

Car Rental Website

*A project submitted in partial fulfilment of the requirements
for the award of the degree of*

Bachelor of Technology

In

COMPUTER SCIENCE AND ENGINEERING



Submitted by:

**Chandra Shekhar Verma
(12211084)**

&

**Yogesh Kumar
(12211103)**

Supervised By:

**Dr. Vinay Kaushik
(Assistant professor)**

**INDIAN INSTITUTE OF INFORMATION TECHNOLOGY,
SONEPAT – 131201, HARYANA, INDIA**

ACKNOWLEDGEMENTS

The success and the outcome of this project required ceaseless guidance and assistance, my team members and I are extremely privileged to have got this all along the project.

We would like to take this opportunity to acknowledge all the people who have helped us wholeheartedly in every stage of this project.

We are indebtedly grateful to Dr. VINAY KAUSHIK , Assistant professor, CSE, IIIT SONEPAT for providing this opportunity in the first place and giving us all the support and guidance possible, in spite of having a busy schedule.

ChandraShekhar
(12211084)

Yogesh Kumar
(12211103)

SELF DECLARATION

I hereby state that work contained in the project titled “Car Rental Website” is original. I have followed the standards of the project ethics to the best of my abilities. I have acknowledged all the sources of knowledge which I have used in the project.

Name: Chandra Shekhar Verma

Roll no: 12211084

Department of Computer

Science and Engineering,

Indian Institute of Information technology,

Sonepat-131201, Haryana, India

SELF DECLARATION

I hereby state that work contained in the project titled “Car Rental Website” is original. I have followed the standards of the project ethics to the best of my abilities. I have acknowledged all the sources of knowledge which I have used in the project.

Name: Yogesh Kumar

Roll no: 12211103

Department of Computer

Science and Engineering,

Indian Institute of Information technology,

Sonepat-131201, Haryana, India

CERTIFICATE

This is to certify that Mr. Chandra Shekhar has worked on the project entitled “Car Rental Website” under my supervision and guidance.

The contents of the project, being submitted to the Department of Computer Science and Engineering, IIIT SONEPAT, HARYANA, for the award of the degree of B. Tech in Computer Science and Engineering, are original and carried out by candidate himself. This project has not been submitted in full or part for award of any other degree or diploma to this or any other university.

Dr. Vinay Kaushik

(Supervisor)

Department of Computer Science and Engineering,

Indian Institute of Information technology,

Sonepat-131201, Haryana, India

CERTIFICATE

This is to certify that Mr. Yogesh Kumar has worked on the project entitled “Car Rental Website” under my supervision and guidance.

The contents of the project, being submitted to the Department of Computer Science and Engineering, IIIT SONEPAT, HARYANA, for the award of the degree of B. Tech in Computer Science and Engineering, are original and carried out by candidate himself. This project has not been submitted in full or part for award of any other degree or diploma to this or any other university.

Dr. Vinay Kaushik

(Supervisor)

Department of Computer Science and Engineering,

Indian Institute of Information technology,

Sonepat-131201, Haryana, India

ABSTRACT

Name of the student: Chandra Shekhar Verma

Roll No: 12211084

Degree for which submitted: BTech

Department of Computer Science And Engineering

Indian Institute of Information Technology, Sonepat

Name of the student: Yogesh Kumar

Roll No: 12211103

Degree for which submitted: BTech

Department of Computer Science And Engineering

Indian Institute of Information Technology, Sonepat

Project Title: Car rental Website

Name of the project supervisor: **Dr. Vinay Kaushik**

Month and year of the project submission: **May, 2024**

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 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.

LIST OF ABBREVIATIONS

S/W	Software
Js	JavaScript
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheets
AUT	Application Under Test
JSON	JavaScript Object Notation
PUW	Project Under Work
SQL	Structure Query Language
DBMS	Database Management System
FIG	Figure
Con	Connection

LIST OF FIGURES

Figure No.	Name Of Figure	Page No.
2.1	API Working Diagram	8
3.1	ER Diagram of Car Rental System	11
3.2	Schema Diagram of Car Rental System	12
3.3	Block Diagram of Car Rental System	14
3.4	Flow Chart of Car Rental System	15
3.5	Code Snippet of Connection module	16
3.6	Code Snippet of Delete Car Module	16
3.7	Code Snippet of Uploading Car Image Module	17
3.8	Code Snippet of Approve Module	18
3.9	Code Snippet for Home page.	21
3.10	Source code for admin login page	22
3.11	Code Snippet Registration Page	24
4.1	Home Page	29
4.2	Registration Page	29
4.3	User Login Page	30
4.4	View Car Page 1	30
4.5	View Car Page 1	31
4.6	Payment Page	31
4.7	Payment Successful Page	32
4.8	Booking Status Page	32
4.9	Feedback Page	33
4.10	Contact Us Page	33
4.11	Admin Login Page	34
4.12	Vehicle Management Page	34
4.13	Users Page	35
4.14	Add Car Page	35
4.15	Booking Page	36

LIST OF TABLES

Table Number	Name Of Table	Page Number
4.1	Test Case for login	27
4.2	Test Case for Signup	27
4.3	Test Case for Payment	27
4.4	Test Case for Booking	28
4.5	Test Case for Feedback	28

Acknowledgments	i
Self-Declaration	ii-iii
Certificate	iv-v
Abstract	vi
List of Abbreviations	vii
List of figures	viii
List of Tables	ix

TABLE OF CONTENTS

Chapter 1	Introduction		1-4
	1.1	Introduction	2
	1.2	Problem Outline	2
	1.3	Project Objectives	2
	1.4	Project Methodology	3
	1.5	Scope Of Project Work	3
	1.6	Limitations	3
	1.7	Summary	4
Chapter 2	Study and Review of Literature		5-9
	2.1	Introduction	6
	2.2	Front End Development	6
	2.3	Back End Development	7
	2.4	Deployment	9
	2.5	Summary	9
Chapter 3	Implementation		10-24
	3.1	Introduction	11
	3.2	System design	11
	3.3	ER Diagram	11
	3.4	Schema Diagram	12
	3.5	Block Diagram	14

	3.6	Flow Chart	14
	3.7	Code Snippets	16
Chapter 4	RESULT AND CONCLUSION		25-37
	4.1	Testing	26
	4.2	Testing Process	26
	4.3	Test Cases	27
	4.4	Results and Screenshots	29
	4.5	Conclusion	36
	References		37

Chapter 1

Introduction

1.1 INTRODUCTION

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 PROBLEM OUTLINE

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.

1.3 PROJECT OBJECTIVES

- 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 PROJECT METHODOLOGY

The methodology for designing a car rental website with HTML, CSS, JavaScript, and PHP involves a structured approach beginning with requirements analysis, followed by research and planning to establish project milestones and timelines. This is succeeded by UI design, where wireframes and designs are crafted for user-friendly interfaces, leading into frontend development where HTML, CSS, and JavaScript are implemented to ensure responsiveness and compatibility. Concurrently, backend development involves setting up servers, databases, and PHP scripts, integrating user authentication and payment gateways. The process continues with integration and testing to combine frontend and backend elements, followed by deployment on a hosting server with domain configuration and SSL setup. Post-launch, maintenance and optimization entail regular updates, analytics analysis, and user feedback iteration for continuous enhancement of the website's performance and user experience.

1.5 SCOPE OF PROJECT WORK

The scope of a car rental website encompasses various components to facilitate efficient vehicle rental services and enhance user experience. It includes features such as user registration and authentication for personalized accounts, a comprehensive vehicle inventory showcasing various models, makes, and rental options, a robust search and filtering system for easy browsing, a booking and reservation system with flexible rental durations and payment options, integrated maps for locating rental outlets or vehicle delivery points, secure payment gateways to facilitate transactions, a customer review and rating system for feedback and reputation management, administrative tools for managing inventory, bookings, and user accounts, as well as responsive design and optimization for seamless access across devices. Additionally, features like insurance options, roadside assistance, loyalty programs, and integration with third-party services like navigation apps or car-sharing platforms can further enhance the website's functionality and value proposition.

1.6 LIMITATIONS: -

Designing a car rental website presents several challenges that can impact its functionality and user experience. These include ensuring data accuracy from third-party sources, addressing security concerns related to handling sensitive customer data, and navigating legal requirements and insurance regulations across different jurisdictions. Additionally, effective customer support and efficient inventory management are essential for maintaining user satisfaction and managing a diverse fleet of rental vehicles. Furthermore, competition in the industry, technological constraints, and considerations for user accessibility add further complexity to the design process, necessitating careful planning and strategic decision-making to overcome these limitations and deliver a reliable and user-friendly car rental platform.

1.7 SUMMARY

The project aims to develop a web-based car rental system to streamline the rental process for customers and improve management efficiency for the rental company. The methodology involves a structured approach, starting with requirements analysis and planning, followed by UI design and frontend/backend development using HTML, CSS, JavaScript, and PHP. The system will offer features like user registration, vehicle inventory, booking/reservation, secure payment, and administrative tools. The scope encompasses various components to enhance user experience, including responsive design, integrated maps, and additional features like insurance options and loyalty programs. Despite challenges such as data accuracy, security concerns, and legal compliance, careful planning and strategic decision-making will ensure the successful implementation of the car rental website, delivering a reliable and user-friendly platform to meet customer needs.

Chapter 2

Study and Review of Literature

2.1 INTRODUCTION

The rise of car rental websites marks a transformative shift in transportation, offering users convenient access to a diverse range of vehicles and services. This study delves into the functionalities and impacts of these platforms through a literature review. By synthesizing existing research, we aim to uncover the driving forces behind their success, challenges they face, and their broader implications for the automotive industry and society at large. Through this exploration, we seek to contribute to a deeper understanding of the evolving landscape of mobility and technology.

An e-commerce website, by definition, is a website that allows the user to buy and sell goods, digital products or services online. Trade, whether being exchange or buying and selling of goods and services has been prevalent for centuries. No one can be self-sufficient. And this brings out the need for demand and supply of goods and services. Transactions have been going on all over the world for centuries, locally, and across locations. And before user begin, user should be well aware of the tools and technologies required to build such a system or a web application. This has evolved more with the emergence of smartphones, where now, user can shop from anywhere and anytime, with a wireless device connected to the Internet. Now user can search for almost any product or service online, without having to go anywhere physically. In 2020, eCommerce sales are expected to account for 15.5 percent of retail sales worldwide.

2.2 FRONT-END DEVELOPMENT

Front-end web development, also known as client-side development is the practice of producing HTML, CSS and JavaScript for a website or Web Application so that a user can see and interact with them directly. The front end of a website is the part that users interact with. Everything that user see when user navigating around the Internet, from fonts and colors to dropdown menus and sliders, is a combo of HTML, CSS, and JavaScript being controlled by user's computer's browser. The challenge associated with front end development is that the tools and techniques used to create the front end of a website change constantly and so the developer needs to constantly be aware of how the field is developing.

HTML

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)

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.

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.

2.3 BACK-END DEVELOPMENT

Backend website development consists of those tasks that allow user to optimize the website. The database, content, plugins, and other elements that make the website operate are all part of its backend. The backend ensures that the website is fully functional and applies to business websites, blogs, ecommerce sites, and others. The needs of the backend development will depend on the nature of the website and the functionality it offers to its users. This is distinct from the website's frontend, which consists of the coding languages used to create the site's pages, navigation, and other parts that result in the visual display with which users interact when engaging content. The backend and frontend of the website play different roles, but they work together to create the sum total experience of the website. Backend development utilizes the applications that access data and deliver it to the frontend for the audiences. The services and tools which were utilized to prepare the back end of the web application have been briefly mentioned below.

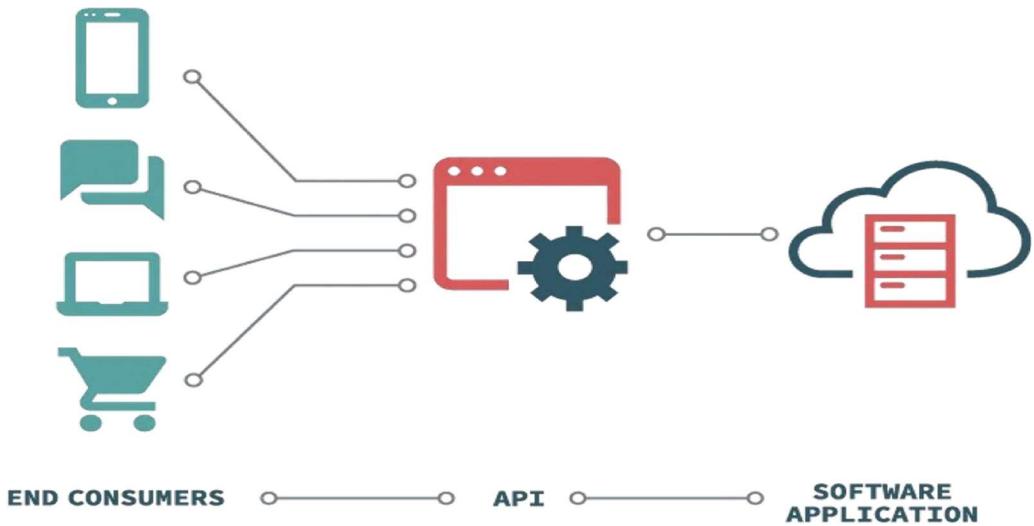


Fig 2.1 API Working Diagram

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.

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).

2.4 DEPLOYMENT

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.

2.5 SUMMARY

The study explores the transformative impact of car rental websites on transportation, highlighting their convenience and accessibility. Through a literature review, the study aims to uncover the functionalities, challenges, and broader implications of these platforms, contributing to a deeper understanding of mobility and technology's evolving landscape. Transitioning to e-commerce, the discussion defines e-commerce websites and their significance in facilitating online transactions for goods and services. The frontend development section delves into HTML, CSS, and JavaScript's role in creating user interfaces, emphasizing constant adaptation to evolving tools and techniques. Backend development, covered next, focuses on optimizing website functionality through PHP and SQL, managing databases and server-side scripting. Deployment involves using the Apache web server to parse and execute PHP pages, ensuring seamless functionality. Overall, the study provides insights into the technical and operational aspects of developing car rental and e-commerce websites, acknowledging the dynamic nature of web development technologies.

Chapter 3

Implementation

3.1 INTRODUCTION

This chapter will describe the key implementation processes, the UML diagrams and some code snippets of car rental website. Implementation in software development is the process of realizing an application's requirements and design. It mainly involves mapping the design into coding in order to achieve the specifications stated for the application. The following Sections define the different modules of the implementation.

3.2 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.

3.3 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.

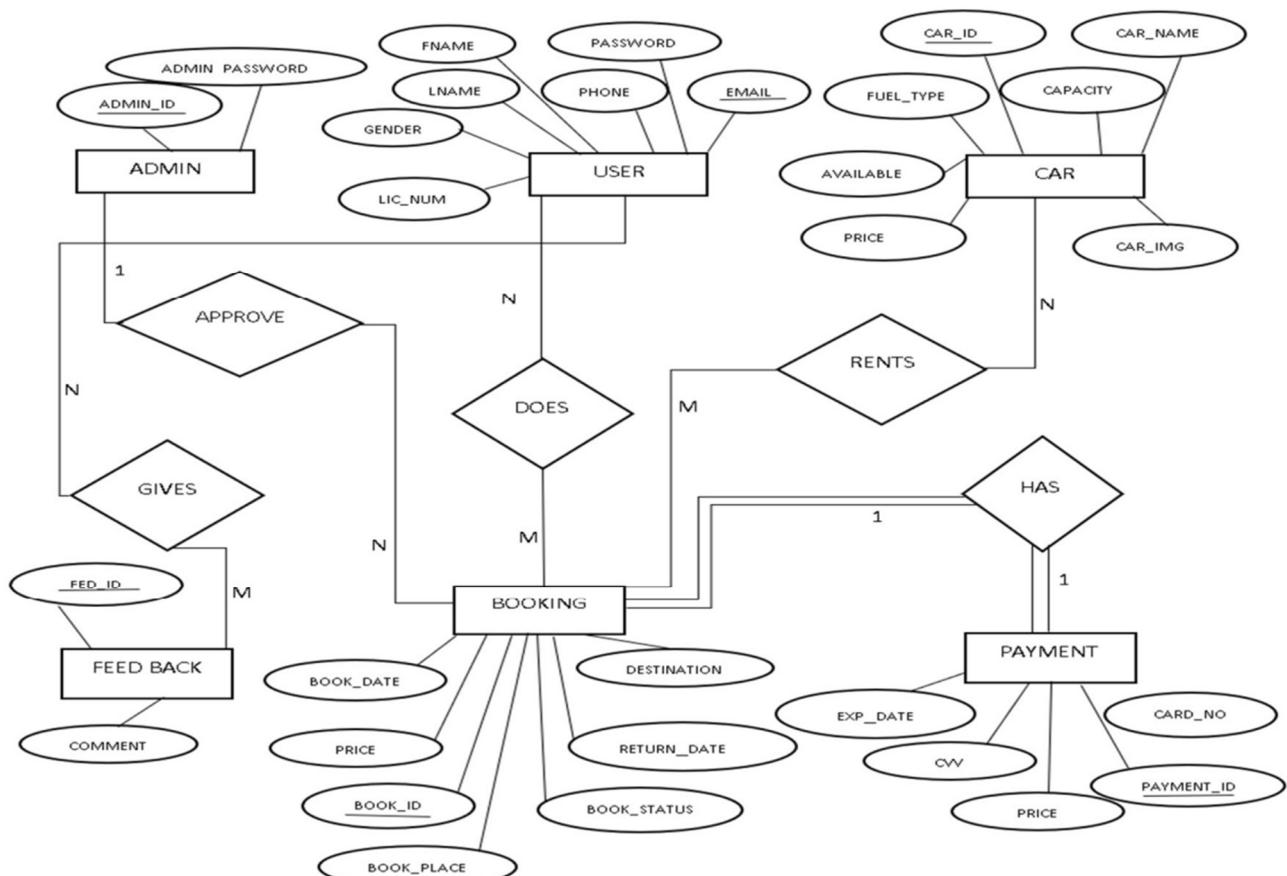


Fig 3.1 ER Diagram of Car Rental System

In the Figure, 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.

3.4 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.

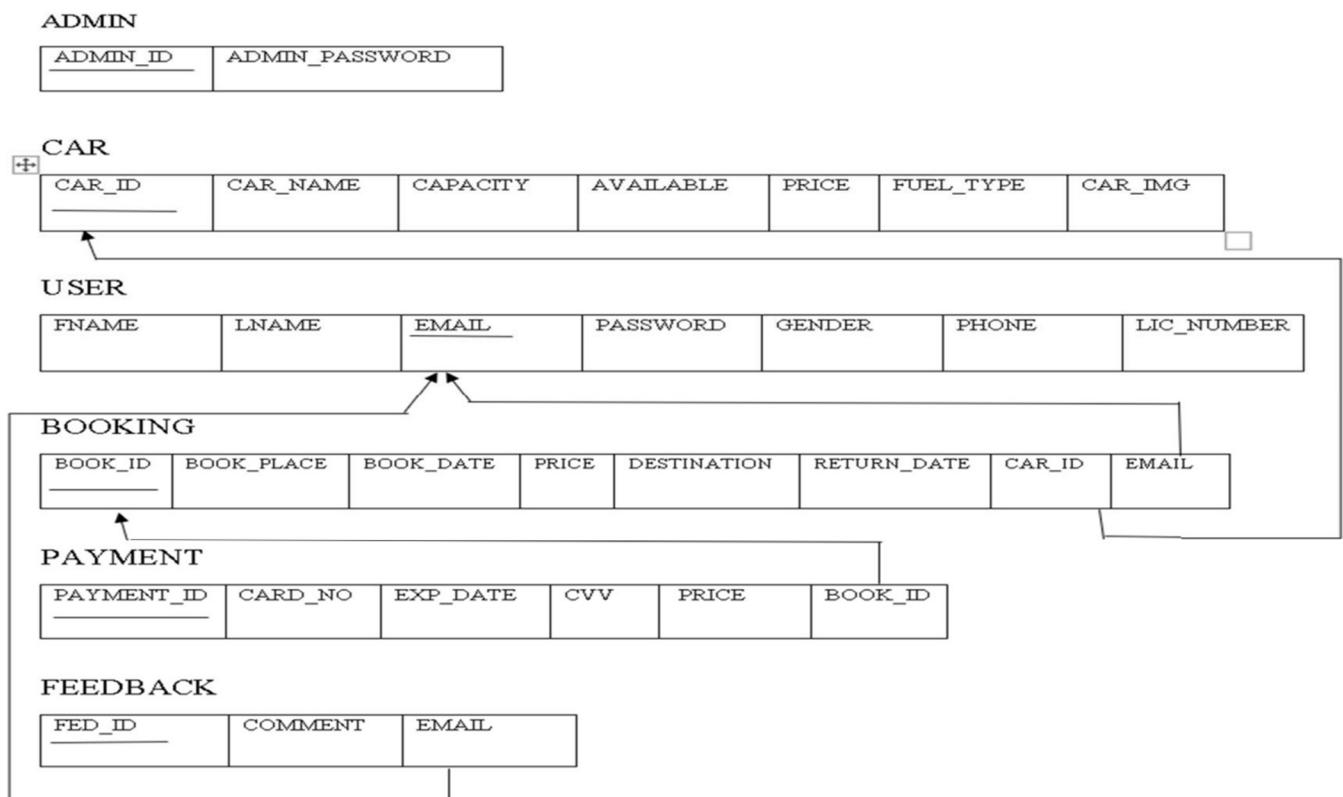


Fig 3.2 Schema Diagram of Car Rental System

3.4.1 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.

3.5 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.

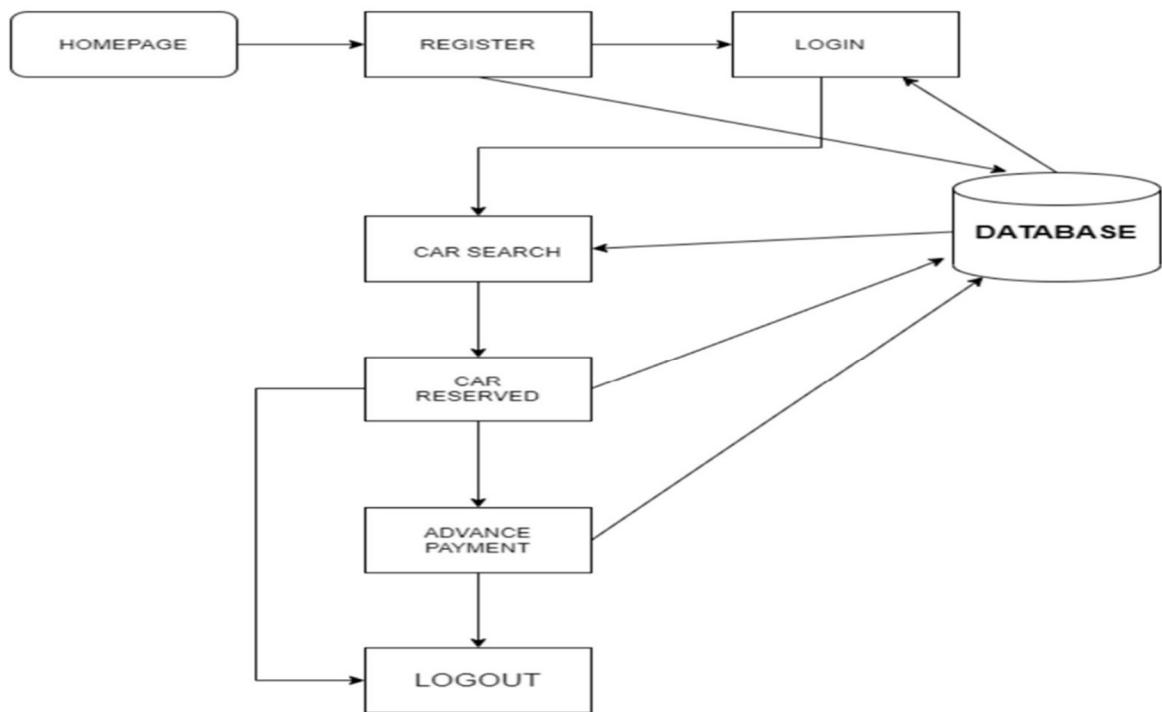


Fig 3.3 Block Diagram of Car Rental System

In the Figure , 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.

3.6 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.

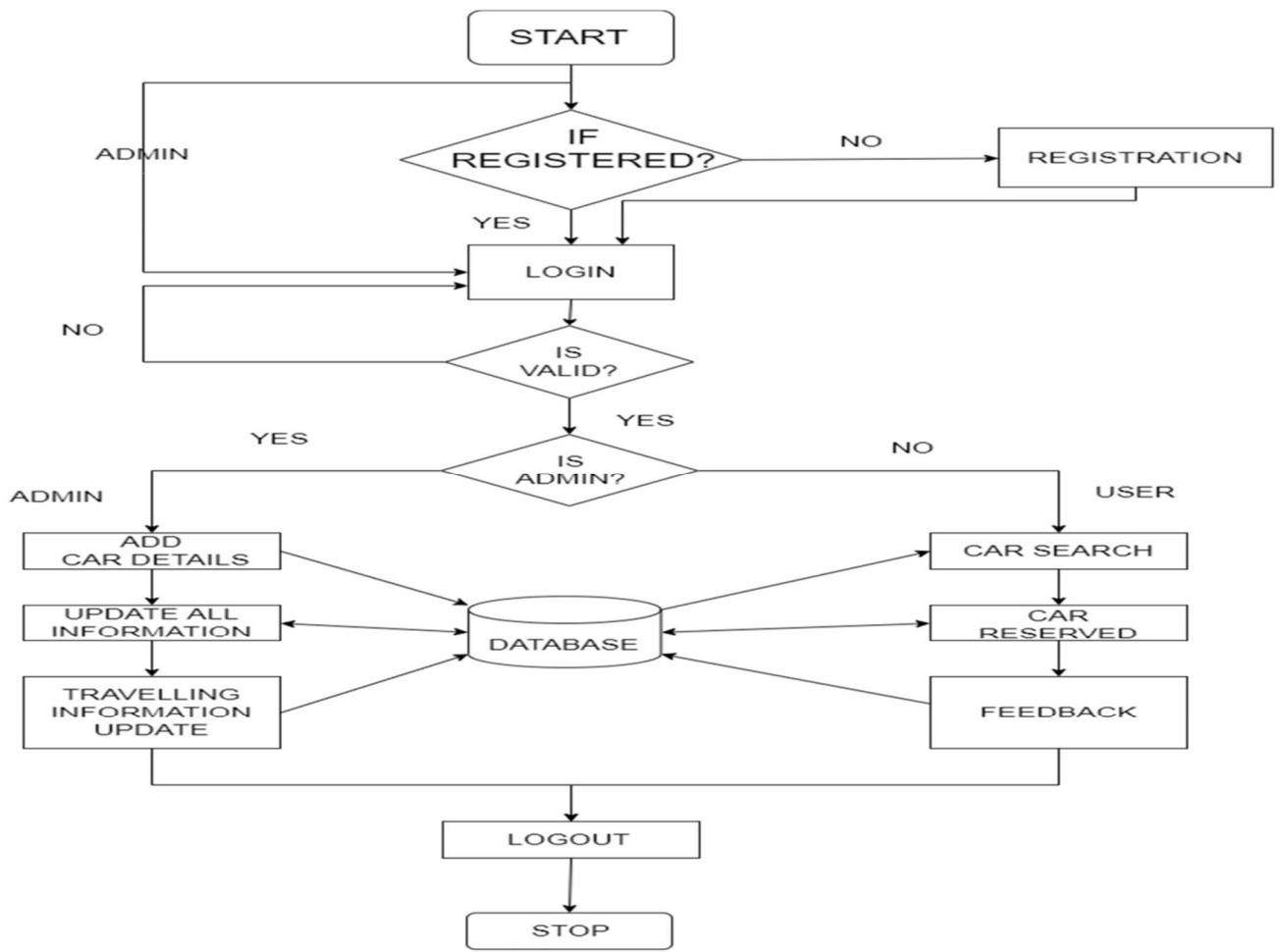


Fig 3.4 Flow Chart of Car Rental System

In the Figure 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.

3.7 CODE SNIPPETS

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.

```
1
2 <?php
3 // Database credentials
4 $servername = "localhost:3307";
5 $username = "root";
6 $password = "";
7 $database = "carproject";
8
9 // Create connection
10 $con = new mysqli($servername, $username, $password, $database);
11
12 // Check connection
13 if ($con->connect_error) {
14     die("Connection failed: " . $con->connect_error);
15 }
16 ?>
```

Fig 3.5 Code Snippet of Connection module

Figure 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.

```
1 <?php
2
3 require_once('connection.php');
4 $carid=$_GET['id'];
5 $sql="DELETE from cars where CAR_ID=$carid";
6 $result=mysqli_query($con,$sql);
7
8 echo '<script>alert("CAR DELETED SUCCESFULLY")</script>';
9 echo '<script> window.location.href = "adminvehicle.php";</script>';
10
11
12
13 ?>
```

Fig 3.6 Code Snippet of Delete Car Module

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

```

<?php
if(isset($_POST['addcar'])){
    require_once('connection.php');
    echo "<prev>";
    print_r($_FILES['image']);
    echo "</prev>";
    $img_name= $_FILES['image']['name'];
    $tmp_name= $_FILES['image']['tmp_name'];
    $error= $_FILES['image']['error'];
    if($error === 0){
        $img_ex = pathinfo($img_name,PATHINFO_EXTENSION);
        $img_ex_lc= strtolower($img_ex);

        $allowed_exs = array("jpg","jpeg","png","webp","svg");
        if(in_array($img_ex_lc,$allowed_exs)){
            $new_img_name=uniqid("IMG-",true).'.'.$img_ex_lc;
            $img_upload_path='images/'.$new_img_name;
            move_uploaded_file($tmp_name,$img_upload_path);

            $carname=mysqli_real_escape_string($con,$_POST['carname']);

            $ftype=mysqli_real_escape_string($con,$_POST['ftype']);
            $capacity=mysqli_real_escape_string($con,$_POST['capacity']);
            $price=mysqli_real_escape_string($con,$_POST['price']);
            $available="Y";
            $query="INSERT INTO cars(CAR_NAME,FUEL_TYPE,CAPACITY,PRICE,CAR_IMG,
            AVAILABLE) values('$carname','$ftype',$capacity,$price,'$new_img_name',
            '$available')";
            $res=mysqli_query($con,$query);
            if($res){
                echo '<script>alert("New Car Added Successfully!!")</script>';
                echo '<script> window.location.href = "adminvehicle.php";</
                script>';}
        }
    }else{
        $em="unknown error occurred";
        header("Location: addcar.php?error=$em");
    }
}

```

Fig 3.7 Code Snippet of Uploading Car Image Module

Figure 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);
$email=$res['EMAIL'];
$carname=$carresult['CAR_NAME'];
if($carresult['AVAILABLE']=='Y')
{
if($res['BOOK_STATUS']=='APPROVED' || $res['BOOK_STATUS']
=='RETURNED')
{
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>';
}
?>
```

Fig 3.8 Code Snippet of Approve Module

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

3.7.1 Home Page

Home Page of the Website gives the user option login , sign up and visit the cars available for rent. Given figures shows the source code for home page.

```
<!DOCTYPE html>
<html lang="en">
<head>

    <title>CAR RENTAL</title>
    <script type="text/javascript">
        window.history.forward();
        function noBack() {
            window.history.forward();
        }
    </script>
    <link rel="stylesheet" href="style.css">
    <script type="text/javascript">
        function preventBack() {
            window.history.forward();
        }
    </script>
    setTimeout("preventBack()", 0);

    window.onunload = function () { null };
</script>
</head>
<body>

<?php
require_once('connection.php');
if(isset($_POST['login']))
{
    $email=$_POST['email'];
    $pass=$_POST['pass'];

    if(empty($email)|| empty($pass))
    {
        echo '<script>alert("please fill the blanks")</
        script>';
    }

    else{
        $query="select *from users where EMAIL='$email'";
        $res=mysqli_query($con,$query);
        if($row=mysqli_fetch_assoc($res)){
            $db_password = $row['PASSWORD'];
        }
    }
}


```

```

        if(md5($pass) == $db_password)
        {
            header("location: cardetails.php");
            session_start();
            $_SESSION['email'] = $email;

        }
        else{
            echo '<script>alert("Enter a proper password")</script>';
        }

    }
    else{
        echo '<script>alert("enter a proper email")</script>';
    }
}

?>
<div class="hai">
    <div class="navbar">
        <div class="icon">
            <h2 class="logo">CaRs</h2>
        </div>
        <div class="menu">
            <ul>
                <li><a href="#">HOME</a></li>
                <li><a href="aboutus.html">ABOUT</a></li>
                <li><a href="#">SERVICES</a></li>

                <li><a href="contactus.html">CONTACT</a></li>
                <li> <button class="adminbtn"><a href="adminlogin.php">ADMIN LOGIN</a></button></li>
            </ul>
        </div>
    </div>

```

```

</div>
<div class="content">
    <h1>Rent Your <br><span>Dream Car</span></h1>
    <p class="par">Live the life of Luxury.<br>
        Just rent a car of your wish from our vast
        collection.<br>Enjoy every moment with your
        family<br>
        Join us to make this family vast. </p>
    <button class="cn"><a href="register.php">JOIN US</
    a></button>
    <div class="form">
        <h2>Login Here</h2>
        <form method="POST">
            <input type="email" name="email"
            placeholder="Enter Email Here">
            <input type="password" name="pass"
            placeholder="Enter Password Here">
            <input class="bttn" type="submit" value="Login"
            name="login"></input>
        </form>
        <p class="link">Don't have an account?<br>
        <a href="register.php">Sign up</a> here</a></p>
    </div>
</div>
</div>
<script src="https://unpkg.com/ionicons@5.4.0/dist/ionicons.
js"></script>
</body>
</html>

```

Fig 3.9 Code Snippet for Home page.

3.7.2 Admin Login Page

Admin Login Page of the Website gives the admin option to login in website and from admin page , he can manage the administrative part of website.

From admin page we can add new models , manage the request of users, revies and suggetions.

```
<?php
    require_once('connection.php');
    if(isset($_POST['adlog'])){
        $id=$_POST['adid'];
        $pass=$_POST['adpass'];

        if(empty($id)|| empty($pass))
        {
            echo '<script>alert("please fill the blanks")</
            script>';
        }

        else{
            $query="select *from admin where ADMIN_ID='$id'";
            $res=mysqli_query($con,$query);
            if($row=mysqli_fetch_assoc($res)){
                $db_password = $row['ADMIN_PASSWORD'];
                if($pass == $db_password)
                {

                    // session_start();
                    // $_SESSION['email'] = $email;
                    echo '<script>alert("Welcome
ADMINISTRATOR!");</script>';
                    header("location: admindash.php");

                }
                else{
                    echo '<script>alert("Enter a proper
password")</script>';
                }
            }
            else{
                echo '<script>alert("enter a proper email")</
script>';
            }
        }
    }
}
```

Fig 3.10 Source code for admin login page

3.7.3 User Sign Up Page

From here new user can join our website and take the benefits of the facilities ,

```
<?php

require_once('connection.php');
if(isset($_POST['regs']))
{
    $fname=mysqli_real_escape_string($con,$_POST['fname']);
    $lname=mysqli_real_escape_string($con,$_POST['lname']);
    $email=mysqli_real_escape_string($con,$_POST['email']);
    $lic=mysqli_real_escape_string($con,$_POST['lic']);
    $ph=mysqli_real_escape_string($con,$_POST['ph']);

    $pass=mysqli_real_escape_string($con,$_POST['pass']);
    $cpass=mysqli_real_escape_string($con,$_POST['cpass']);
    $gender=mysqli_real_escape_string($con,$_POST['gender']);
    $Pass=md5($pass);
    if(empty($fname)|| empty($lname)|| empty($email)|| empty($lic)|| empty($ph)|| empty($pass) || empty($gender))
    {
        echo '<script>alert("please fill the place")</script>';
    }
    else{
        if($pass==$cpass){
            $sql2="SELECT *from users where EMAIL='".$email."'";
            $res=mysqli_query($con,$sql2);
            if(mysqli_num_rows($res)>0){
                echo '<script>alert("EMAIL ALREADY EXISTS PRESS OK FOR LOGIN!!")</script>';
                echo '<script> window.location.href = "index.php";</script>';
            }
            else{
                $sql="insert into users (FNAME,LNAME,EMAIL,LIC_NUM,PHONE_NUMBER,PASSWORD,GENDER) values('$fname','$lname','$email','$lic',$ph,'$Pass','$gender')";
                $result = mysqli_query($con,$sql);

                if($result){
                    echo '<script>alert("Registration Successful Press ok to login")</script>';
                    echo '<script> window.location.href = "index.php";</script>';
                }
            }
        }
    }
}
```

```
    }
} else{
    echo '<script>alert("please check the connection")</
    script>';
}

}

else{
    echo '<script>alert("PASSWORD DID NOT MATCH")</
    script>';
    echo '<script> window.location.href = "register.php";
    </script>';
}
}

?>
```

Fig 3.11 Code Snippet Registration Page

3.8 SUMMARY

This chapter was about showing and Explaining the Use Case diagrams for the administrator as well as the general user, the Class Diagrams to show the interaction between different parts of system, also the admin login sequence diagram. Code Snippets have been provided for further showing the implementation of various functionalities.

CHAPTER 4

RESULT AND CONCLUSION

4.1 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.

4.2 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.

4.2.1 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.

4.2.2 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.

4.2.3 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.

4.3 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 4.1 Test Case for login

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

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 4.2 Test Case for Signup

Table 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 4.3 Test Case for Payment

Table 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 4.4 Test Case for Booking

Table 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 4.5 Test Case for Feedback

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

4.4 RESULT AND SCREENSHOTS

After successfully compiling the Final Code and Deploying the Car rental website the final Results achieved are shown below.

4.4.1 Home Page

The figure 4.1 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.

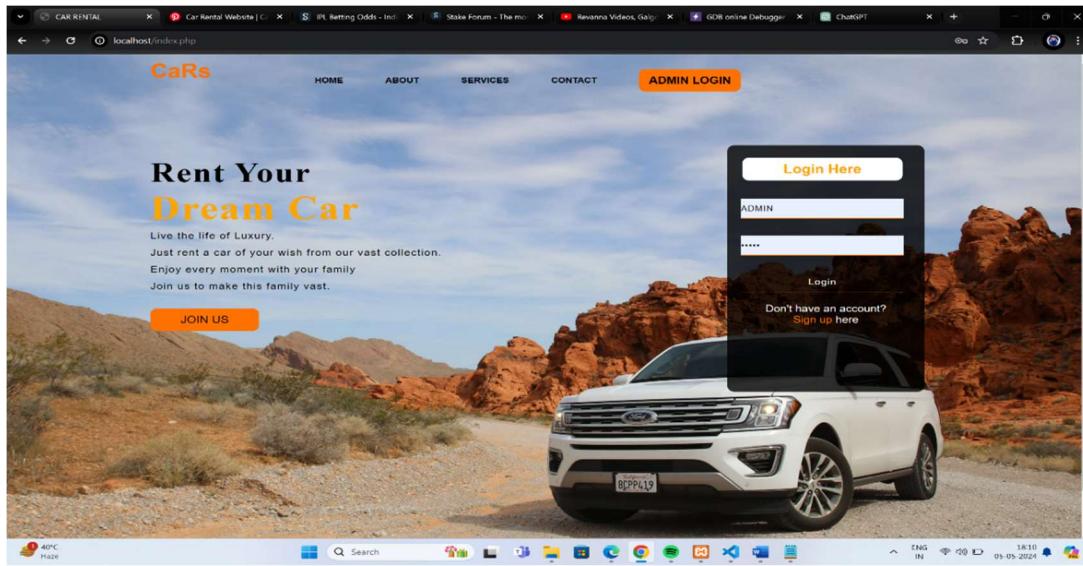


Fig 4.1 Home Page

4.4.2 Registration Page

Figure 4.2 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.

A screenshot of a web browser showing the registration page of the website. The page has a dark background with a car on fire. The title 'JOIN OUR FAMILY OF CARS!' is at the top. There is a 'HOME' button on the left. A registration form titled 'Register Here' is in the center. The form fields are: First Name (with placeholder 'Enter Your First Name'), Last Name (with placeholder 'Enter Your Last Name'), Email (with placeholder 'ADMIN'), Your License number (with placeholder 'Enter Your License number'), Phone Number (with placeholder 'Enter Your Phone Number'), Password (with placeholder '.....'), and Confirm Password (with placeholder 'Reenter the password').

Fig 4.2 Registration Page

4.4.3 User Login Page

Figure 4.3 indicates user login page. It asks the user to enter the email and password in order to enter the website CaRs

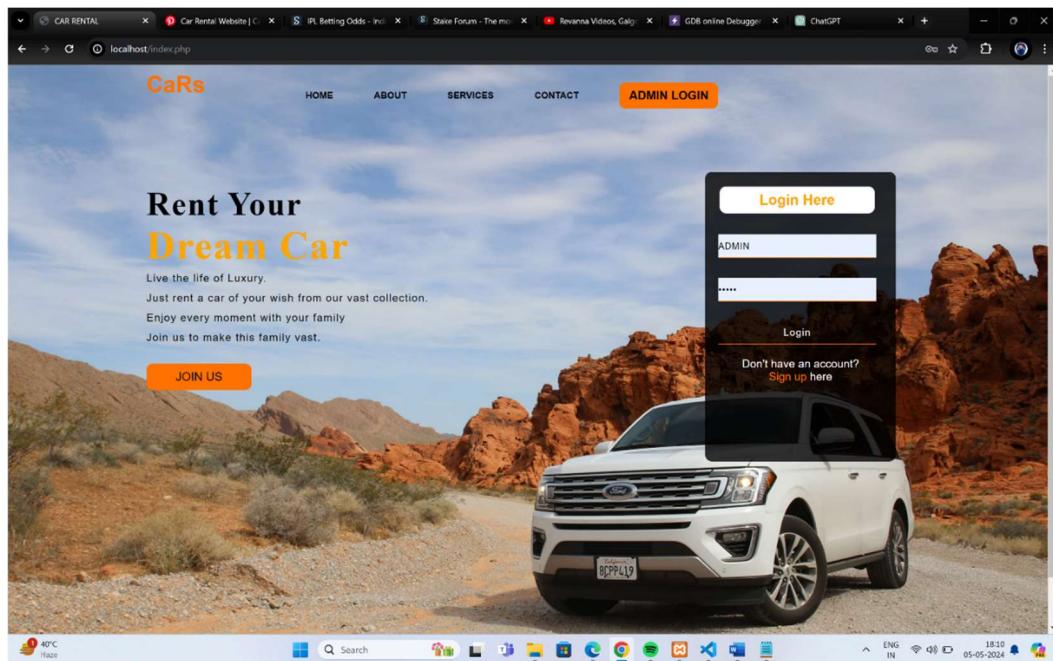


Fig 4.3 User Login Page

4.4.4 View Car Page

Figure 4.4 and 4.5 indicates view car page. It asks the user to choose the car it contains the details like fuel type, capacity, rent per day.

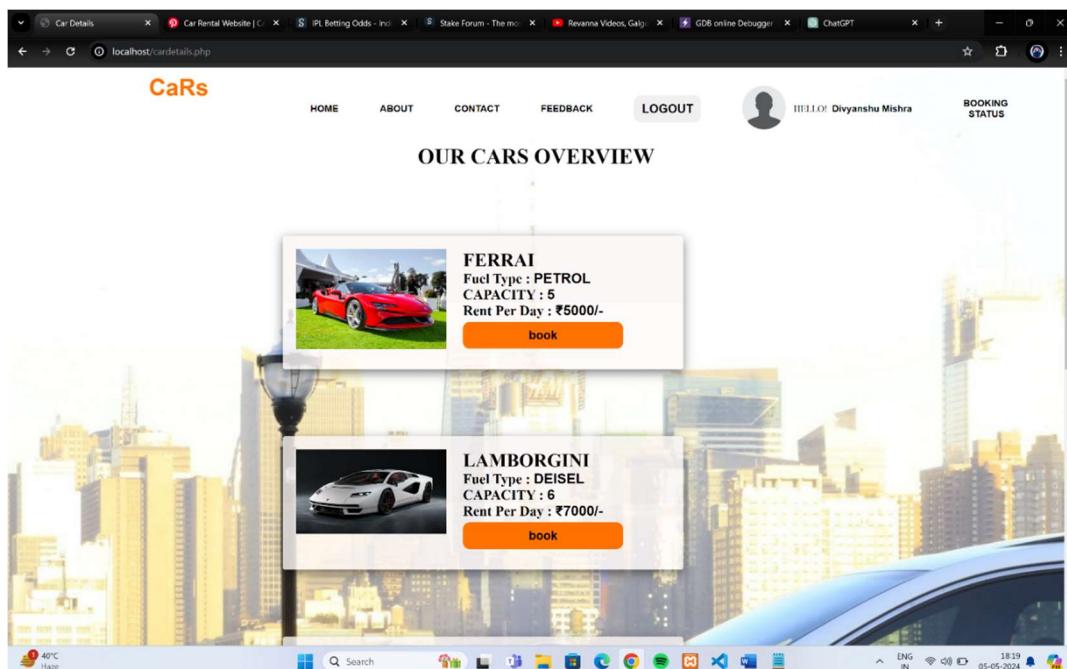


Fig 4.4 View Car Page1

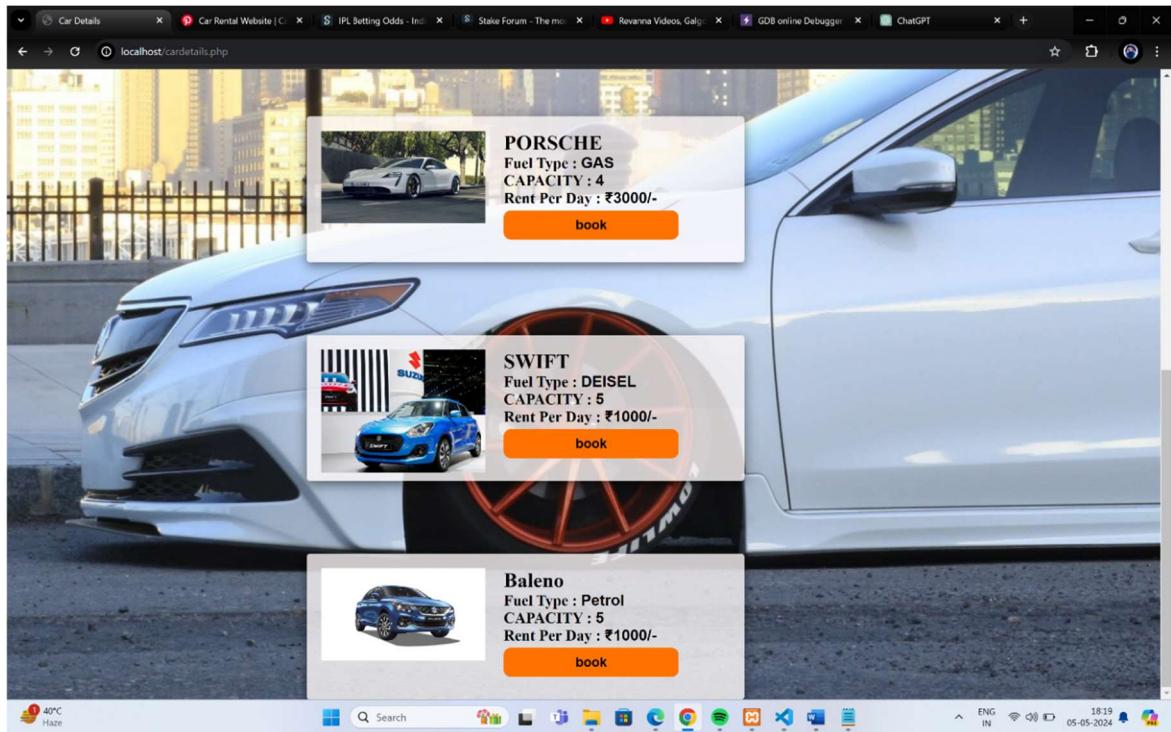


Fig 4.5 View Car Page2

4.4.5 Payment Page

Figure 4.6 indicates Payment page. It asks the user to enter details like card number, expiry date, ccv.

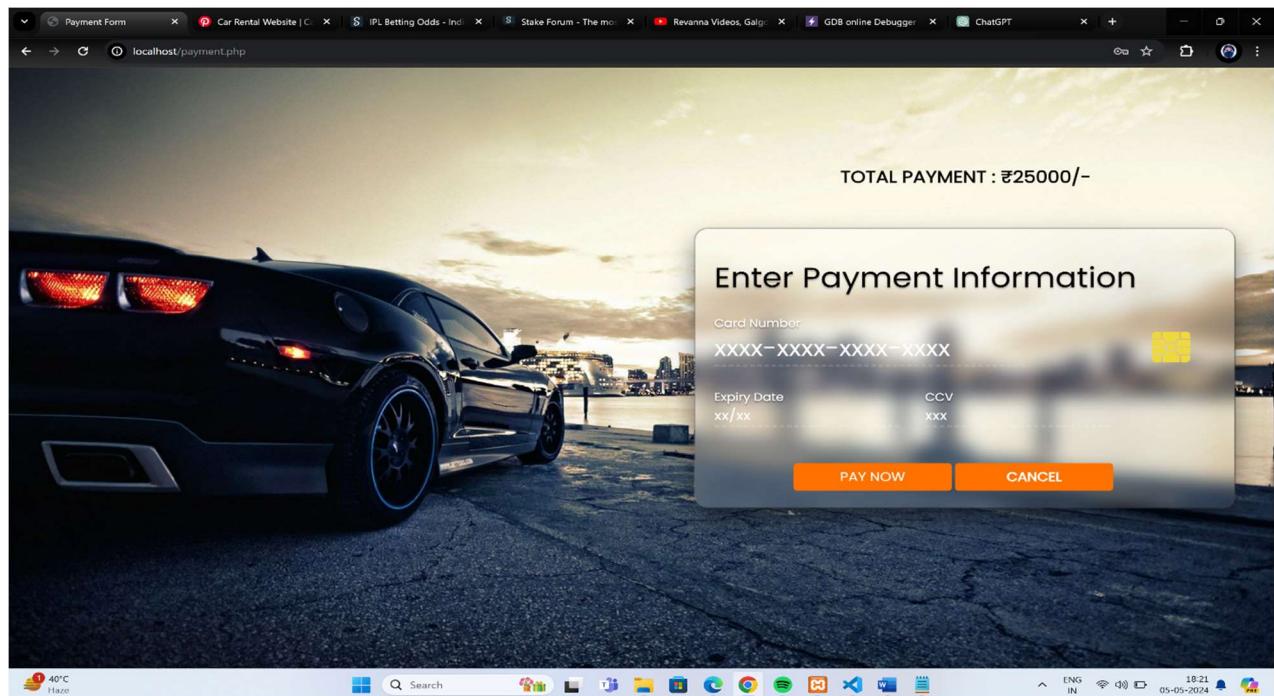


Fig 4.6 Payment Page

4.4.6 Payment Successful Pages

Figure 4.7 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.

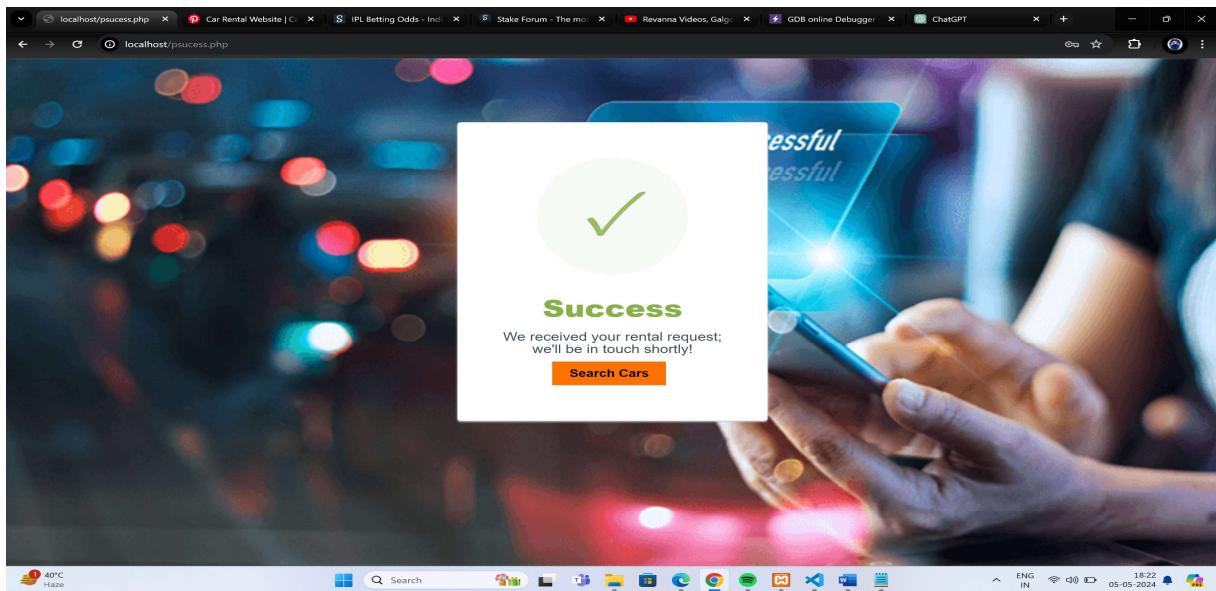


Fig 4.7 Payment Successful Page

4.4.7 Booking status Pages

Figure 4.8 indicates booking status page .It shows user booking details.

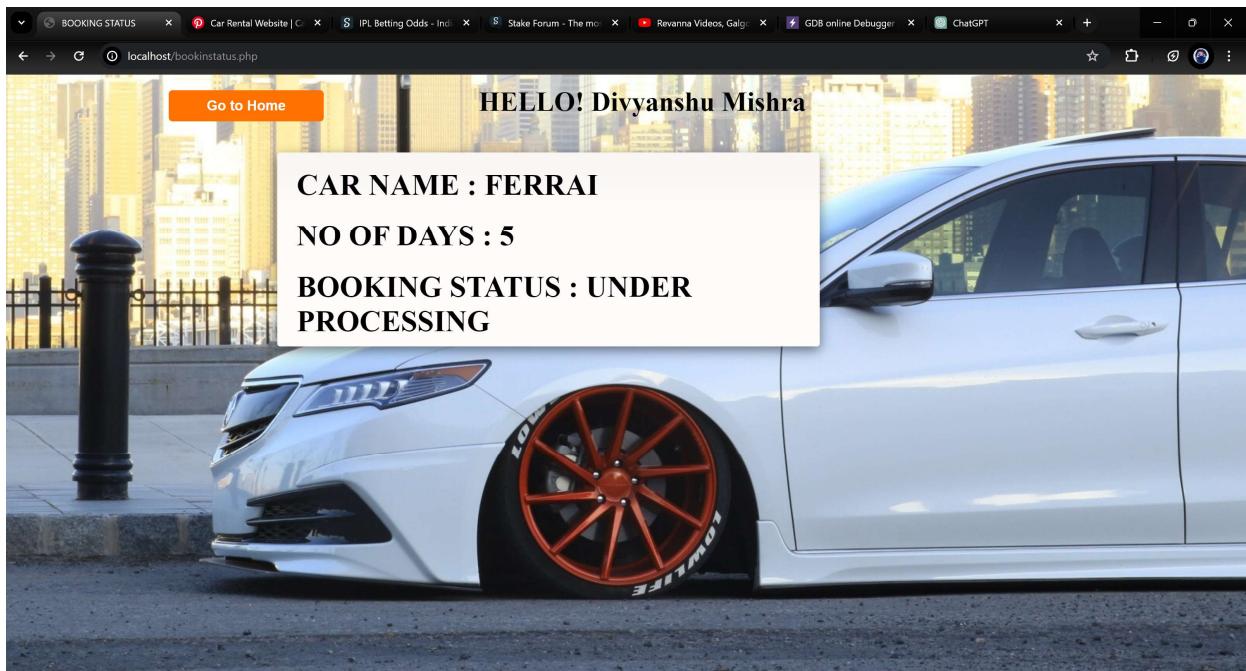


Fig 4.8 Booking Status Page

4.4.8 Feedback Pages

Figure 4.9 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 .

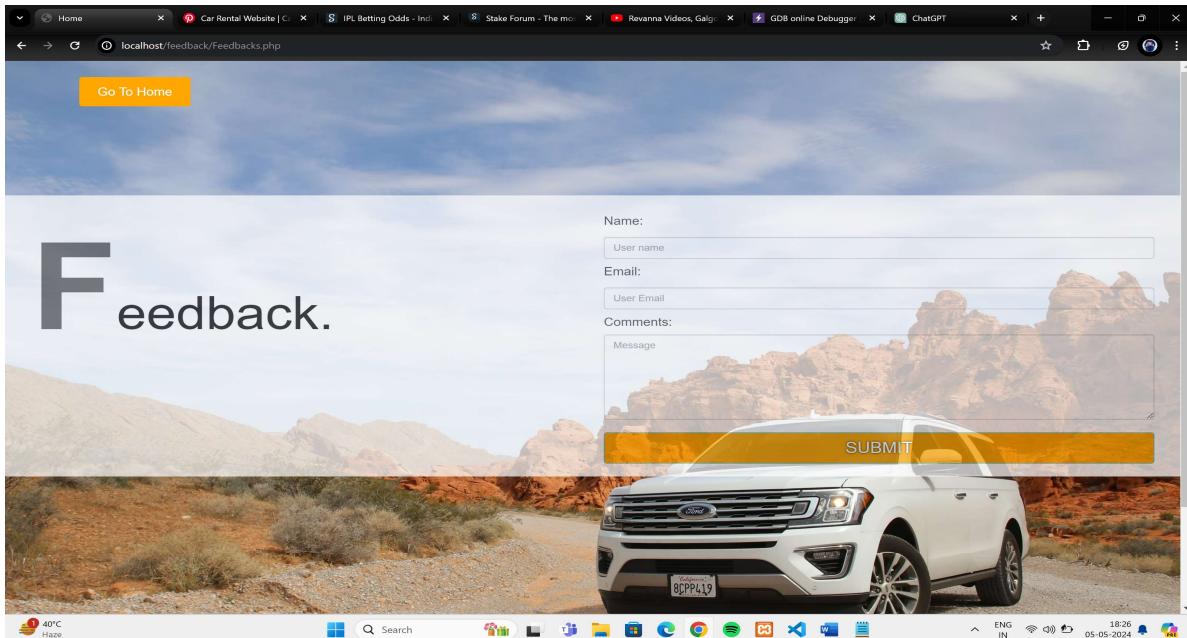


Fig 4.9 Feedback Page

4.4.9 Contact us Pages

Figure 4.10 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.

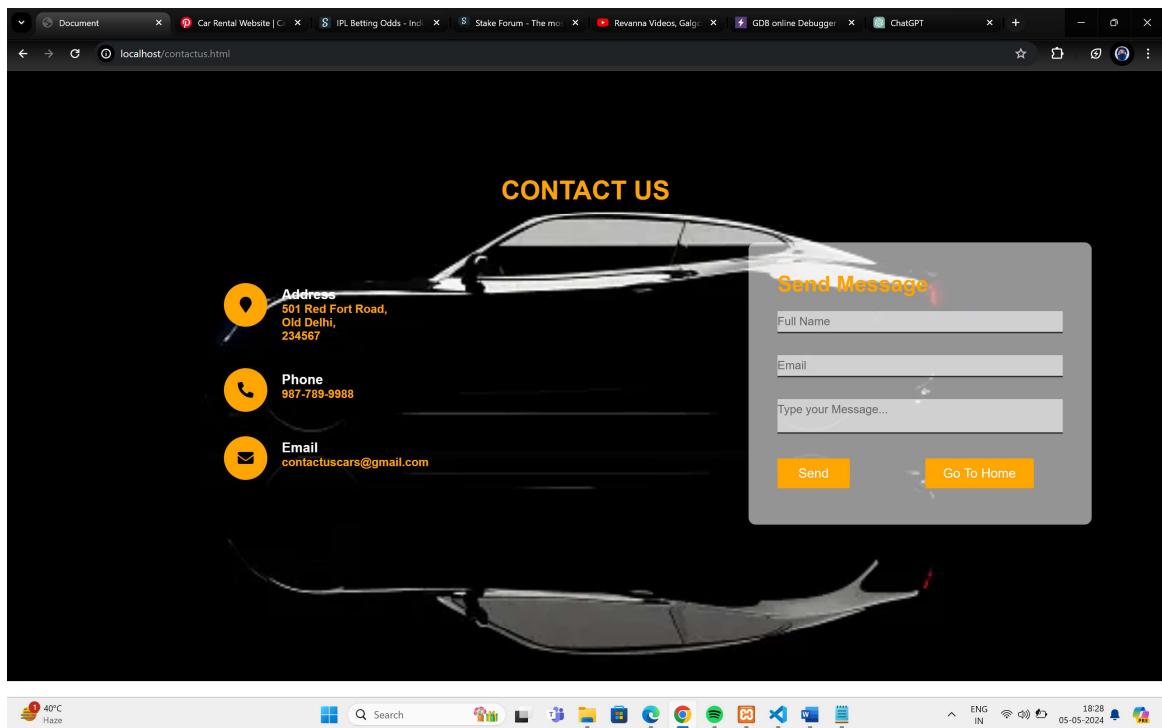


Fig 4.10 Contact Us Page

4.4.10 Admin Login Pages

Figure 4.11 indicates admin login page. It asks the admin to enter details like admin_id and password.

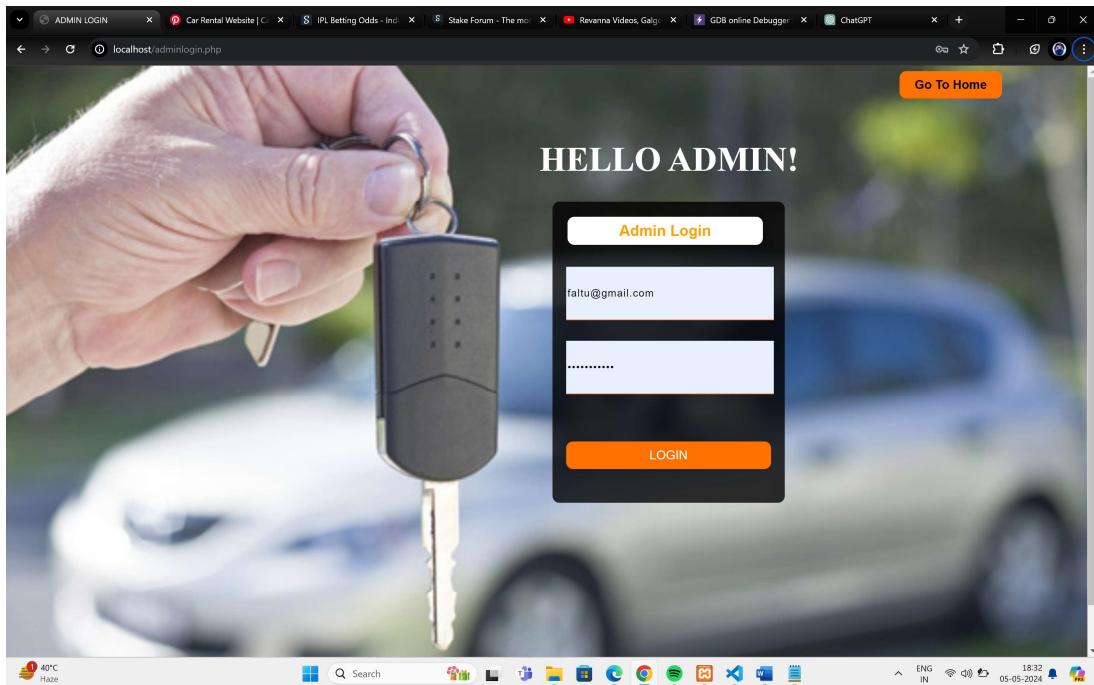


Fig 4.11 Admin Login Page

4.4.11 Vehicle Management Pages

Figure 4.12 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.

A screenshot of a web browser window titled "ADMINISTRATOR". The URL is "localhost/adminvehicle.php". The page has a header with the title "CaRs" in orange, followed by navigation links: "VEHICLE MANAGEMENT", "USERS", "FEEDBACKS", "BOOKING REQUEST", and "LOGOUT". Below the header is a large image of a city skyline at sunset. In the center of the page is a table titled "CARS" with a "ADD CARS" button in an orange box. The table has columns: CAR ID, CAR NAME, FUEL TYPE, CAPACITY, PRICE, AVAILABLE, and DELETE. There are five rows of data:

CAR ID	CAR NAME	FUEL TYPE	CAPACITY	PRICE	AVAILABLE	DELETE
1	FERRAI	PETROL	5	\$000	YES	DELETE CAR
2	LAMBORGINI	DEISEL	6	7000	YES	DELETE CAR
3	PORSCHE	GAS.	4	3000	YES	DELETE CAR
4	SWIFT	DEISEL	5	1000	YES	DELETE CAR
5	Santro	Petrol	5	1000	YES	DELETE CAR

At the bottom of the page is another image of a white car with red wheels. The taskbar at the bottom shows various open tabs and system icons.

Fig 4.12 Vehicle Management Page

4.4.12 Users Pages

Figure 4.13 indicates User page .It includes details like name, email, license no, phone number, gender, delete user where admin can delete the user details.

NAME	EMAIL	LICENSE NUMBER	PHONE NUMBER	GENDER	DELETE USER
Divyanshu Mishra	falta@gmail.com	B2344	9898989898	male	<button>DELETE USER</button>
Swasthik Jain	swasthik@gmail.com	B2343	9845687555	male	<button>DELETE USER</button>
Varshith Hegde	varshithv@gmail.com	B2344	983549133	male	<button>DELETE USER</button>
Varshith hegde	varshithvhegde@gmail.com	ghhdhd	6363549133	male	<button>DELETE USER</button>

Fig 4.13 Users Page

4.4.13 Add Car Pages

Figure 4.14 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.

HOME

Enter Details Of New Car

Car Name :

Fuel Type :

Capacity :

Price :

Car Image :

ADD CAR

Fig 4.14 Add Car Page

4.4.14 Booking Pages

Figure 4.15 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.

CAR ID	EMAIL	BOOK PLACE	BOOK DATE	DURATION	PHONE NUMBER	DESTINATION	RETURN DATE	BOOKING STATUS	APPROVE	CAR RETURNED
1	faltu@gmail.com	Sonepat	2024-05-05	5	9898989898	sonepat	2024-05-11	UNDER PROCESSING	APPROVE	[RETURNED]
1	faltu@gmail.com	PARASIPUR GONDA	2024-05-04	5	9898989898	bhdbn	2024-05-09	RETURNED	APPROVE	[RETURNED]
1	varshithvh@gmail.com	Delhi	2024-04-03	10	6363549133	Kashmere Gate	2024-04-14	RETURNED	APPROVE	[RETURNED]
1	varshithvh@gmail.com	Sonepat	2024-04-01	10	6363549133	bahalgadh	2024-04-10	RETURNED	APPROVE	[RETURNED]
2	swastimk@gmail.com	Bangalore	2024-05-01	5	6363549133	moodabidri	2024-04-06	UNDER PROCESSING	APPROVE	[RETURNED]

Fig 4.16 Booking Page

4.5 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

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/>