

# **SRM UNIVERSITY AP**

**A MINI PROJECT REPORT  
ON**



## **CAR RENTAL SYSTEM**

**BY**

- **MOPIDEVI LOHITH – AP22110011229**
- **KETHU HIMAKAR REDDY – AP22110011215**
- **LINGAM PUSHPESH – AP22110011231**
- **GUNDA VASUDEVA AKSHAY – AP22110011223**

*In the partial fulfillment of the requirement for 4th semester*

**WEB TECHNOLOGY PROJECT (CSE210)**

**Department of Computer Science & Engineering  
SRM AP INSTITUTE OF TECHNOLOGY &ENGINEERING**

**Neerukonda,Mangalagiri - 522502**

***2023-2024***

## **ABSTRACT**

The Car Rental 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 harder. 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**

The successful completion of any significant task is the outcome of invaluable aggregate combination of different people in radial direction explicitly and implicitly. We would therefore take opportunity to thank and express our gratitude to all those without whom the completion of project would not be possible.

We express our thanks to **Mr. Kalluri Hemantha Kumar** Department of Computer Science and Engineering for having provided all the facilities that helped us in timely completion of this report.

I express my sincere gratitude to our institution and management for providing us with good infrastructure, laboratory facilities, qualified and inspiring staffs, and whose guidance was of immense help in completion of this seminar successfully.

**HIMAKAR REDDY KETHU – AP22110011215**

**LOHITH MOPIDEVI – AP22110011229**

**PUSHPESH LINGAM - AP22110011231**

**VASUDEVA AKSHAY GUNDA – AP22110011223**

## LIST OF FIGURES

| <b>SI NO</b> | <b>TITLE</b>                               | <b>PAGE NO</b> |
|--------------|--|----------------|
| Figure 3.1   | ER Diagram of Car Rental System            | 8              |
| Figure 3.2   | Schema Diagram of Car Rental System        | 9              |
| Figure 3.3   | Block Diagram of Car Rental System         | 14             |
| Figure 3.4   | Flow Chart of Car Rental System            | 15             |
| Figure 4.1.1 | Code Snippet of Connection Module          | 18             |
| Figure 4.1.2 | Code Snippet of Delete Car Module          | 18             |
| Figure 4.1.3 | Code Snippet of Uploading Car Image Module | 19             |
| Figure 4.1.4 | Code Snippet of Approve Module             | 20             |
| Figure 6.1   | Screenshot of Home page                    | 24             |
| Figure 6.2   | Screenshot of User Register page           | 25             |
| Figure 6.3   | Screenshot of User Login page              | 25             |
| Figure 6.4   | Screenshot of View Car page                | 26             |
| Figure 6.5   | Screenshot of Booking page                 | 27             |
| Figure 6.6   | Screenshot of Payment page                 | 27             |
| Figure 6.7   | Screenshot of Payment successful page      | 28             |
| Figure 6.8   | Screenshot of Booking Status page          | 28             |
| Figure 6.9   | Screenshot of Feedback page                | 29             |
| Figure 6.10  | Screenshot of Contact Us page              | 29             |
| Figure 6.11  | Screenshot of Admin Login page             | 30             |
| Figure 6.12  | Screenshot of Vehicle Management page      | 30             |
| Figure 6.13  | Screenshot of Users page                   | 31             |
| Figure 6.14  | Screenshot of Add Car page                 | 31             |
| Figure 6.15  | Screenshot of Bookings page                | 32             |

## LIST OF TABLES

| <b>SI NO</b>     | <b>TITLE</b>                  | <b>PAGE NO</b> |
|------------------|-------------------------------|----------------|
| <b>Table 5.1</b> | <b>Test Case for Login</b>    | <b>22</b>      |
| <b>Table 5.2</b> | <b>Test Case for Register</b> | <b>22</b>      |
| <b>Table 5.3</b> | <b>Test Case for Payment</b>  | <b>23</b>      |
| <b>Table 5.4</b> | <b>Test Case for Booking</b>  | <b>23</b>      |
| <b>Table 5.5</b> | <b>Test Case for Feedback</b> | <b>23</b>      |

# TABLE OF CONTENTS

|                      |   |
|----------------------|---|
| Abstract.....        | 1 |
| Acknowledgement..... | 2 |

| CHAPTER | CONTENT                                | PAGE NO |
|---------|--|---------|
| 1       | Introduction                           | 1       |
| 2       | Requirement Analysis and Specification | 5       |
| 3       | System Design                          | 8       |
| 4       | Implementation                         | 17      |
| 5       | Testing                                | 21      |
| 6       | Snapshots                              | 24      |
| 7       | Conclusion                             | 33      |
|         | References                             | 34      |

## Chapter 1

### INTRODUCTION

Web technology encompasses the tools and methods for facilitating communication between computers via markup languages and multimedia packages. It enables interaction with hosted information, primarily through websites. Key components include HTML and CSS. Data in web technology refers to meaningful information stored on servers and accessed when users visit web pages. The aim is to create and update web pages efficiently. Technologies like HTML, CSS, and JavaScript define structures and provide mechanisms for data manipulation. Ensuring data safety despite system crashes or unauthorized access is crucial. SSL, TLS, and authentication protocols play roles in this. Web browsers like Chrome, Firefox, and Edge are used to access data stored on servers, transmitted via HTTP. In summary, web technology enables building, communicating, and maintaining internet-based applications and data. It ensures smooth operation, secure data transfer, and accurate representation to users. 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.

#### 1.1 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 Objective

- 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.
- To ease customer's task whenever they need to rent a car.
- 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 SQL**

SQL (Structured Query Language) is a standardized programming language that's used to manage relational databases and perform various operations on the data in them. The uses of SQL include modifying database table and index structures; adding, updating and deleting rows of data; and retrieving subsets of information from within a database for transaction processing and analytics applications. Queries and other SQL operations take the form of commands written as statements -- commonly used SQL statements include select, add, insert, update, delete, create, alter and truncate. MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack (and other 'AMP' stacks).

## **1.5 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: Hyper Text Preprocess. PHP code is interpreted by a webserver with a PHP processor module, which generates the resulting web page. PHP commands can be embedded directly into an 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 graphical applications. PHP is free software released under the PHP License. PHP can be deployed on most web servers and also as a standalone shell on almost every operating system and platform, free of charge.

## **1.6 HTML5**

HTML5 is a markup language used for structuring and presenting content on the World Wide Web. It is the fifth and last major HTML version that is a World Wide Web Consortium (W3C) recommendation. The current specification is known as the HTML Living Standard. It is maintained by the Web Hypertext Application Technology Working Group (WHATWG), a consortium of the major browser vendors (Apple, Google, Mozilla, and Microsoft).

## **1.7 CSS**

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility; provide more flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, which reduces complexity and repetition in the structural content; and enable the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

## **1.8 JAVASCRIPT**

JavaScript(JS) is a dynamic computer programming language. It is most commonly used as part of web browsers, whose implementations allow client side scripts to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed. It is also being used in server-side programming, game development and the creation of desktop and mobile applications. JavaScript is a prototype-based scripting language with dynamic typing and has first-class functions. Its syntax was influenced by C. The key design principles within JavaScript are taken from the Self and Scheme programming languages. It is a multiparadigm language, supporting object-oriented, imperative, and functional programming styles.

## **1.9 APACHE WEB SERVER**

In this project apache server is used to parse and execute PHP pages, before deploying websites on the server, the website should be tested at the developer side to get a feel of how the website will work on actual server. Therefore apache server is like a local server on the developer side, apache server should be informed about the environment on which it should work. In our project apache server is configured to work with PHP, in this way all the PHP pages are parsed and executed by the server. When apache is installed on the system, then its services are controlled by apache service monitor.

## Chapter 2

### REQUIREMENT ANALYSIS AND SPECIFICATION

#### 2.1 Functional Requirements Car Rental 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 the software and the general approach taken by the organization when writing requirements. When expressed as 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 RENTAL 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.

**Payment Module:**

- User should be able to make payment by filling card number, expiry date and CVV are provided by the admin.
- After payment user will get the payment successful popup window.

**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,CSS,JavaScript,JQuery
- Back End : PHP, SQL
- Frame work : Bootstrap
- Software : Visual Studio Code, XAMPP

**2.2 Non-Functional Requirements of Car Rental 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.
- System provides security for the admin by allowing them to enter into the account with their respective ID and password.
- A user can only enter to their account by using their email and password. Only admin have privileges to update database contents which are used by the user.

**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.
- The system provides user friendly interface, any common people with little knowledge can use the system.
- System is robust, reliable and fast, provides more efficiency.

**Reliability:**

- It is the probability and percentage of the system performing without any failure for a specific number of uses or amount of time.
- Car rental system provides reliable interface as it provides data security and data safety.
- User can rely on the details present in the system, since it is provided by the admin.

**Consistency:**

- The car rental system provides consistency services, by retaining the data present in the database.
- The user gets the details that are only provided by the admin, thus achieving correctness of data 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 ER 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.

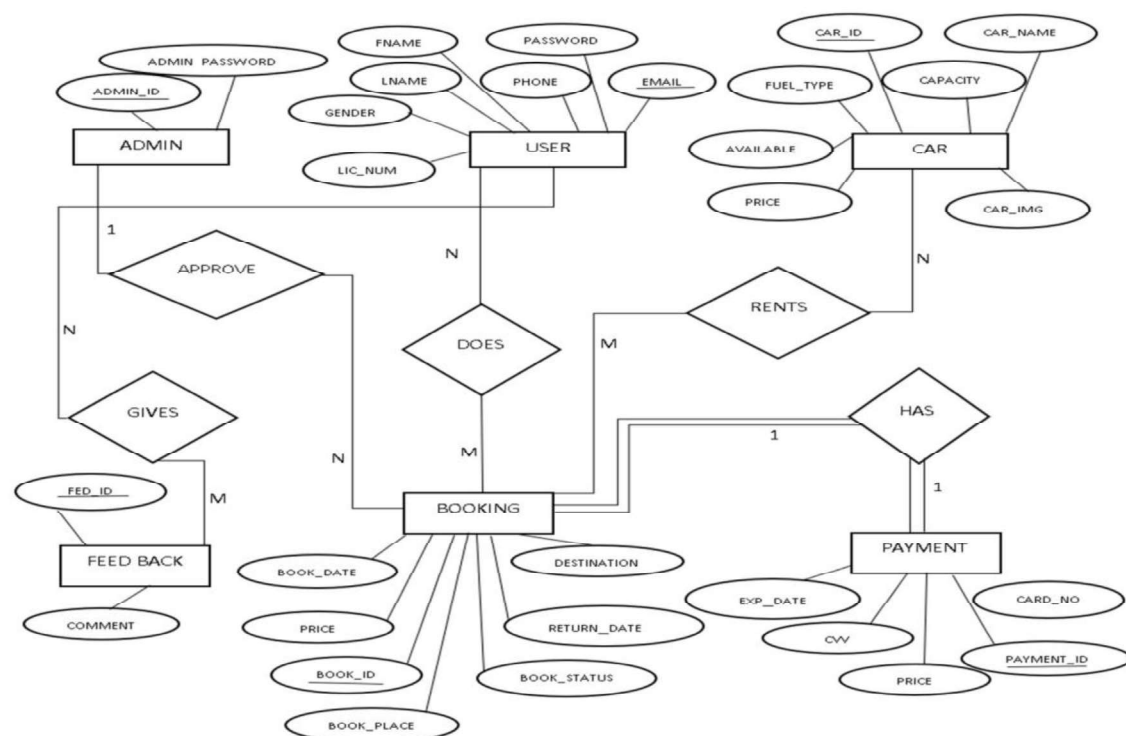


Figure 3.1 ER Diagram of Car Rental System

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.

### 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.

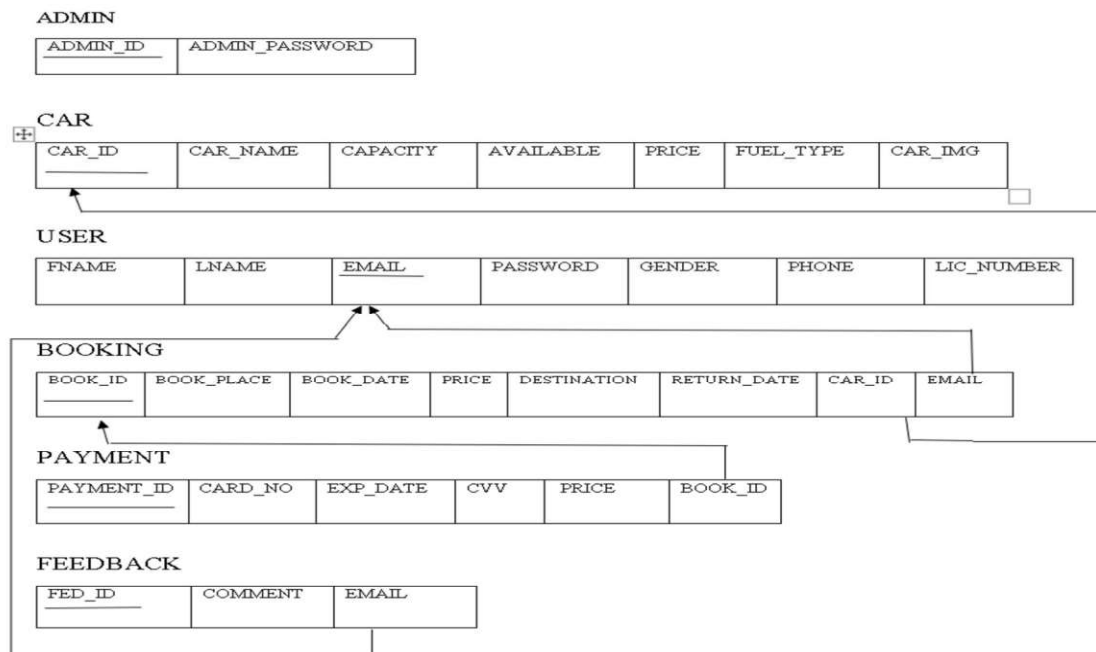


Figure 3.2 Schema Diagram of Car Rental System

### 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.

| Field name     | Data type   | Constraints | Description       |
|----------------|-------------|-------------|-------------------|
| Admin_ID       | Varchar(45) | Not Null    | Id of role        |
| Admin_Password | Varchar(45) | Not null    | Password of 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.

| Field name | Data type   | Constraints | Description         |
|------------|-------------|-------------|---------------------|
| Car_ID     | Integer     | Not null    | Id of car           |
| Car_Name   | varchar(45) | Not null    | Name of Car         |
| Capacity   | Integer     | Not null    | Capacity of Car     |
| Available  | varchar(45) | Not null    | Availability        |
| Price      | int         | Not null    | Rental Price        |
| Fuel_Type  | varchar(45) | Not null    | Type of Fuel        |
| Car_Image  | varchar(45) | Not null    | Visual 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.

| Field name        | Data type   | Constraints | Description     |
|-------------------|-------------|-------------|-----------------|
| FNAME             | varchar(45) | Not null    | First Name      |
| LNAME             | varchar(45) | Not null    | Last Name       |
| EMAIL             | varchar(45) | Not null    | Email           |
| PASSWORD          | varchar(45) | Not null    | Password        |
| GENDER            | varchar(10) | Not null    | Gender          |
| PHONE_NUMBER      | integer     | Not null    | Phone Number    |
| LICENCE<br>NUMBER | varchar(45) | Not null    | Liscence Number |

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

| Field name  | Data type   | Constraints | Description  |
|-------------|-------------|-------------|--------------|
| BOOK_ID     | Int(11)     | Primary Key | Book ID      |
| BOOK_PLACE: | varchar(45) | Not null    | Return place |
| BOOK_DATE   | date        | Not null    | Return date  |
| PRICE       | Int(11)     | Not null    | Cost         |
| DESTINATION | varchar(45) | Not null    | Destination  |
| CAR_ID      | Int(11)     | Foreign key | Car ID       |
| EMAIL       | varchar(45) | Not null    | Mail ID      |
| RETURN_DATE | Date        | Not null    | Return date  |

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.

| Field name | Data type   | Constraints | Description |
|------------|-------------|-------------|-------------|
| PAYMENT_ID | Int(11)     | Primary Key | Payment ID  |
| CARD_NO    | varchar(45) | Not null    | Card No     |
| EXP_DATE   | varchar(45) | Not null    | Exp date    |
| CVV        | Int(11)     | Not null    | CVV         |
| PRICE      | Int(11)     | Not null    | Cost        |
| BOOK_ID    | Int(11)     | Foreign Key | Book ID     |

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.

| Field name | Data type   | Constraints | Description |
|------------|-------------|-------------|-------------|
| FEED_ID    | Int(11)     | Primary Key | Feedback    |
| COMMENT    | Text        | Not null    | Comment     |
| EMAIL      | varchar(45) | Not null    | Email       |

### 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.

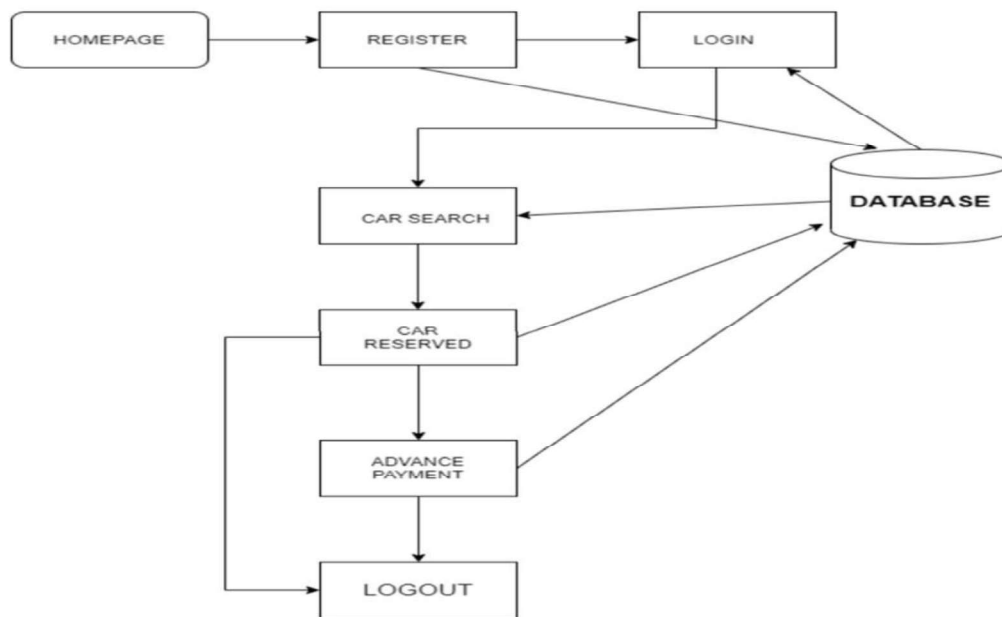


Figure 3.3 Block Diagram of Car Rental System

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.

### 3.4 Flowchart

A flowchart is a diagram that depicts a process, system or computer algorithm. They are widely used in multiple fields to document, study, plan, improve and communicate often complex processes in clear, easy-to-understand diagrams. Flowcharts, sometimes spelled as flow charts, use rectangles, ovals, diamonds and potentially numerous other shapes to define the type of step, along with connecting arrows to define flow and sequence. They can range from simple, hand drawn charts to comprehensive computer-drawn diagrams depicting multiple steps and routes.

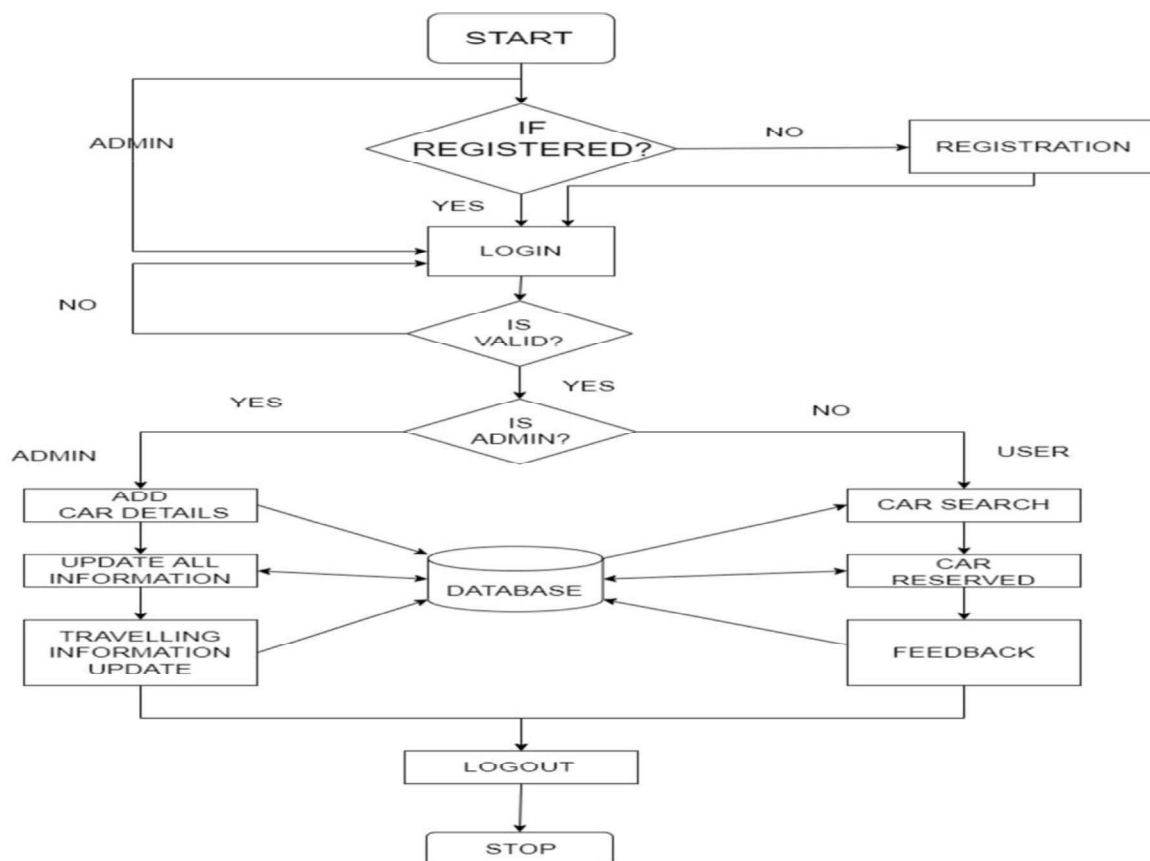


Figure 3.4 Flow Chart of Car Rental System

In the Figure 3.4 the user/admin enters his credentials for registration and later logins. If the login credentials are correct and authentic, he is redirected to his web page. If an admin is an authorized user, he is given the privilege of adding, updating the car, booking, payment details. If an authentic user logs in, he gets the details of car and other related details, then he can view list of cars then he can book car by providing valid details then he can get the booking status of booked status of after making payment. After performing all the operations, the user and admin can save and log out.

## Chapter 4

### IMPLEMENTATION

**4.1** PHP is a server-side scripting language primarily used for web development, allowing for dynamic content generation. It can be embedded into HTML or utilized with various web technologies like templates and frameworks. PHP is executed by a server-side interpreter and can also be run via command-line interface for standalone applications. HTML, the standard markup language for web pages, is used in conjunction with CSS and JavaScript to create web content. It structures web pages semantically, defining elements like headings, paragraphs, lists, and links. HTML elements are enclosed in tags, which browsers use to interpret and render the content.

#### 4.2 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 to make sure the backend implementation is free from SQL injection attacks by escaping the query value

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

**Figure 4.1.1 Code Snippet of Connection module**

Figure 4.1.1 Code snippet is used for connect to the SQL server. This required because other code snippet to connect to the database and then access the data. If the connection fails, it throws an error.

```
<?php
require_once('connection.php');
$carid=$_GET['id'];
$sql="DELETE from cars where CAR_ID=$carid";
$result=mysqli_query($con,$sql);
echo '<script>alert("CAR DELETED SUCCESFULLY")</script>';
echo '<script> window.location.href = "adminvehicle.php";</script>';
?>
```

**Figure 4.1.2 Code Snippet of Delete Car Module**

Figure 4.1.2 Code snippet indicates the code for the deletion of the car. Only the admin can delete the car.



Figure 4.1.3 Code snippet indicates the code for the adding the car image from the admin and shows it to user.

```

<?php

require_once('connection.php');
$bookid=$_GET['id'];
$sql="SELECT *from booking where BOOK_Id=$bookid";
$result=mysqli_query($con,$sql);
$res = mysqli_fetch_assoc($result);
$car_id=$res['CAR_ID'];
$sql2="SELECT *from cars where CAR_ID=$car_id";
$carres=mysqli_query($con,$sql2);
$carresult = mysqli_fetch_assoc($carres);
if($carresult['AVAILABLE']=='Y')
{
if($res['BOOK_STATUS']=='APPROVED')
{
    echo '<script>alert("ALREADY APPROVED")</script>';
    echo '<script> window.location.href = "adminbook.php";</script>';
}
}
else{
    $query="UPDATE booking set BOOK_STATUS='APPROVED' where
BOOK_ID=$bookid";
    $queryy=mysqli_query($con,$query);
    $sql2="UPDATE cars set AVAILABLE='N' where CAR_ID=$res[CAR_ID]";
    $query2=mysqli_query($con,$sql2);
    echo '<script>alert("APPROVED SUCCESSFULLY")</script>';
    echo '<script> window.location.href = "adminbook.php";</script>';
}
}
else{
    echo '<script>alert("CAR IS NOT AVAILABLE")</script>';
    echo '<script> window.location.href = "adminbook.php";</script>';
}
?>

```

**Figure 4.1.4 Code Snippet of Approve Module**

Figure 4.1.4 Code snippet indicates the code for approving the car registered by the user and sends confirmation email to user.

## **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 Testing process**

Best testing process is to test each subsystem separately, as we have done in project. Best done during implementation. Best done after small sub-steps of the implementation rather than large chunks. Once each lowest level unit has been tested, units are combined with related units and retested in combination. This proceeds hierarchically bottom-up until the entire system is tested as a whole. Typical levels of testing:

- Module- package, abstract data type, class.
- Sub-system- collection of related modules, cluster of classes, method-message paths.
- Acceptance testing- whole system with real data (involve customer, user) Alpha

testing is acceptance testing with a single client (common for bespoke systems).

Beta testing involves distributing system to potential customers to use and provide feedback. In this project, beta testing has been followed. This exposes system to situations and errors that might not be anticipated by us.

##### **5.1.1 Unit testing**

Unit testing is the process of testing individual software components unit or modules. Since it needs the detailed knowledge of the internal program design and code this task is done by the programmer and not by testers.

##### **5.1.2 Integration Testing**

Integration testing is another aspect of testing that is generally done in order to uncover errors associated with the flow of data across interfaces. The unit-tested modules are grouped together and tested in small segment, which makes it easier to isolate and correct errors. This approach is continued until we have integrated all modules to form the system as a whole.

After the completion of each module it has been combined with the remaining module to ensure that the project is working properly as expected.

### **5.1.3 System Testing**

System testing tests a completely integrated system to verify that it meets its requirements. After the completion of the entire module they are combined together to test whether the entire project is working properly.

## **5.2 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.

| <b>Test cases no</b> | <b>Test Case</b>     | <b>Expected results</b>             | <b>Status</b> |
|----------------------|----------------------|-------------------------------------|---------------|
| 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 Test Case for Login**

Table 5.1 represents the test case for login module. It shows both successful and unsuccessful results for the test cases.

| <b>Test cases no</b> | <b>Test Case</b>          | <b>Expected results</b>        | <b>Status</b> |
|----------------------|---------------------------|--------------------------------|---------------|
| 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 5.2 Test Case for Signup**

Table 5.2 represents the test case for sign up module. It shows both successful and unsuccessful results for the test cases.

| 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 5.3 Test Case for Payment**

Table 5.3 represents the test case for Payment module. It shows both successful and unsuccessful results for the test cases.

| 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 5.4 Test Case for Booking**

Table 5.4 represents the test case for Booking module. It shows both successful and unsuccessful results for the test cases.

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

**Table 5.5 Test Case for Feedback**

Table 5.5 represents the test case for Feedback module. It shows both successful and unsuccessful results for the test cases.

## Chapter 6

### SNAPSHOTS

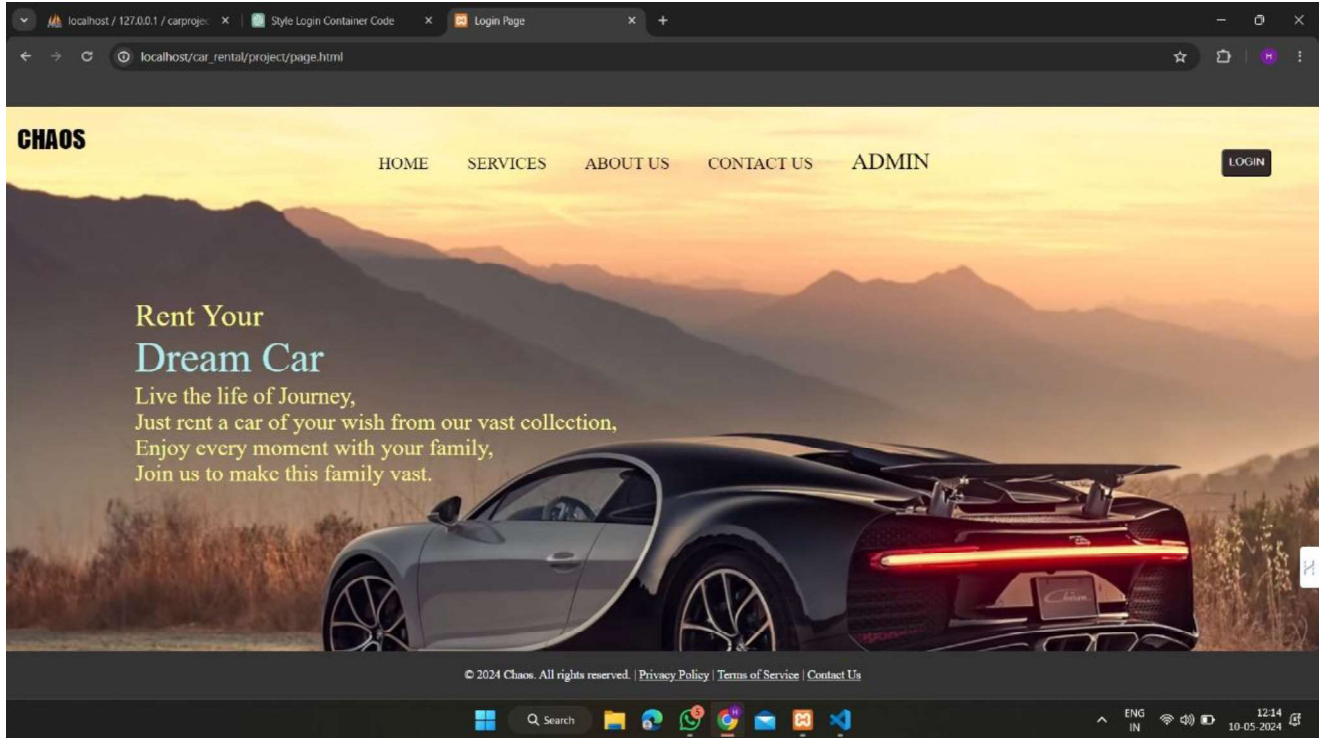
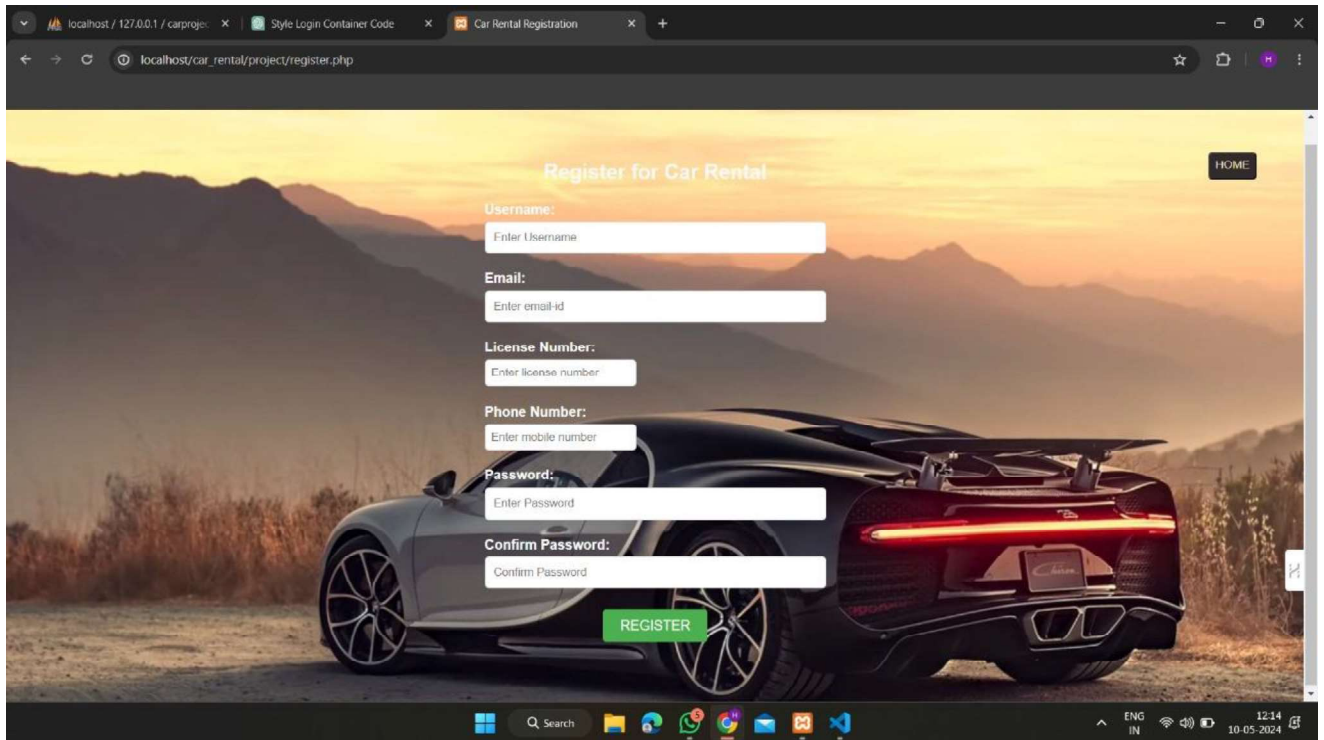


Figure 6.1 Screenshot of Home page

Figure 6.1 indicates the home page of our website CaRs. This contains navigation bar, through which can navigate to other pages. It also contains some details about the website at the home page.



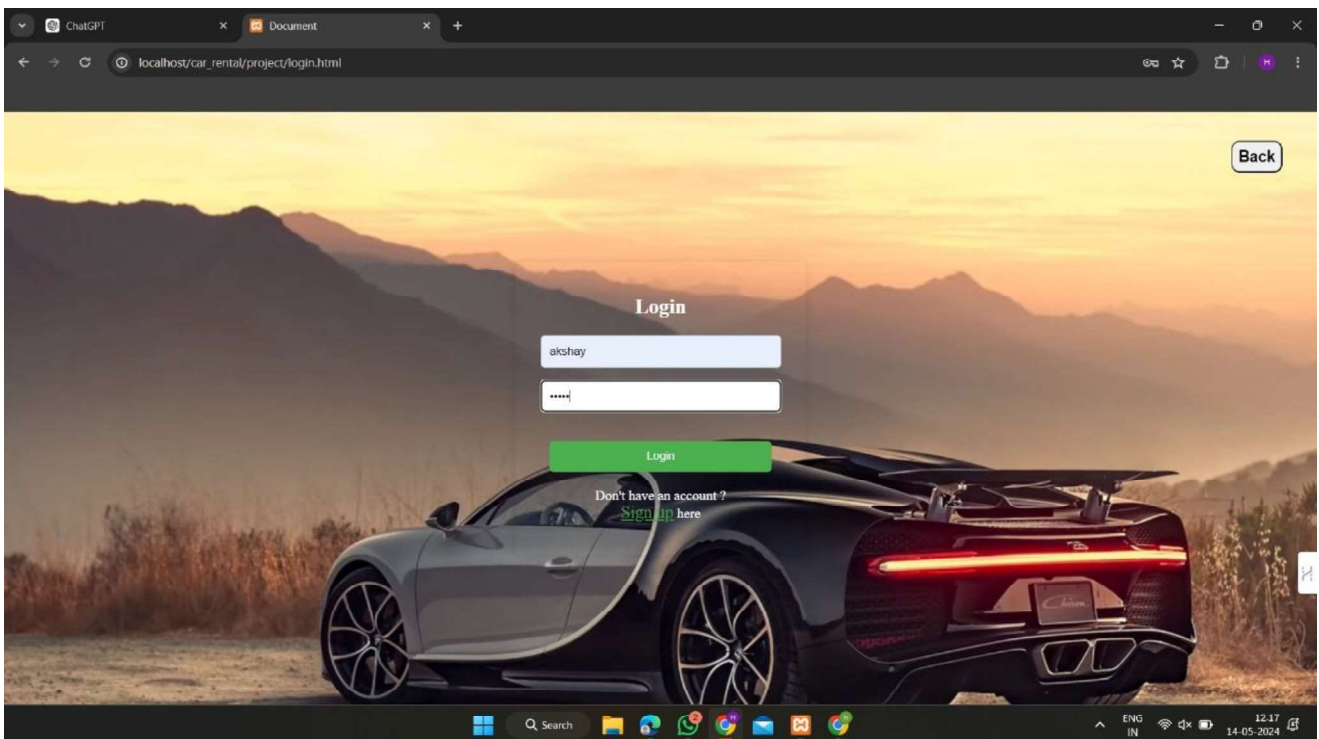
The screenshot shows a web browser window with the URL `localhost/car_rental/project/register.php`. The page is titled "Register for Car Rental" and has a "HOME" button in the top right. The registration form consists of the following fields and labels:

- Username: Enter Username
- Email: Enter email-id
- License Number: Enter license number
- Phone Number: Enter mobile number
- Password: Enter Password
- Confirm Password: Confirm Password

A green "REGISTER" button is located below the password fields. The background image shows a dark sports car with its rear light bar illuminated, parked in a desert-like landscape at sunset.

**Figure 6.2 Screenshot of User Registration page**

Figure 6.2 indicates the user registration page. It asks the user to enter the details like first name, last name, email, license number, phone number, password, confirm password and gender.



The screenshot shows a web browser window with the URL `localhost/car_rental/project/login.html`. The page is titled "Login" and has a "Back" button in the top right. The login form consists of the following fields and labels:

- Email: akshay
- Password: [masked with dots]

A green "Login" button is located below the password field. Below the password field, there is a link that says "Don't have an account? Sign up here". The background image is the same as in Figure 6.2, showing a dark sports car at sunset.

**Figure 6.3 Screenshot of User Login page**

Figure 6.3 indicates user login page. It asks the user to enter the email and password in order to enter the website CaRs.

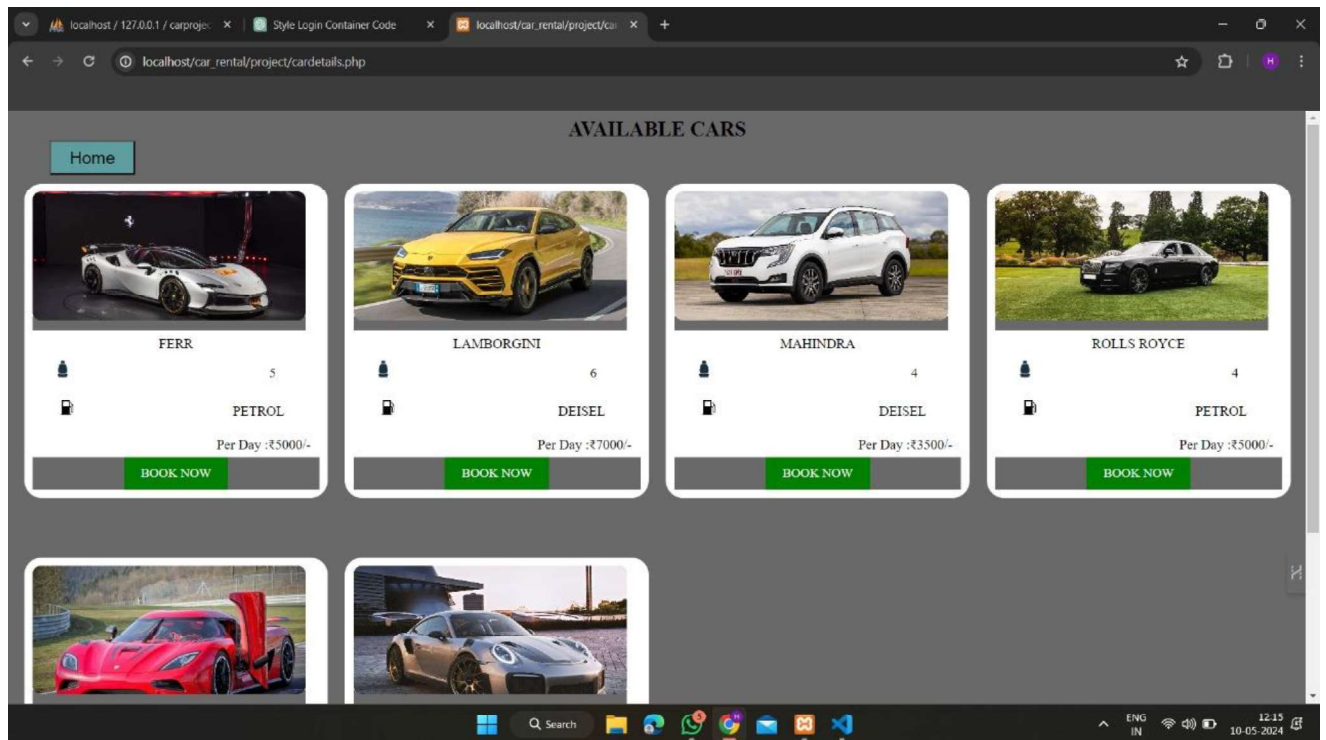


Figure 6.4 Screenshot of View Car page

Figure 6.4 indicates view car page. It asks the user to choose the car it contains the details like fuel type, capacity, rent per day.



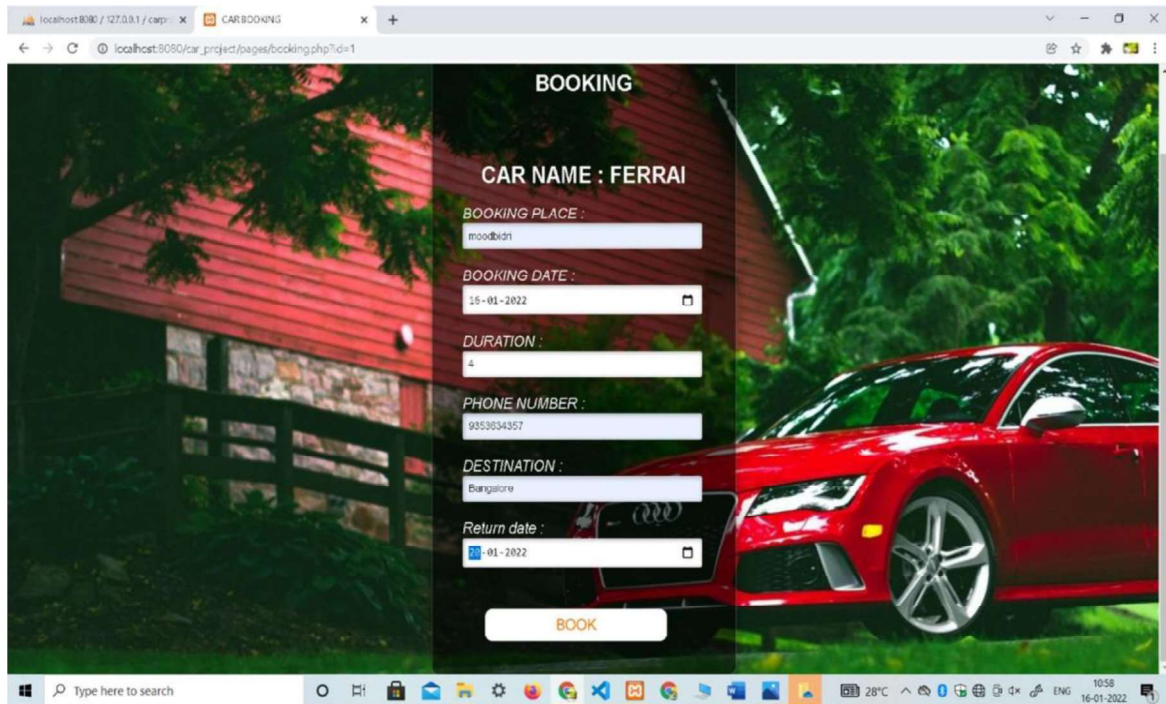


Figure 6.5 Screenshot of Booking page

Figure 6.5 indicates the booking page. It asks the user to enter booking place, booking date, duration, phone number, destination, return date.

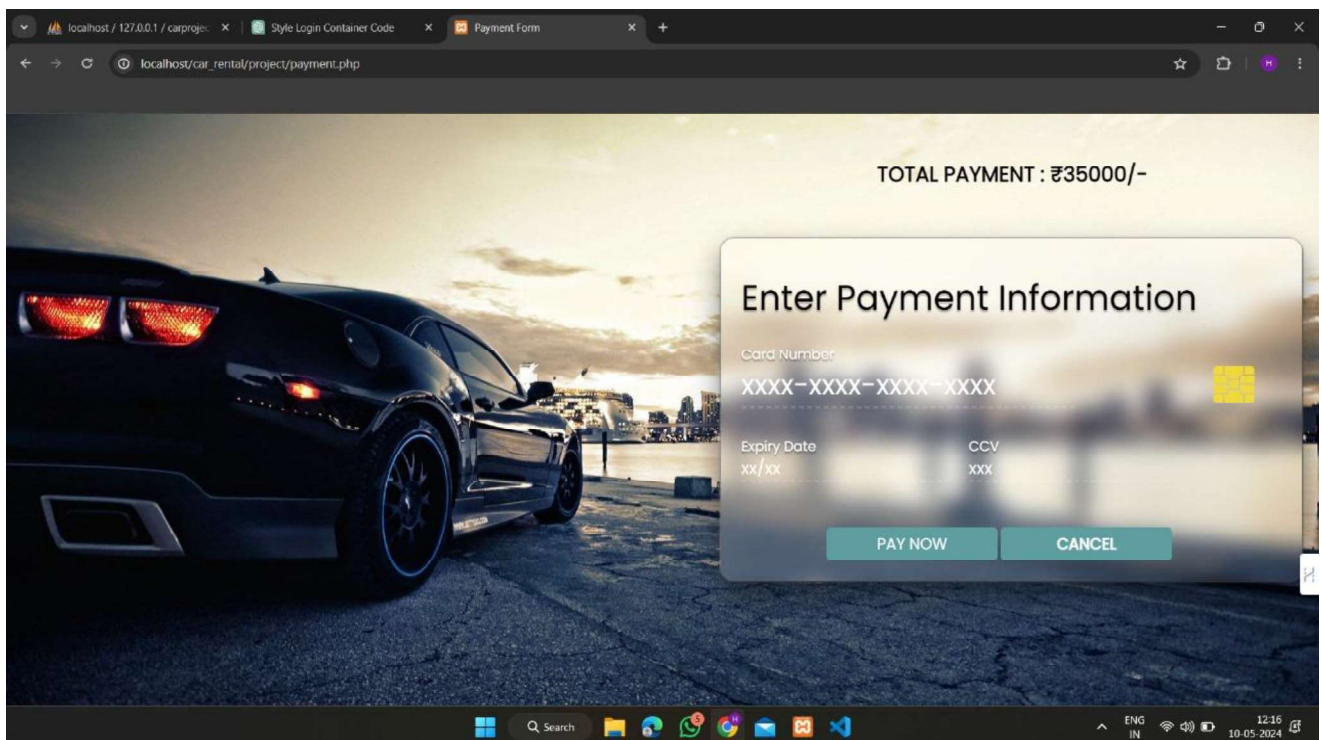
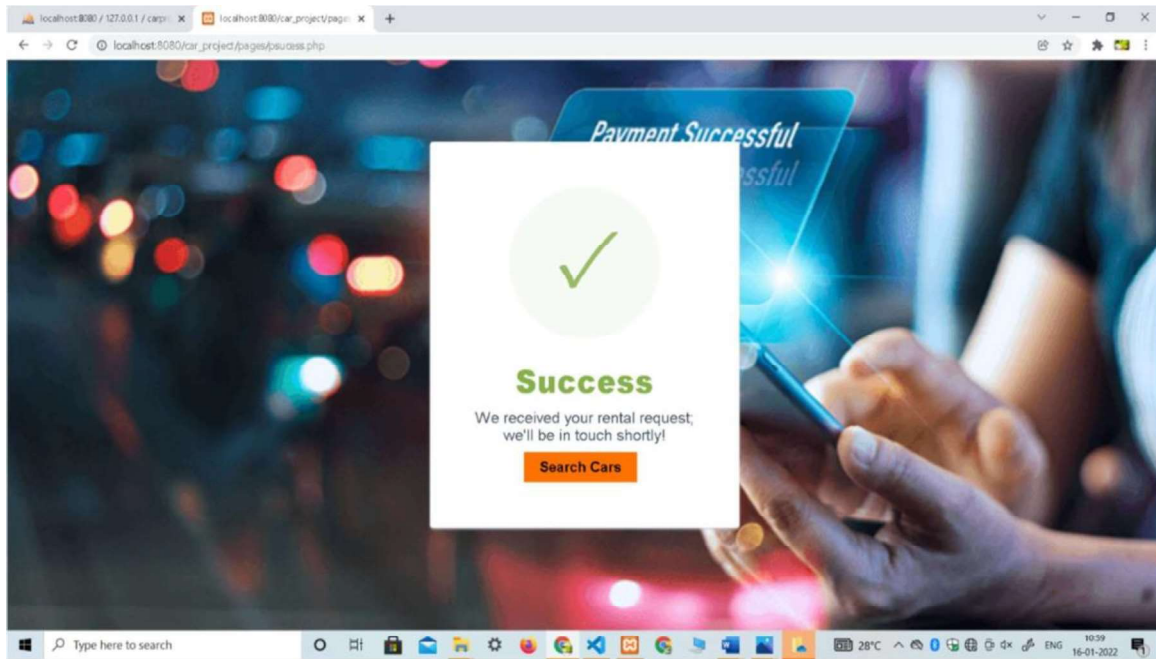


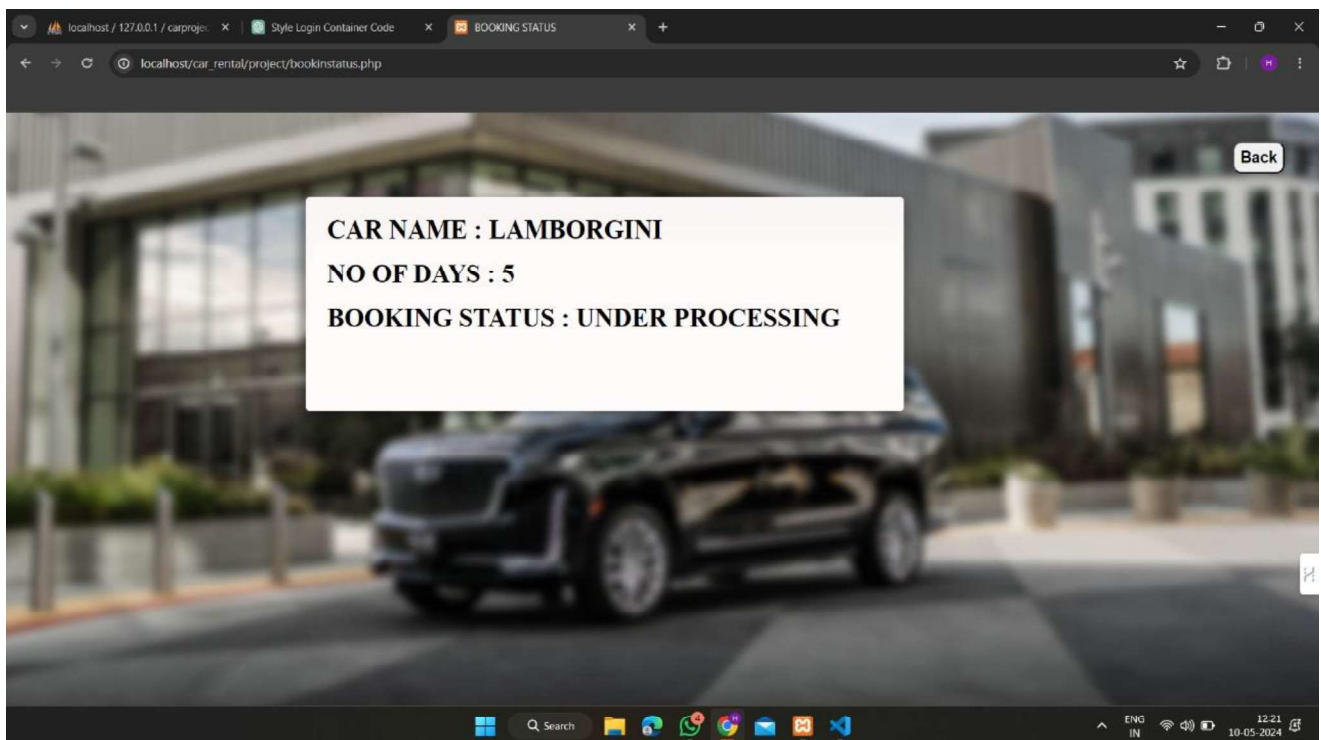
Figure 6.6 Screenshot of Payment page

Figure 6.6 indicates Payment page. It asks the user to enter details like card number, expiry date, ccv.



**Figure 6.7 Screenshot of Payment Successful page**

Figure 6.7 indicates payment successful page. After paying the payment by providing valid information if the payment is successful it shows pop up message that payment is successful.



**Figure 6.8 Screenshot of Booking Status page**

Figure 6.8 indicates bookig status page. It shows user booking details.

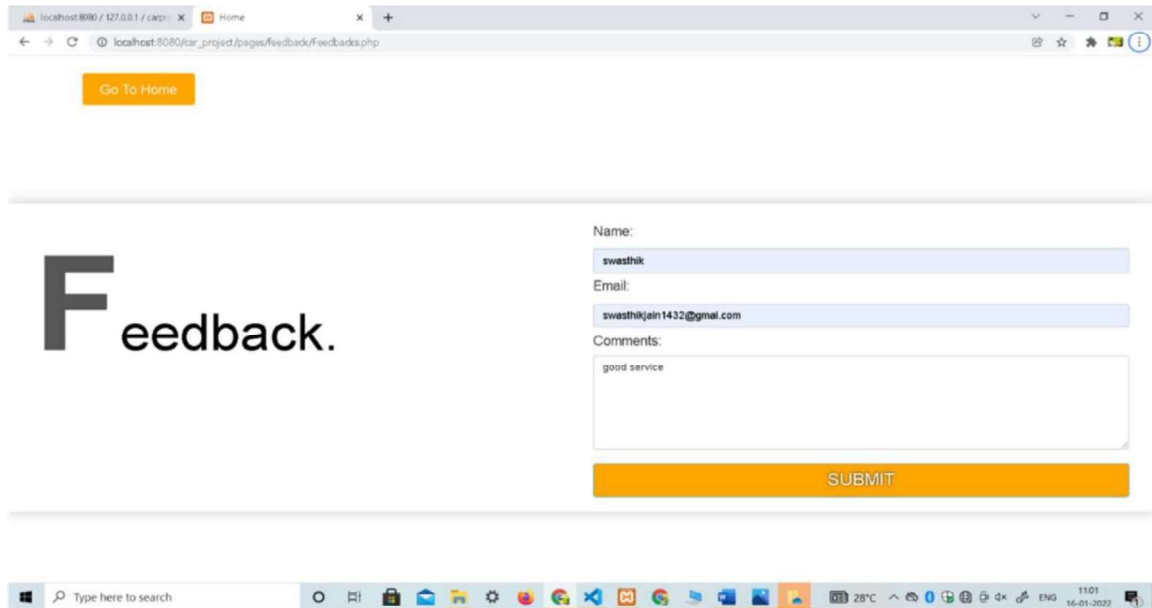


Figure 6.9 Screenshot of Feedback Page

Figure 6.9 indicates Feed back page.It asks the user enter details like name,email,comments where registered user can give their valuable feed of their experience .

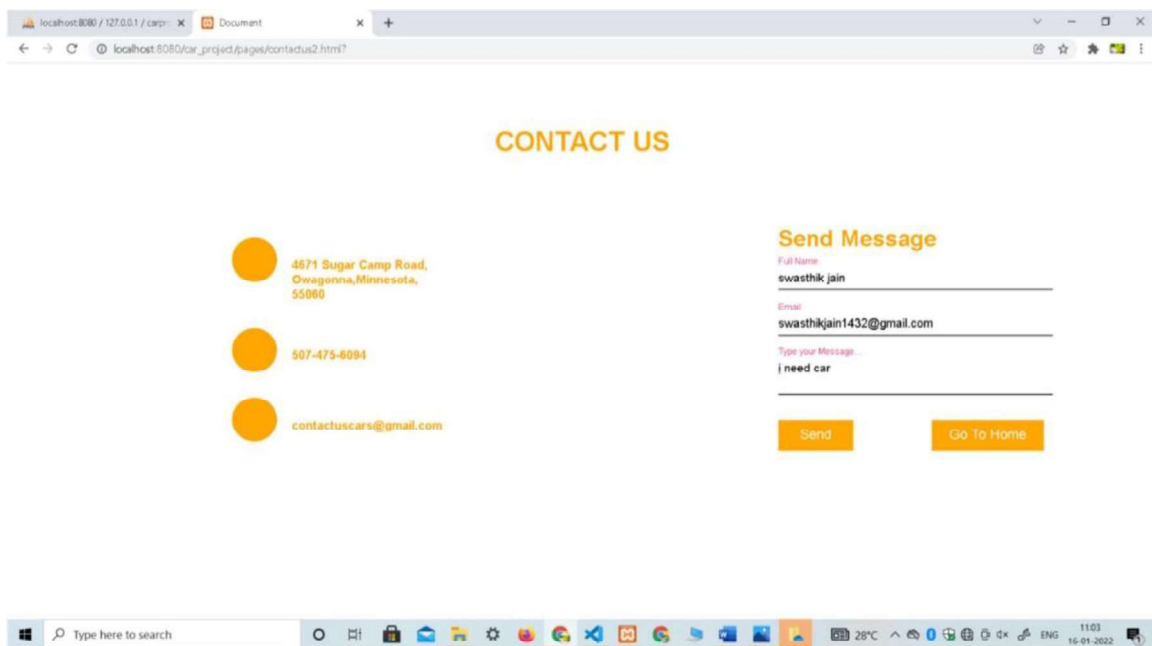


Figure 6.10 Screenshot of Contact Us page

Figure 6.10 indicates contact us page. It asks the user to enter details like full name, email, message where user can contact the company Cars by sending message.



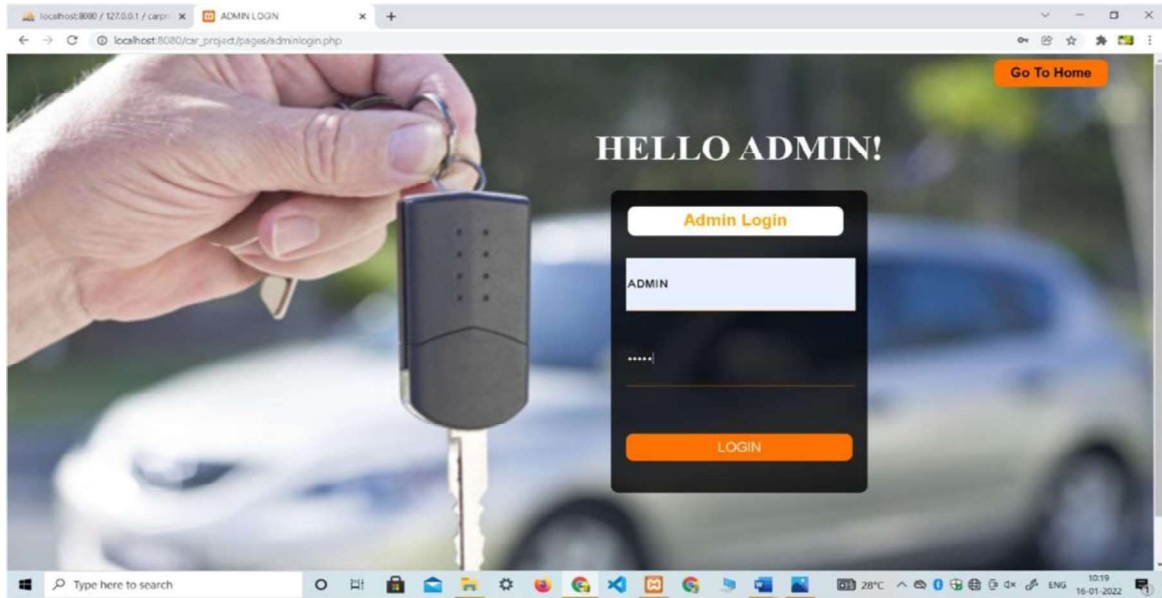


Figure 6.11 Screenshot of Adminn Login page

Figure 6.11 indicates admin login page. It asks the admin to enter details like admin\_id and password.

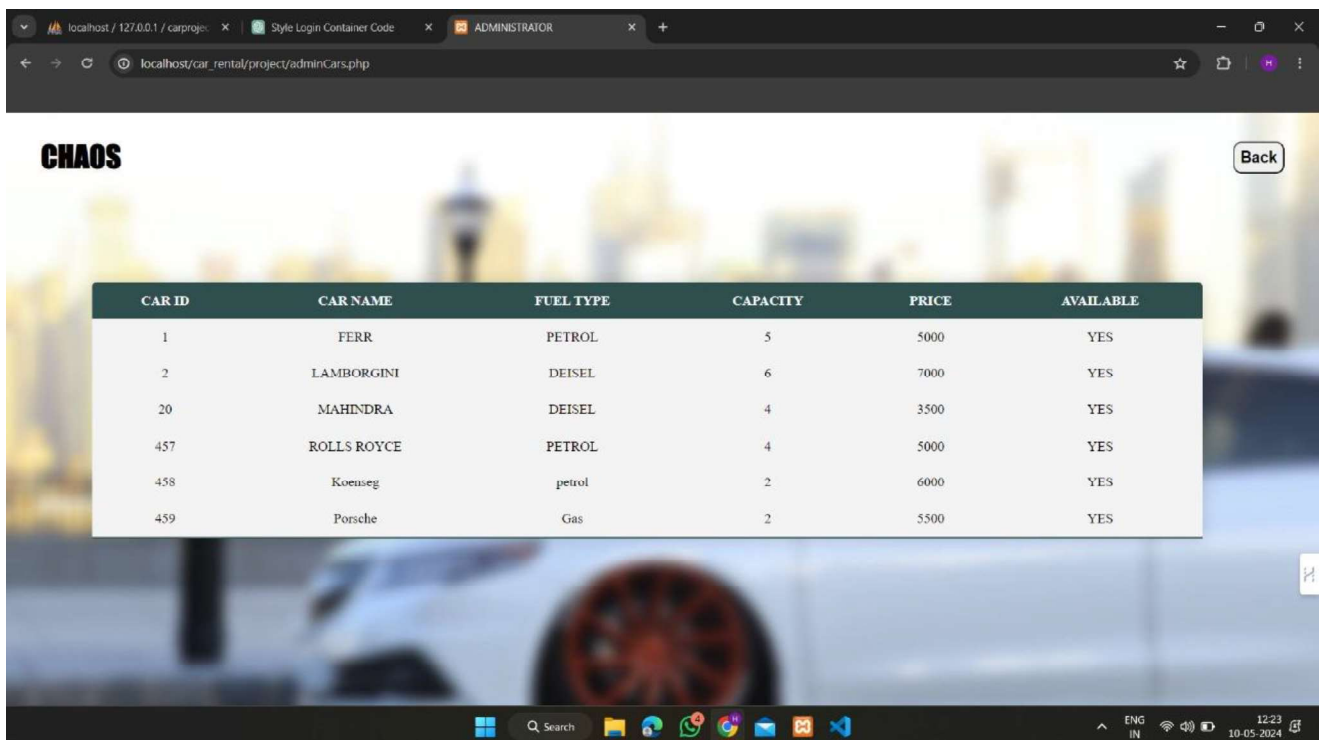


Figure 6.12 Screenshot of Vehicle Management page

Figure 6.12 indicates vehicle management page. It contains the details like car id, car name, fuel type, capacity, price, available, delete where admin can see the car availability status and delete the car details.

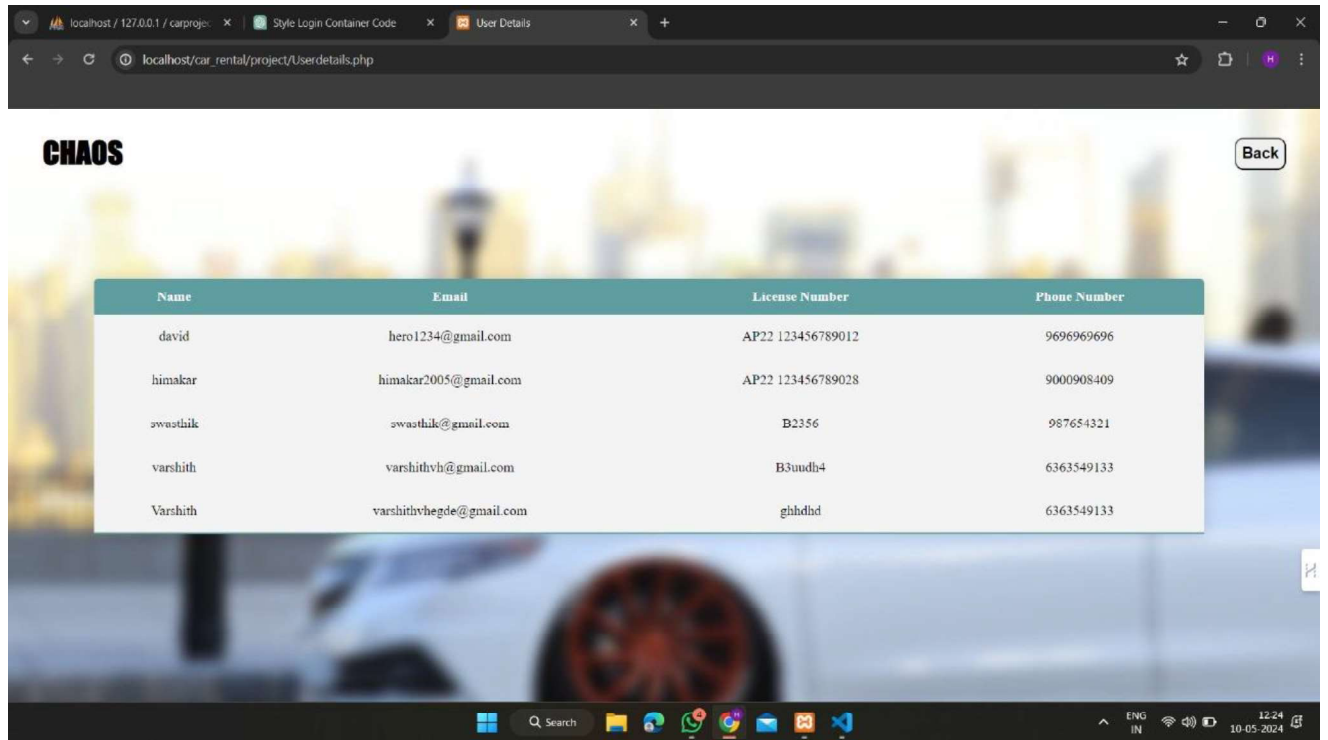


Figure 6.13 Screenshot of Users page

Figure 6.13 indicates User page. It includes details like name, email, license no, phone number, gender, delete user where admin can delete the user details.

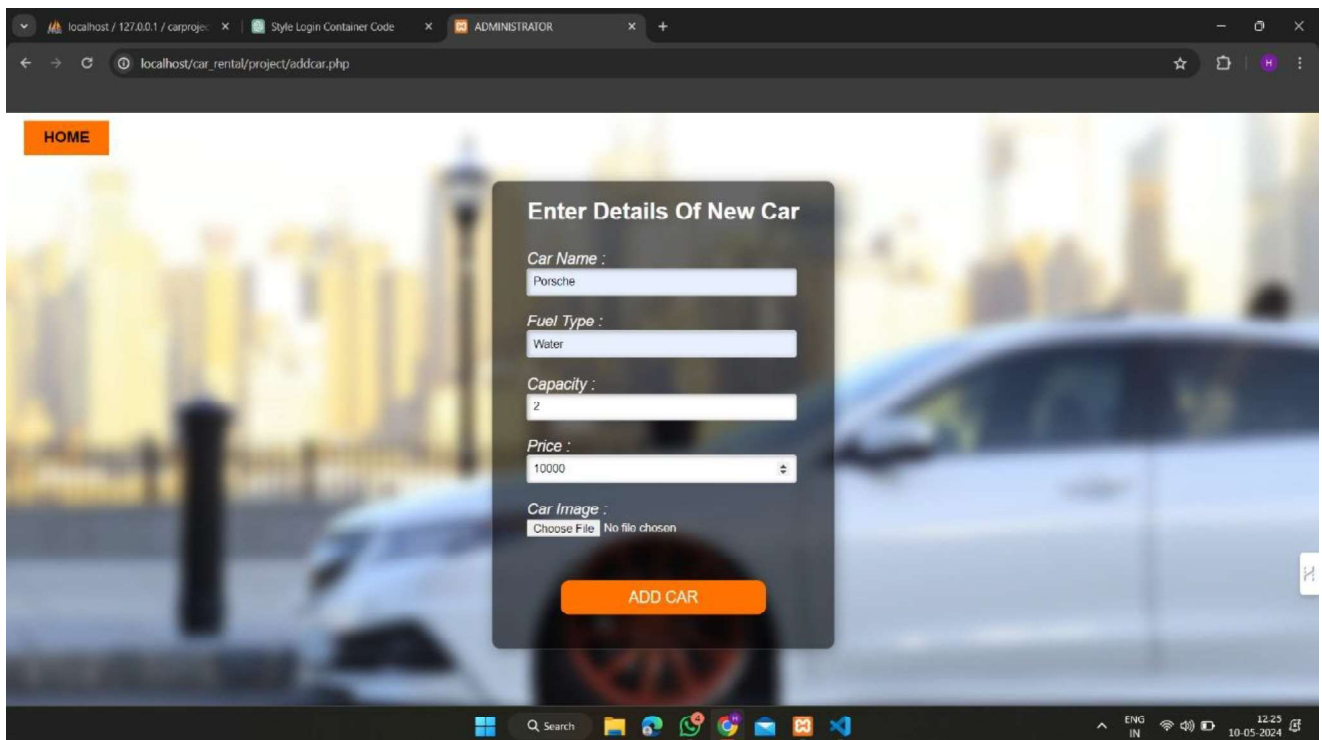
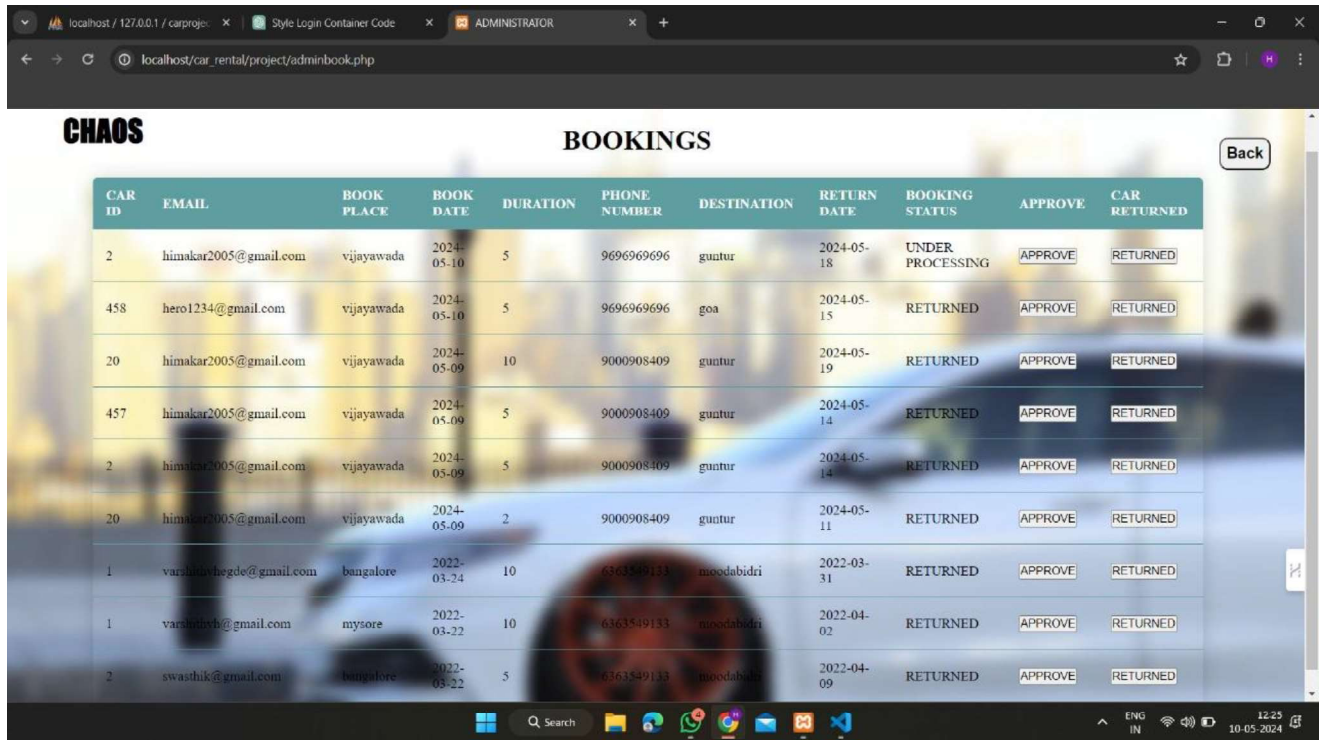


Figure 6.14 Screenshot of Add Car page

Figure 6.14 indicates add car page. It includes details like car name, fuel type, capacity, price, car image where admin can add the new car details by specifying it's details.



The screenshot shows a web application interface for a car rental system. The browser address bar indicates the URL is localhost/car\_rental/project/adminbook.php. The page title is "CHAOS BOOKINGS". A "Back" button is visible in the top right corner. The main content is a table listing bookings. The table has 11 columns: CAR ID, EMAIL, BOOK PLACE, BOOK DATE, DURATION, PHONE NUMBER, DESTINATION, RETURN DATE, BOOKING STATUS, APPROVE, and CAR RETURNED. The table contains 10 rows of booking data. The first row shows a booking with CAR ID 2, EMAIL himakar2005@gmail.com, BOOK PLACE vijayawada, BOOK DATE 2024-05-10, DURATION 5, PHONE NUMBER 9696969696, DESTINATION guntur, RETURN DATE 2024-05-18, BOOKING STATUS UNDER PROCESSING, and buttons for APPROVE and RETURNED. The second row shows a booking with CAR ID 458, EMAIL hero1234@gmail.com, BOOK PLACE vijayawada, BOOK DATE 2024-05-10, DURATION 5, PHONE NUMBER 9696969696, DESTINATION goa, RETURN DATE 2024-05-15, BOOKING STATUS RETURNED, and buttons for APPROVE and RETURNED. The third row shows a booking with CAR ID 20, EMAIL himakar2005@gmail.com, BOOK PLACE vijayawada, BOOK DATE 2024-05-09, DURATION 10, PHONE NUMBER 9000908409, DESTINATION guntur, RETURN DATE 2024-05-19, BOOKING STATUS RETURNED, and buttons for APPROVE and RETURNED. The fourth row shows a booking with CAR ID 457, EMAIL himakar2005@gmail.com, BOOK PLACE vijayawada, BOOK DATE 2024-05-09, DURATION 5, PHONE NUMBER 9000908409, DESTINATION guntur, RETURN DATE 2024-05-14, BOOKING STATUS RETURNED, and buttons for APPROVE and RETURNED. The fifth row shows a booking with CAR ID 2, EMAIL himakar2005@gmail.com, BOOK PLACE vijayawada, BOOK DATE 2024-05-09, DURATION 5, PHONE NUMBER 9000908409, DESTINATION guntur, RETURN DATE 2024-05-14, BOOKING STATUS RETURNED, and buttons for APPROVE and RETURNED. The sixth row shows a booking with CAR ID 20, EMAIL himakar2005@gmail.com, BOOK PLACE vijayawada, BOOK DATE 2024-05-09, DURATION 2, PHONE NUMBER 9000908409, DESTINATION guntur, RETURN DATE 2024-05-11, BOOKING STATUS RETURNED, and buttons for APPROVE and RETURNED. The seventh row shows a booking with CAR ID 1, EMAIL varshavshgde@gmail.com, BOOK PLACE bangalore, BOOK DATE 2022-03-24, DURATION 10, PHONE NUMBER 6363549133, DESTINATION moodabidri, RETURN DATE 2022-03-31, BOOKING STATUS RETURNED, and buttons for APPROVE and RETURNED. The eighth row shows a booking with CAR ID 1, EMAIL varshavshgde@gmail.com, BOOK PLACE mysore, BOOK DATE 2022-03-22, DURATION 10, PHONE NUMBER 6363549133, DESTINATION moodabidri, RETURN DATE 2022-04-02, BOOKING STATUS RETURNED, and buttons for APPROVE and RETURNED. The ninth row shows a booking with CAR ID 2, EMAIL swasthik@gmail.com, BOOK PLACE bangalore, BOOK DATE 2022-03-22, DURATION 5, PHONE NUMBER 6363549133, DESTINATION moodabidri, RETURN DATE 2022-04-09, BOOKING STATUS RETURNED, and buttons for APPROVE and RETURNED. The bottom of the screenshot shows a Windows taskbar with the search bar and various application icons. The system clock shows 12:25 on 10-05-2024.

| CAR ID | EMAIL                  | BOOK PLACE | BOOK DATE  | DURATION | PHONE NUMBER | DESTINATION | RETURN DATE | BOOKING STATUS   | APPROVE | CAR RETURNED |
|--------|------------------------|------------|------------|----------|--------------|-------------|-------------|------------------|---------|--------------|
| 2      | himakar2005@gmail.com  | vijayawada | 2024-05-10 | 5        | 9696969696   | guntur      | 2024-05-18  | UNDER PROCESSING | APPROVE | RETURNED     |
| 458    | hero1234@gmail.com     | vijayawada | 2024-05-10 | 5        | 9696969696   | goa         | 2024-05-15  | RETURNED         | APPROVE | RETURNED     |
| 20     | himakar2005@gmail.com  | vijayawada | 2024-05-09 | 10       | 9000908409   | guntur      | 2024-05-19  | RETURNED         | APPROVE | RETURNED     |
| 457    | himakar2005@gmail.com  | vijayawada | 2024-05-09 | 5        | 9000908409   | guntur      | 2024-05-14  | RETURNED         | APPROVE | RETURNED     |
| 2      | himakar2005@gmail.com  | vijayawada | 2024-05-09 | 5        | 9000908409   | guntur      | 2024-05-14  | RETURNED         | APPROVE | RETURNED     |
| 20     | himakar2005@gmail.com  | vijayawada | 2024-05-09 | 2        | 9000908409   | guntur      | 2024-05-11  | RETURNED         | APPROVE | RETURNED     |
| 1      | varshavshgde@gmail.com | bangalore  | 2022-03-24 | 10       | 6363549133   | moodabidri  | 2022-03-31  | RETURNED         | APPROVE | RETURNED     |
| 1      | varshavshgde@gmail.com | mysore     | 2022-03-22 | 10       | 6363549133   | moodabidri  | 2022-04-02  | RETURNED         | APPROVE | RETURNED     |
| 2      | swasthik@gmail.com     | bangalore  | 2022-03-22 | 5        | 6363549133   | moodabidri  | 2022-04-09  | RETURNED         | APPROVE | RETURNED     |

Figure 6.15 Screenshot of Bookings page

Figure 6.15 indicates booking page. It includes the details like car id, email, book place, book date, phone number, destination, return date, booking status, approve, car returned where admin can give approve to the bookings done by the user and can also check the returned status of the car.

## **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/>
- [5] <https://www.w3schools.com/php>
- [6] <https://www.tutorialspoint.com/javascript/index.htm/>