number of user.
- LIC_NUMBER: License no of user.

4. BOOKING: It give the booking details for user.

- BOOK_ID: Booking id done by auto increment.
- BOOK_PLACE: Place of booking.
- BOOK_DATE: Date of booking.
- PRICE: Price of car.
- DESTINATION: Destination.
- RETURN_DATE: Return date.
- CAR_ID: Id given to car and foreign key car associated with booking.
- EMAIL: Email of user and foreign key of user associated with booking.

5. PAYMENT: It provides payment option for users.

- PAYMENT_ID: Id given to payment and done by auto increment.
- CARD_NO: Card number.
- EXP_DATE: Expiry Date of card.
- CVV: CVV of card.
- PRICE: Price of car.
- BOOK_ID: Id given to booking and foreign key of booking associated with payment.

6. FEEDBACK: It provides user to give their feedback.

- FEED_ID: id given to the feedback done by auto increment.
- COMMENT: Message about their experience.
- EMAIL: Email of user and foreign key of user associated with feedback.

# 3.3 BLOCK DIAGRAM

A Block diagram is a diagram of a system in which the principal parts or functions are represented by blocks connected by lines that show the relationships of the blocks. They are heavily used in engineering in hardware design, electronic design, software design, and diagrams. Block diagrams are typically used for higher level, less detailed descriptions that are intended to clarify overall concepts without concern for the details of implementation. Contrast this with the schematic diagrams and layout diagrams used in electrical engineering, which show the implementation details of electrical components and physical construction.

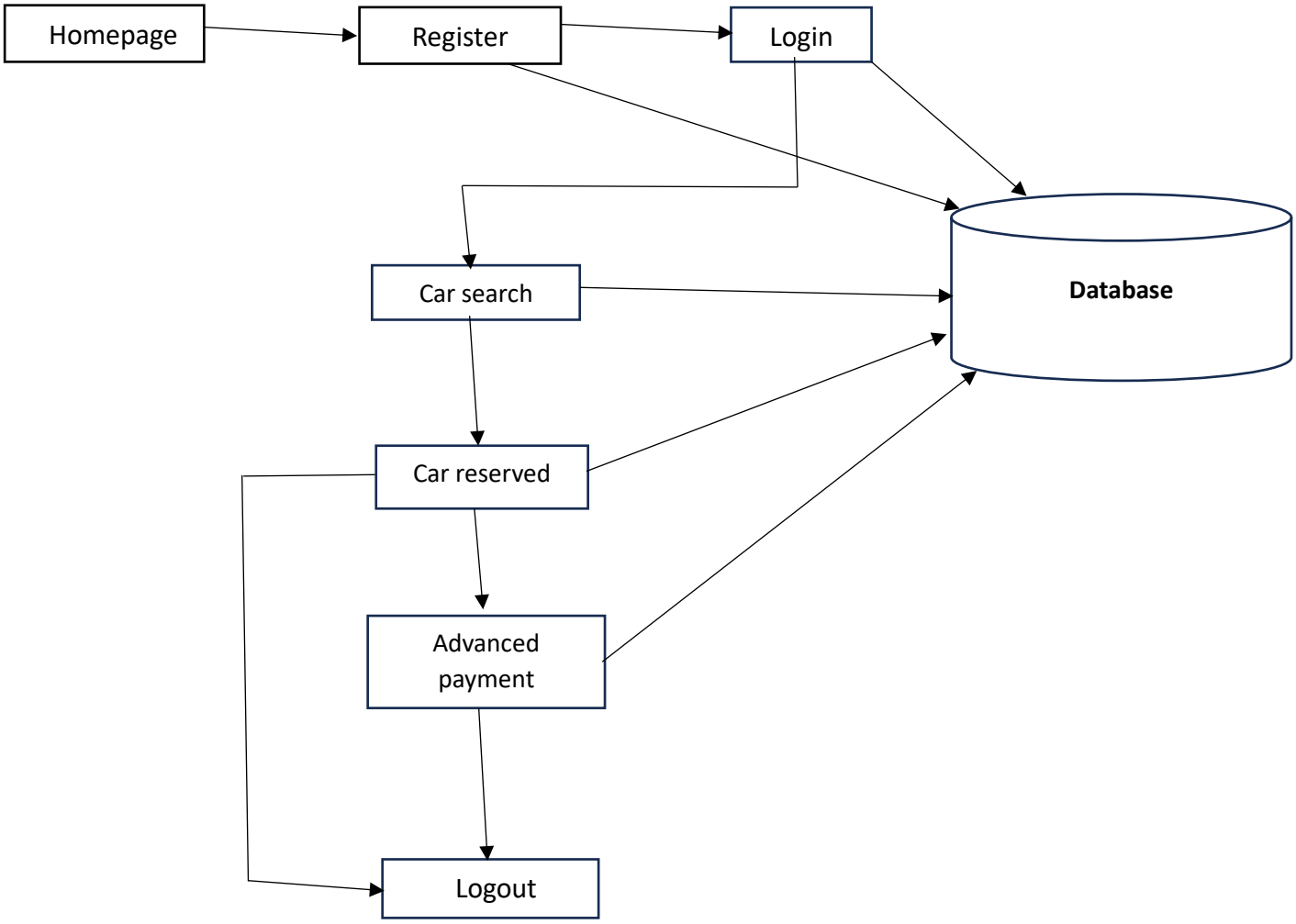



**Figure 3.3 : Block Diagram of Car Rental System**

In the Figure 3.3, the user and admin register themselves by providing their credentials. These credentials are stored in the database. During the login phase, the user and admin details are verified with the data present in the database. After verification, the authentic user can only be allowed to enter and perform necessary operations. These operations include fetching data from the database like adding/updating the car and booking details. The users are allowed to see available cars and can done booking of cars and payment for their booked cars. All these operations are performed on the database and are updated accordingly. After all the intended operations are completed, the user can log out. The details will be present in the database for the next time the user logs in.

# Chapter 4

# IMPLEMENTATION

PHP: Hypertext Pre-processor (or simply PHP) is a server-side scripting language designed for web development, and also used as a general-purpose programming language. PHP code may be embedded into HTML code, or it can be used in combination with various web template systems, web content management systems, and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command-line interface (CLI) and can be used to implement standalone graphical applications.

This project uses HTML as front-end tool. Hypertext Mark-up Language is the standard mark-up language for creating web pages and web applications. With Cascading Style Sheets and JavaScript, it forms a triad of cornerstone technologies for the world wide web. Web browser receive HTML documents from a web server or from local storage and render the documents into multimedia web pages.HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

## 4.1 Code Snippet

Snippet is a programming term for a small region of re-usable code, machine code or text. Ordinarily, these are formally defined operative units to incorporate into large programming modules. These are few codes written in PHP that includes a database access to fetch, update or delete the data stored and display these data in the website where required. We have also ensured to make sure the backend implementation is free from SQL injection attacks by escaping the query values. **Connection mysql to all php and html files in our project:**

```
<?php
  mysqli_report(MYSQLI_REPORT_ERROR | MYSQLI_REPORT_STRICT);
  $con = mysqli_connect('localhost','root','','carproject');
  if(!$con)
  {
    echo 'please check your Database connection';
  }
```

# Chapter 5

# TESTING

Software testing is the process of used to identify the correctness, security, completeness and quality of developed computer software. This includes the process of executing the program or applications with the intent of finding errors. An individual unit, functions or procedures of developed project is verified and validated and these units are fit for use.

## 5.1 Test Cases

A Test Case is a software testing document, which consists of events, action, input, output, expected result and actual result. Technically a test case includes test description, procedure, expected result and remarks. Test cases should be based primarily on the software requirements and developed to verify correct functionality and to establish conditions that reveal potential errors.

| Test cases no | Test case | Expected results | Status |
|---|---|---|---|
| 1 | Logging into website | Email and password provided correct | Successful |
| 2 | Logging into website | Email incorrect | Unsuccessful |
| 3 | Logging into website | Password Incorrect | Unsuccessful |
| 4 | Logging into website | Any field left empty | Unsuccessful |

**Table 5.1.1 : Test Case for Login**

| Test cases no | Test case | Expected results | Status |
|---|---|---|---|
| 1 | Registration for new user | All details provided correctly | Successful |
| 2 | Registration for new user | Any one field is incorrect | Unsuccessful |
| 3 | Registration for new user | Any field left empty | Unsuccessful |

**Table 5.1.2 : Test Case for Signup**

| Test cases no | Test case | Expected results | Status |
|---|---|---|---|
| 1 | Payment | All details provided correctly | Successful |
| 2 | Payment | Any one field is incorrect | Unsuccessful |
| 3 | Payment | Any field left empty | Unsuccessful |

**Table 5.1.3 : Test Case for Payment**

| Test cases no | Test case | Expected results | Status |
|---|---|---|---|
| 1 | Booking | All details provided correctly | Successful |
| 2 | Booking | Any one field is incorrect | Unsuccessful |
| 3 | Booking | Any field left empty | Unsuccessful |

**Table 5.1.4 : Test Case for Booking**

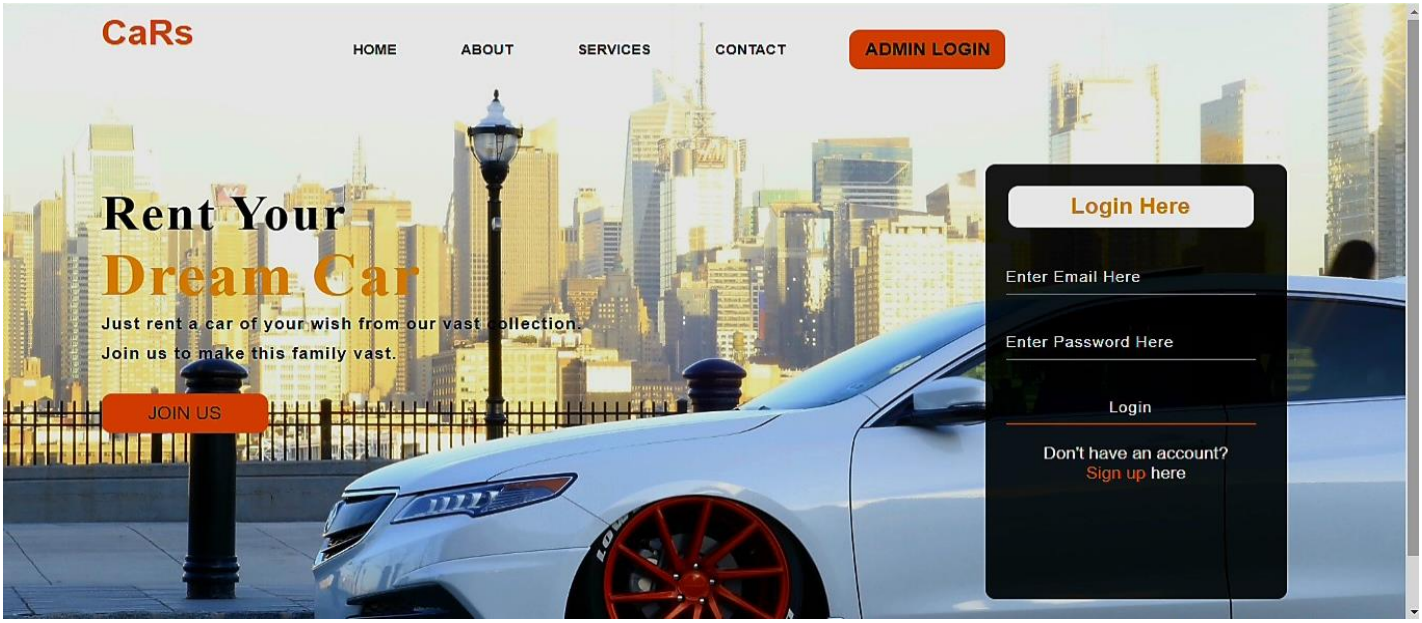
| Test cases no | Test case | Expected results | Status |
|---|---|---|---|
| 1 | Booking | All details provided correctly | Successful |
| 2 | Booking | Any one field is incorrect | Unsuccessful |
| 3 | Booking | Any field left empty | Unsuccessful |

**Table 5.1.5 : Test Case for Feedback**

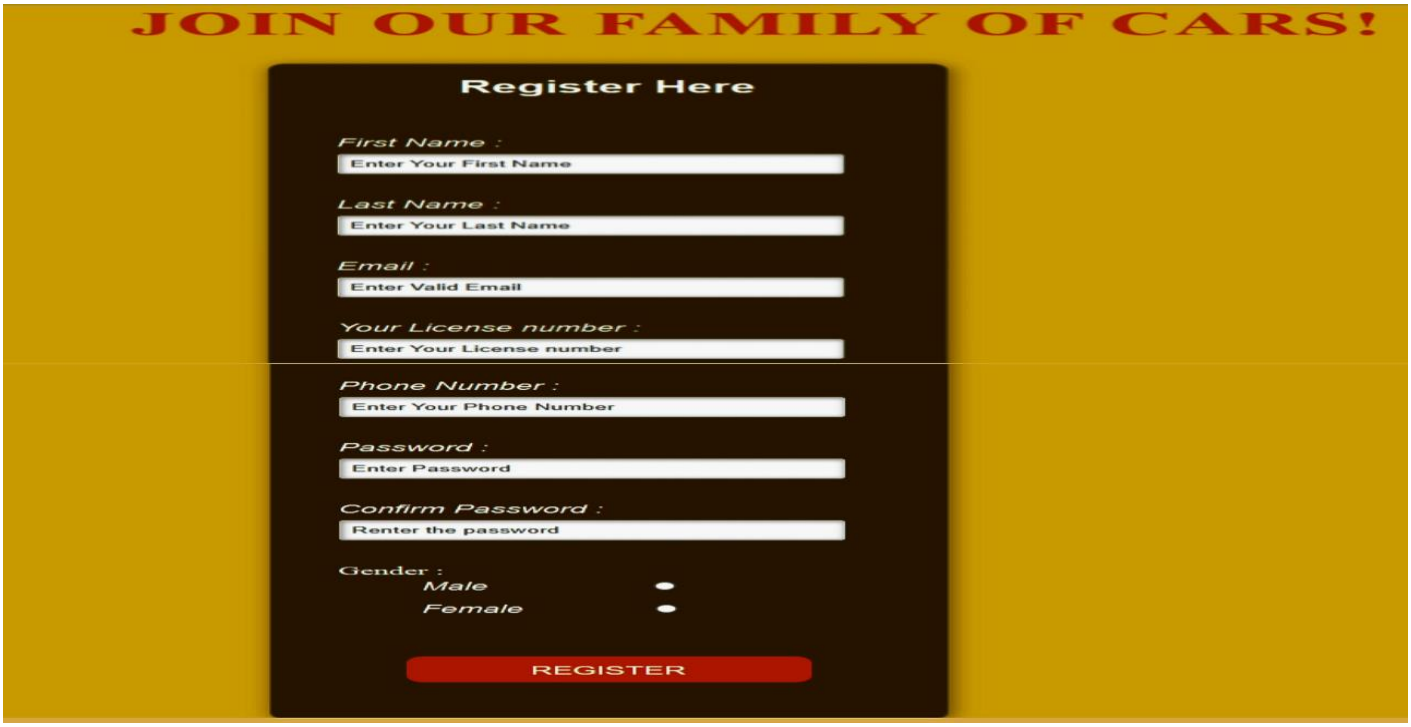
# Chapter 6

# USER MANUAL

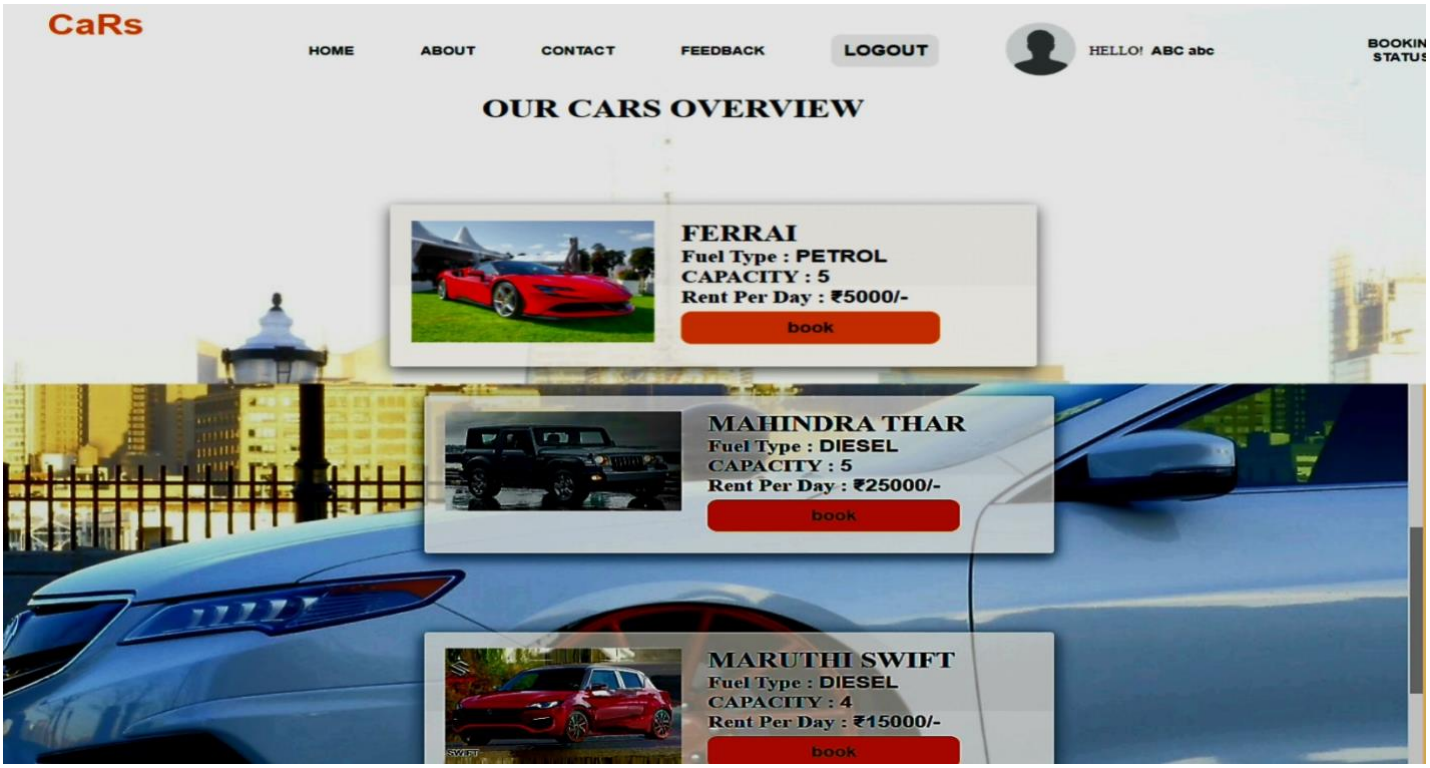
## 6.1 Snapshots of User Interfaces



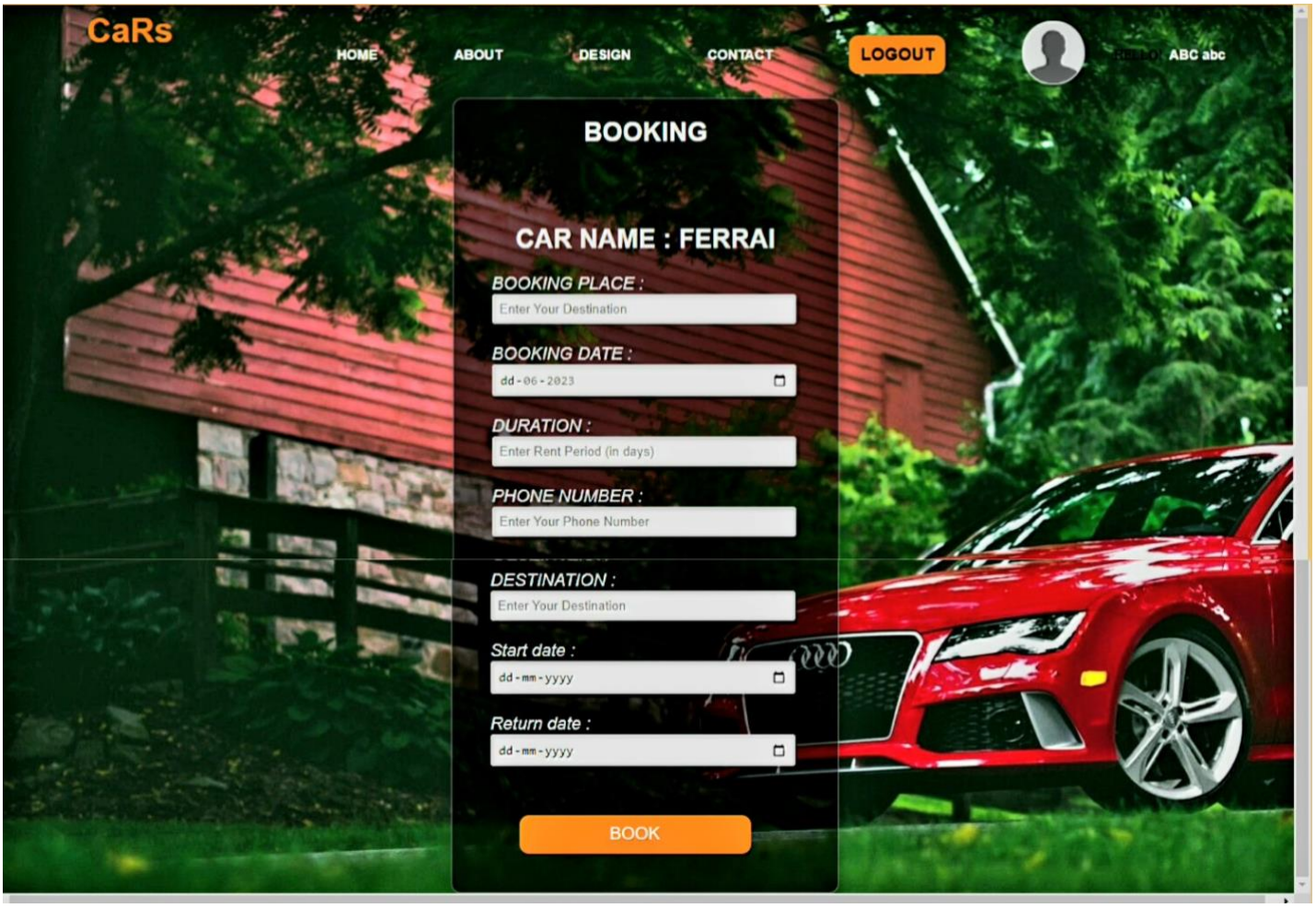

**Snapshot 6.1.1 : Home page**



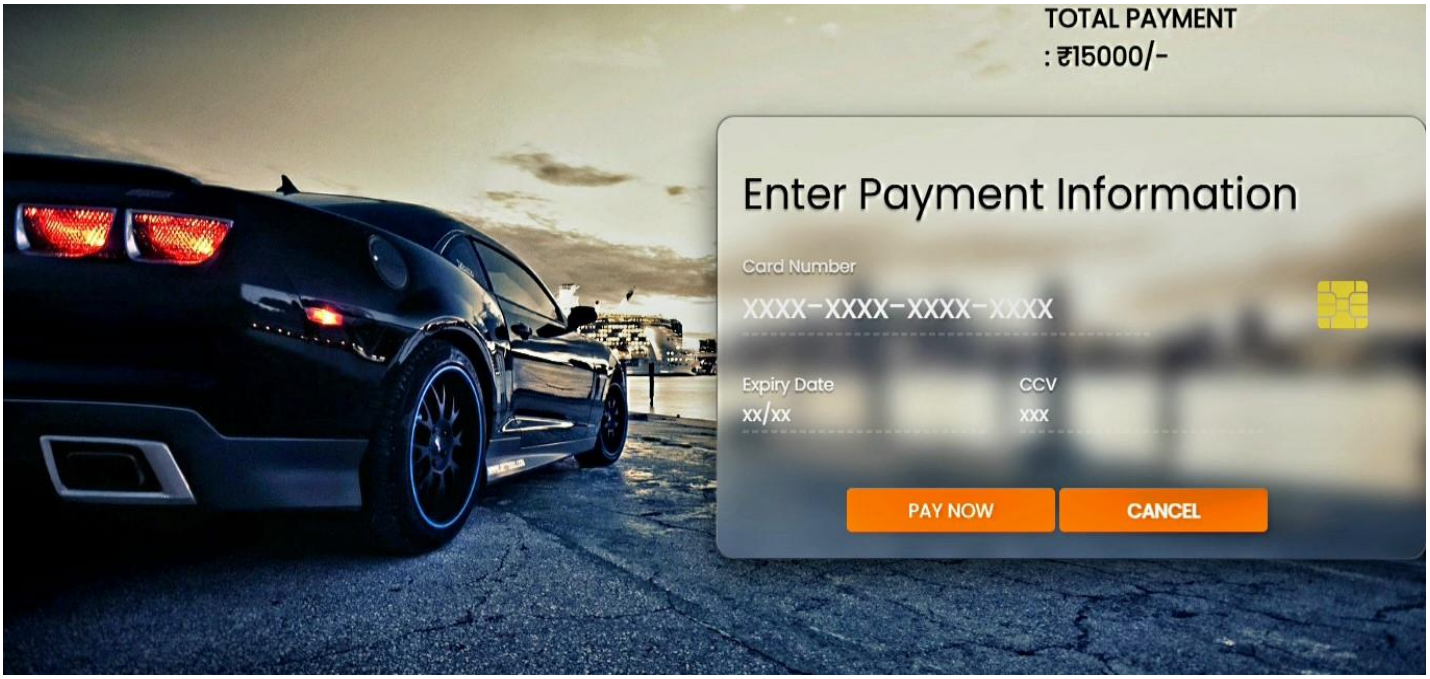

**Snapshot 6.1.2 : User Registration page**



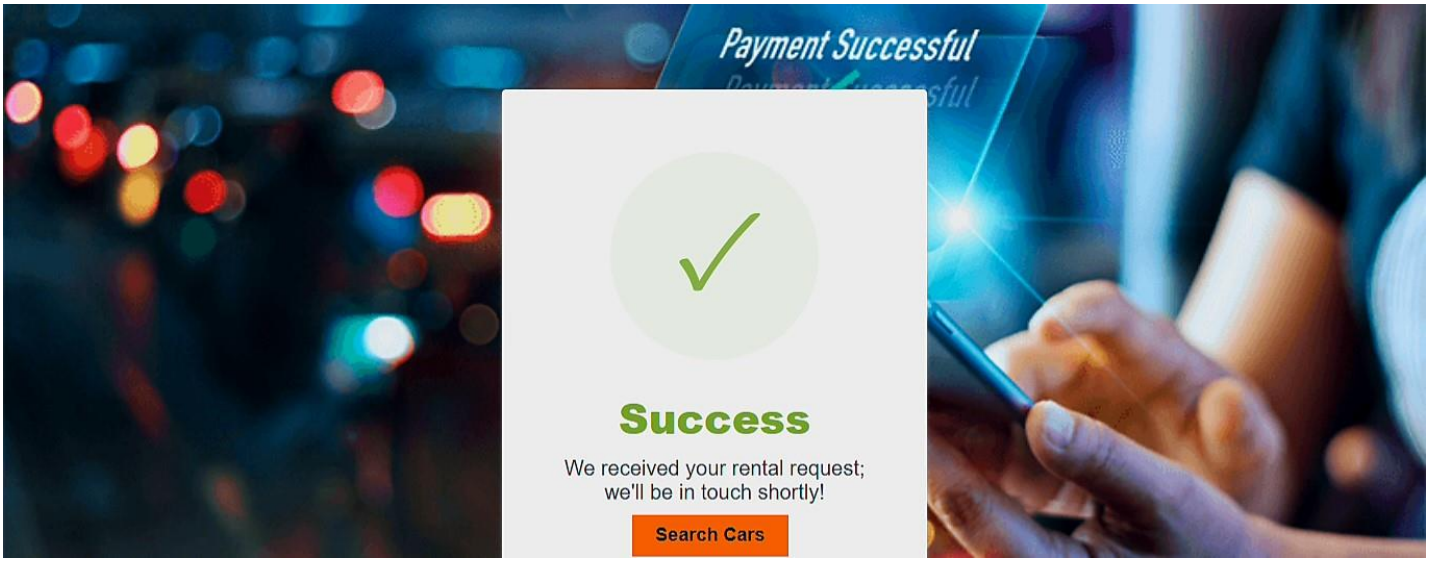

**Snapshot 6.1.3 : View Car page**



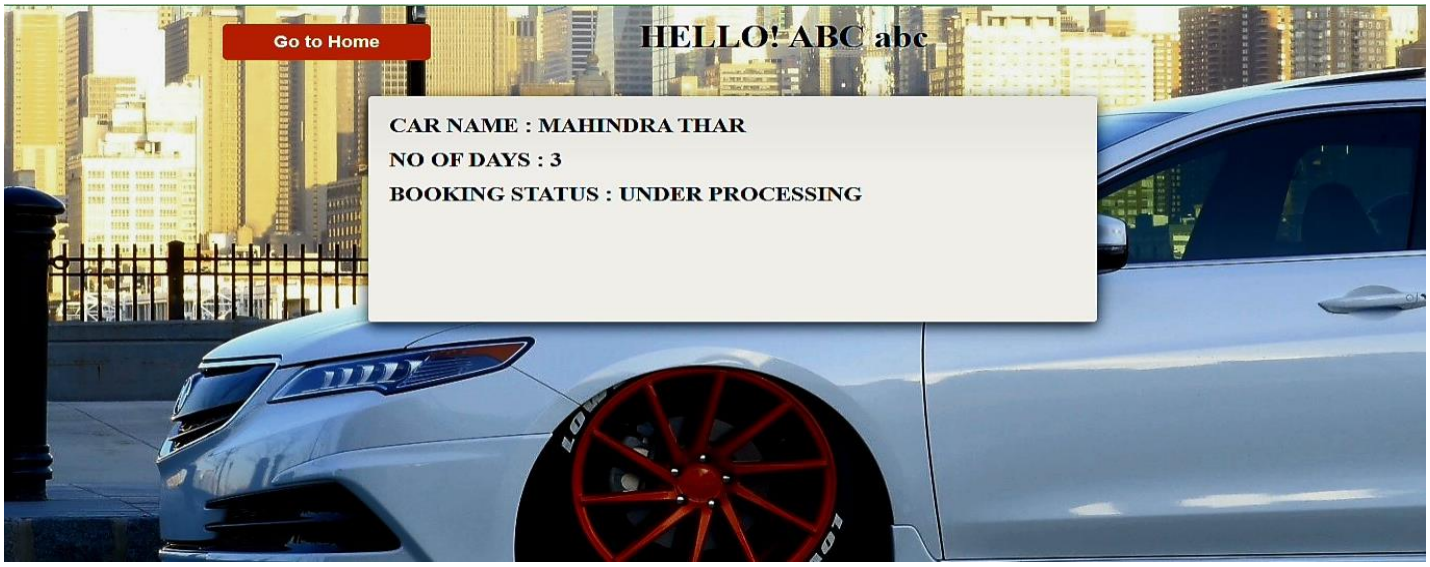

**Snapshot 6.1.4 : Booking page**



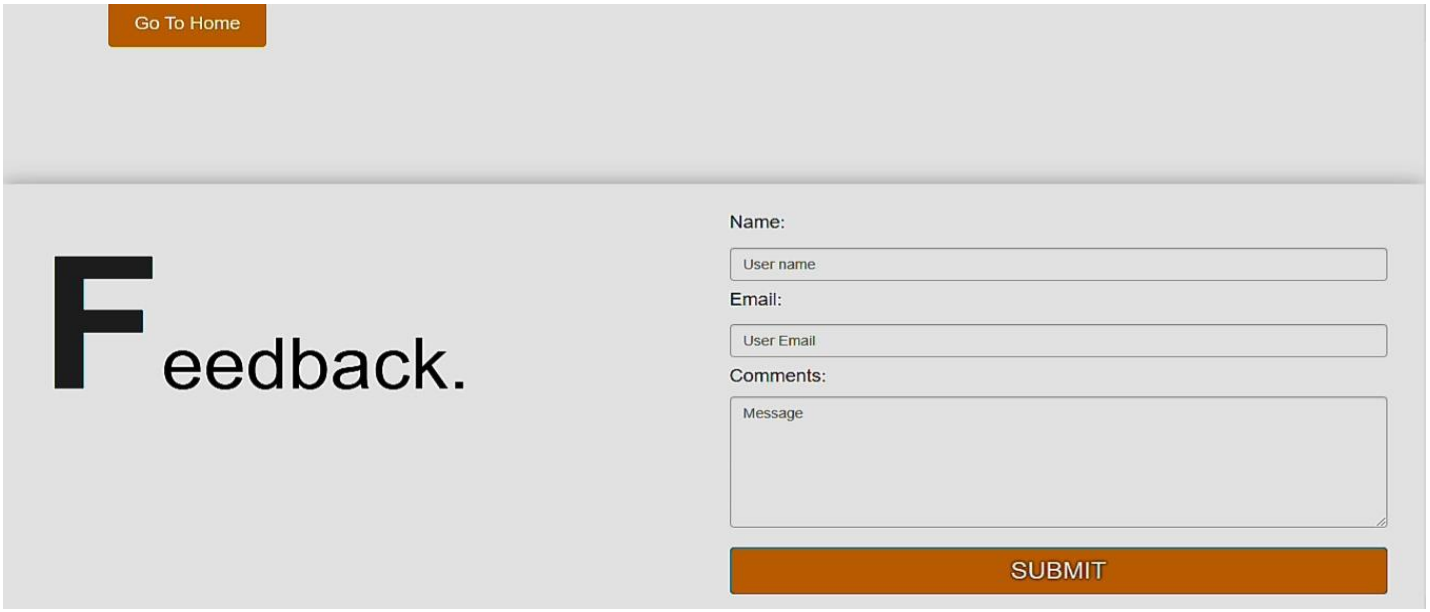

**Snapshot 6.1.5 : Payment page**




**Snapshot 6.1.6 : Payment Successful page**

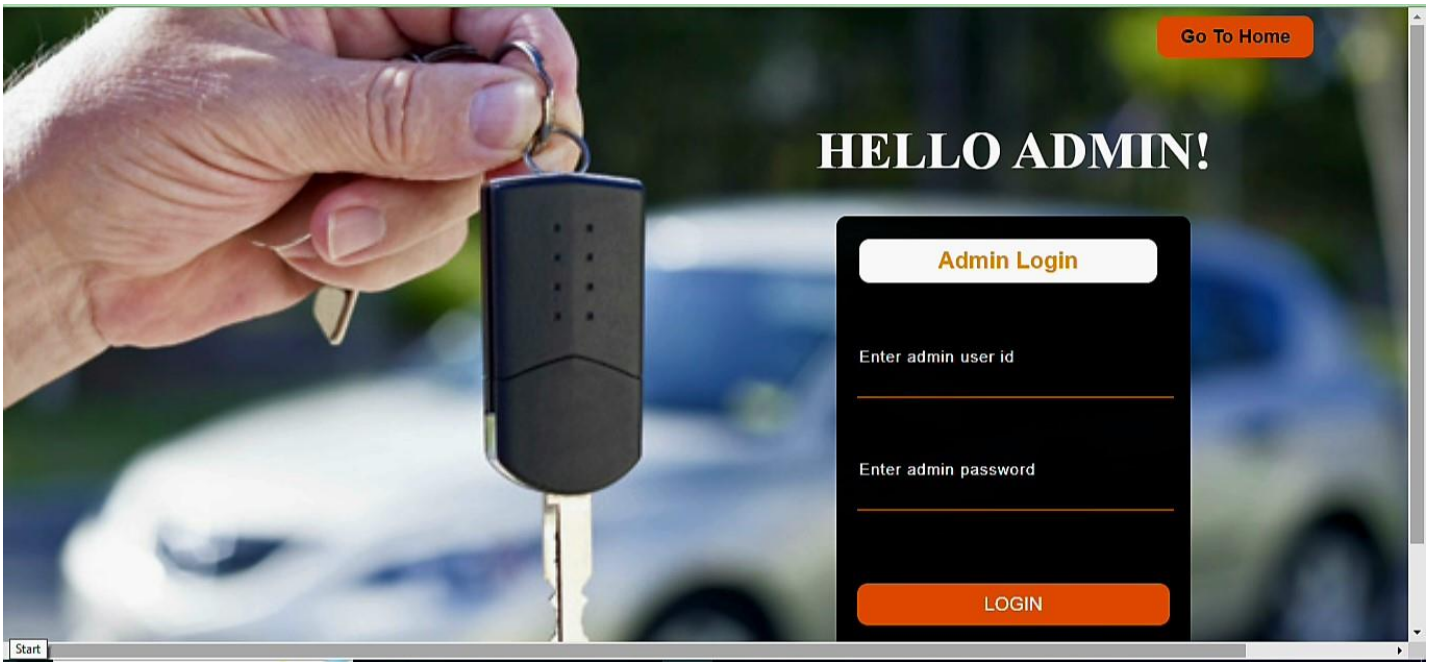



**Snapshot 6.1.7 : Booking status page**

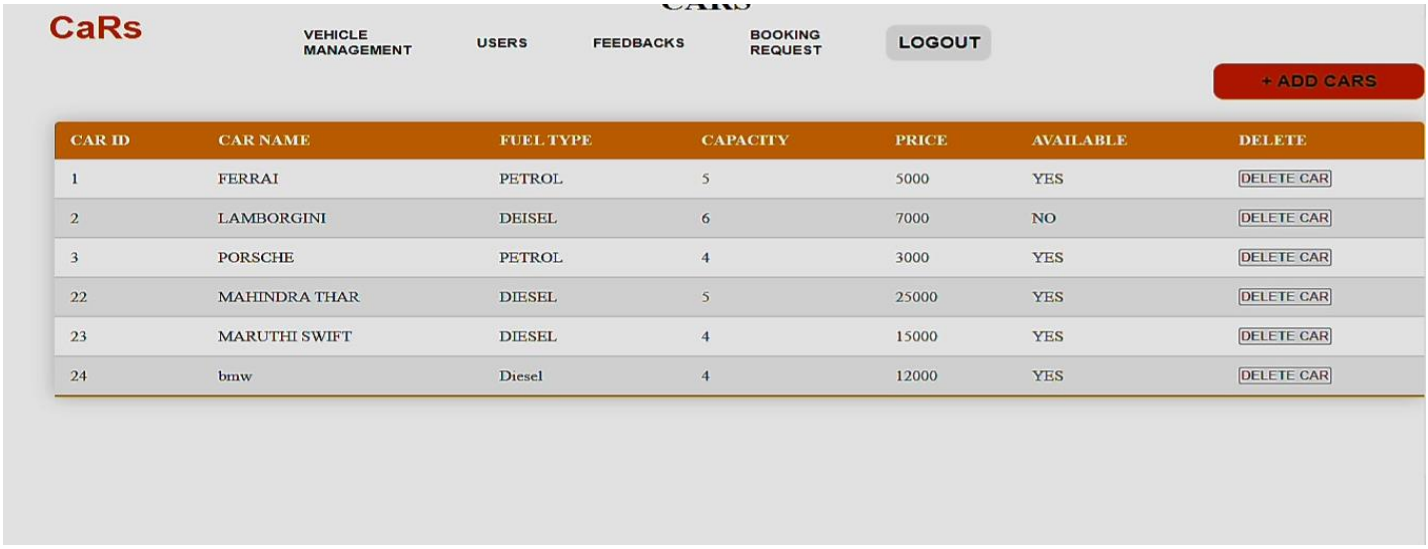



**Snapshot 6.1.8 : Feedback page**

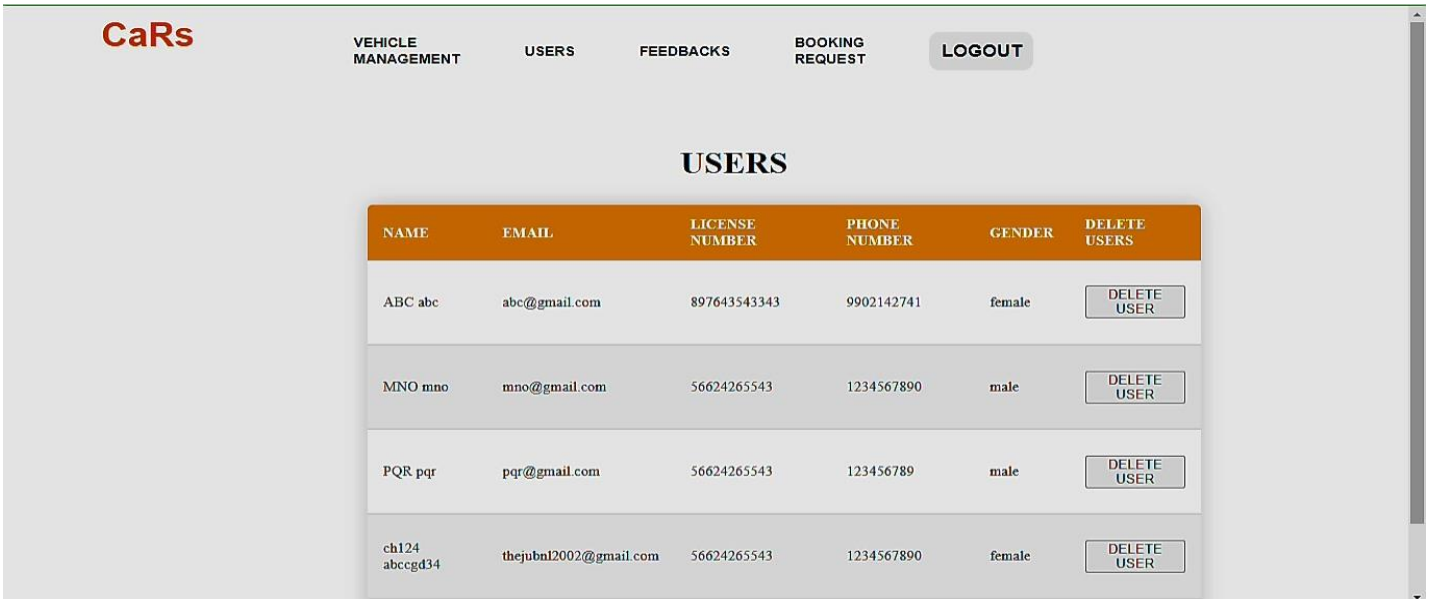

**Snapshot 6.1.9 : Contact Us page**



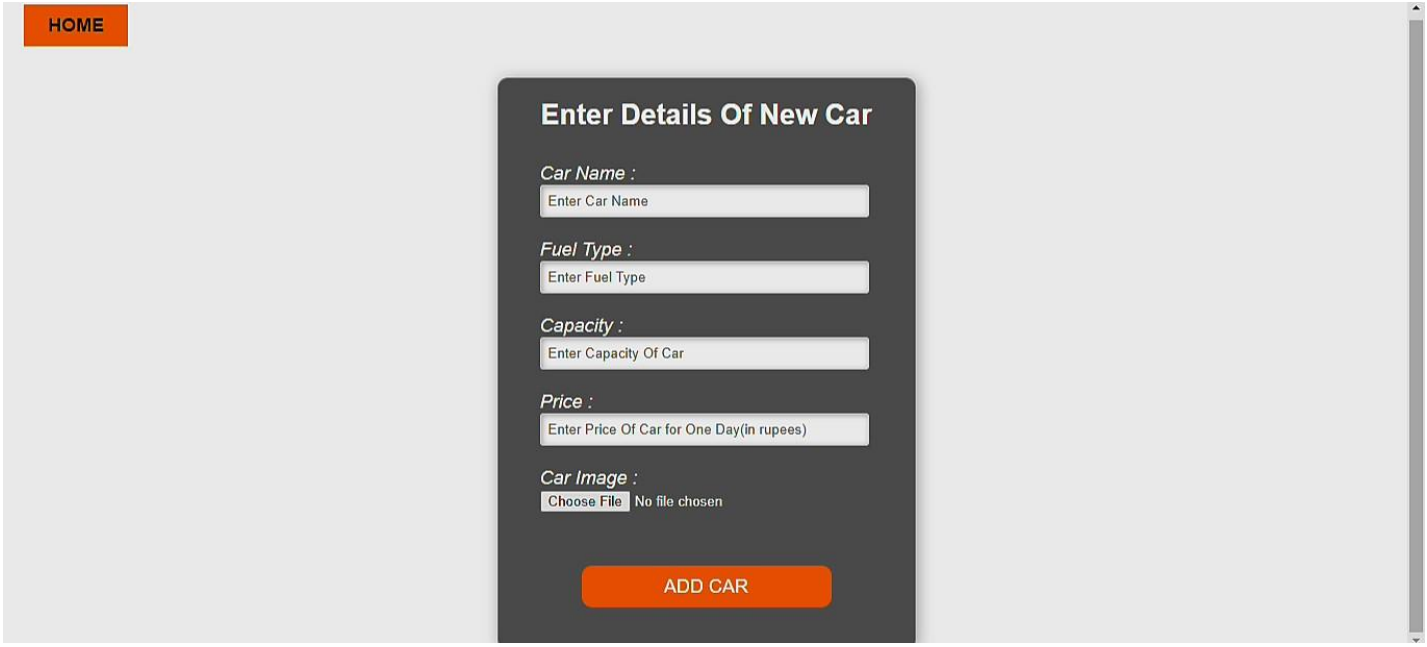

**Snapshot 6.1.10 : Admin login page**




**Snapshot 6.1.11 : Vehicle management page**



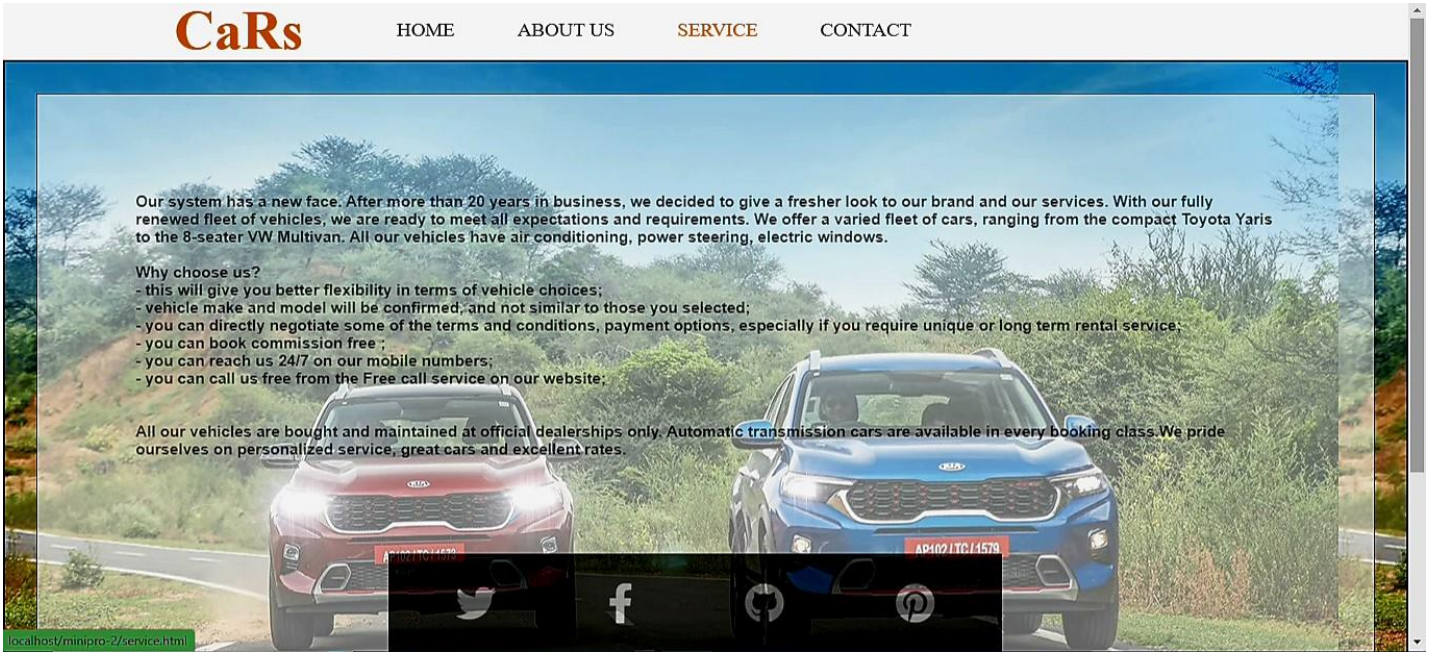

**Snapshot 6.1.12 : Users page**




**Snapshot 6.1.13 : Add car page**


| CAR ID | EMAIL | BOOK PLACE | BOOK DATE | DURATION | PHONE NUMBER | DESTINATION | RETURN DATE | BOOKING STATUS | APPROVE | CAR RETURNED |
|---|---|---|---|---|---|---|---|---|---|---|
| 22 | abc@gmail.com | Hassan | 2023-06-16 | 3 | 6362125942 | Honnavar | 2023-06-24 | UNDER PROCESSING | APPROVE | RETURNED |
| 1 | abc@gmail.com | Hassan | 2023-06-16 | 3 | 9902142741 | Honnavar | 2023-06-24 | UNDER PROCESSING | APPROVE | RETURNED |
| 1 | abc@gmail.com | Hassan | 2023-06-16 | 1993 | 9902142741 | Honnavar | 2023-06-22 | UNDER PROCESSING | APPROVE | RETURNED |
| 1 | abc@gmail.com | Hassan | 2023-06-16 | 3 | 123456789 | Honnavar | 2023-06-27 | UNDER PROCESSING | APPROVE | RETURNED |
| 1 | abc@gmail.com | Hassan | 2023-06-09 | 2 | 9902142741 | Honnavar | 2023-06-11 | RETURNED | APPROVE | RETURNED |
| 3 | pqr@gmail.com | Hassan | 2023-06-09 | 3 | 123456789 | goa | 2023-06-12 | UNDER PROCESSING | APPROVE | RETURNED |
| 2 | abc@gmail.com | Hassan | 2023-06-09 | 2 | 6362125942 | Gokarna | 2023-06-11 | UNDER PROCESSING | APPROVE | RETURNED |
| 2 | mno@gmail.com | Hassan | 2023-06-09 | 4 | 9902142741 | Gokarna | 2023-06-13 | APPROVED | APPROVE | RETURNED |

**Snapshot 6.1.14 : Booking request page**




**Snapshot 6.1.15 : Service page**

# Chapter 7

# CONCLUSION

Online Car Rental Management System is user-friendly and customized software for car renting company. Online Car Rental Management System has been developed to manage and automate the overall processing of any large car renting company. Online Car Rental Management System project is capable of managing cars, booking, feedbacks, payment etc. It is a user friendly and customized software for providing support for company admin. This project is a very flexible software and it can be upgraded according to the individual needs.

# REFERENCES


[1] Fundament also database system, Remez elmarsri Shamkanth b 7th edition,2017,Pearson.

[2] The Joy of PHP Programming, Alan Forbes 5th edition, Plum Island.

[3] http://www.carrentingsolutions.com/

[4] https://youtu.be/BsDoLVMnmZs/ Access Date and Time 12/11/21 18:30:00

[5] https://www.w3schools.com/php

[6] https://www.tutorialspoint.com/javascript/index.htm/

[7]https://youtube.com/playlist?list=PLprzxDsHXuDY89qR7Le3ocU37UBXA8fA-