



CHRIST

(DEEMED TO BE UNIVERSITY)

BANGALORE • INDIA

MCS281: ADBMS Project Car Rental System

by

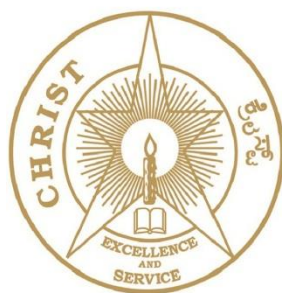
Laxmy Narayan
(2047118)

Rajkumar B L
(2047120)

Under the guidance of
Dr. Sreeja C S

ADBMS project report submitted in partial fulfillment of
the requirements of II semester MSc(Computer Science),
CHRIST(Deemed to be University)

April - 2021



CHRIST

(DEEMED TO BE UNIVERSITY)

BANGALORE • INDIA

CERTIFICATE

*This is to certify that the report titled **Car Rental System** is a bona fide record of work done by **Laxmy Narayan (2047118)** and **Rajkumar B L (2047120)** of CHRIST(Deemed to be University), Bangalore, in partial fulfillment of the requirements of II Semester MSc(Computer Science) during the year 2021.*

Dr. Joy Paulose

Head of the Department

Dr. Sreeja C S

Project Guide

Valued-by:

1.

Name : Rajkumar B L
Register Number : 2047120
Examination Centre : CHRIST(Deemed to
University)

2.

Date of Exam :

ACKNOWLEDGEMENTS

First and foremost, all praises and thanks to the forever merciful and graceful God who bestowed upon me the courage, patience and strength to embark upon this work and carry it to completion.

I express my special gratitude to my parents for their moral support and constant encouragement.

I express our sincere thanks to the Vice Chancellor Rev. **Fr. Abraham Vettiyankal M** of CHRIST (Deemed to be University), Bangalore for providing all facilities in our college for successful completion of this project work.

We pay our deep sense of gratitude to **Dr. Joy Paulose** Head of the Computer Science Department and Statistics, CHRIST (Deemed to be University), Bangalore to encourage us to the highest peak and to provide us the opportunity to prepare the project.

I feel privileged to express sincere and deepest gratitude to my guide, **Dr. Sreeja C S**, for providing me the opportunity to carry out this work under her invaluable guidance and continuous support.

We are immensely obliged to our friends for their elevating inspiration, encouraging guidance and kind supervision in the completion of my project.

I would like to thank all other teaching and non-teaching staffs of the Department of Computer Science, CHRIST (Deemed to be University), for their support and encouragement during the course of study.

I am much grateful to one and all those who directly or indirectly rendered their help for the completion of the project in time.

RAJKUMAR B L

ABSTRACT

This application is designed so as to be used by Car Rental Company specializing in renting cars to customers. It is an online system through which customers can view available cars, register, view profile and book car.

The Car Rental System is being developed for customers so that they can book their desired cars from any part of the world. This application collects information from the customers through them filling their details in the registration page. A customer being registered with the company has the facility to look the availability of different cars at a given range of date and book the desired the vehicle.

The proposed system is completely integrated online system. It automates manual procedure in an effective and efficient way. This automated system facilitates customer and provides option to fill up the details according to their requirements. It includes type of vehicle the customer is trying to rent and the location.

The purpose of this project is to develop a web-based application for an car rental business. The project objective is to deliver the online web application that ease customer task to register and reserve car online and for the company to effectively manage their car rental business.

LIST OF FIGURES

Fig. No.	Figure Name	Page No.
3.1	System Model	4
4.1	Architectural Design	7
4.2	Data Flow Diagram : Level 0	10
4.3	Data Flow Diagram : Level 1	10
4.4	Data Flow Diagram : Level 2	11
4.5	Entity Relationship Diagram	12
4.6	Use Case Diagram	13
4.7	Class Diagram	13
4.8	Object Diagram	14

LIST OF TABLES

Tab. No.	Table Name	Page No.
4.2.1	Customer	8
4.2.2	Login Credentials	8
4.2.3	Location	8
4.2.4	Car	9
4.2.5	Car Availability	9
4.2.6	Booking	9
4.2.7	Customer_Rent_Car	9
4.2.8	Billing	10

Table of Contents

Acknowledgements	iii
Abstract	iv
List of Figures	v
List of Tables	vi
1. Introduction	
1.1 Overview of the System	
2. System Analysis	
2.1 Existing System	
2.2 Proposed System	
2.3 Software Tools Used	
3. System Requirements	
3.1 System Model	
3.2 Functional Requirements	
3.2 Hardware Requirements	
3.3 Software Requirements	
4. Design Specification	
4.1 Architectural Design	
4.2 Database Design	
4.2.1 Table Design	
4.2.2 Data Flow Diagrams	
4.2.3 Entity Relationship Diagrams	
4.2.4 Use Case Diagrams	
4.2.5 Class Diagram	
4.2.6 Object Diagram	
4.3 User Interface Design	
4.4 Report Design	
5. Implementation	
5.1 Source Code	
5.2 Screen shots	
6. Testing	
6.1 Test Plan	
6.2 Test Case AND Test Reports	
7. Conclusion	
7.1 Advantages	
7.2 Limitations	
7.3 Future Enhancement	

1. INTRODUCTION

1.1 OVERVIEW OF THE SYSTEM

This project focuses on creating a web-based application for a company to run a car rental business. It would be designed as an online portal where customers can view available vehicles based on their make, model, and cost, register for an account with the company, and book a suitable car.

There are a lot of small-scale car rental companies that do not have a online web-portal available for the customers to view and book desired cars. This limits the company's projection to the local customers, and they have to come to the company's office to know about the availability or to rent a car. Our car rental system will break these limitations with a web-based system that will help the customers view and rent the desired vehicles on the go. A registration portal to hold customer's details, monitor their transaction and used same to offer better and improve services to them.

The company owned cars are parked and made available to the customers at hotspots of the city like malls, showrooms, service centers. The customers can book the car near their location or at their preferred location. The company facilitates the customers with the following four easy steps (Register, Reserve, Unlock&Go, Return). The service provided helps the company go global and increase its income by attracting more customers.

There would be registration fee of Rs.5000 to all the members registering with the car rental system. This will be taken as deposit and will be used when required. The deposit fee will act as emergency fund that will be used if the user has not paid their due in long time or if there has been any damages caused to the vehicle during a booking.

The car rental system aims 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.

2. SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

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 to rent a car must contact a rental car company and contract out for a car.

There are a lot of car rental companies that do not have a website available for the customers to view and book desired cars. This limits the company's projection to the local customers, and they have to come to the company's office to know about the availability or to rent a car.

Many existing sites which provide details about various brands often do not have the facility for its users to opt for cleaning of vehicles before renting them and most of the times they lack to provide support for the customer bookings during emergency situations.

The Existing system does not have much provision for decision making. Any person who wishes to rent a new car may not be aware of the different features available. It thus becomes difficult to choose the right car which can satisfy all needs. Also, in the existing system, there is less provision on road assistance. Users are left stranded in case of car breakdowns and face less service.

2.2 PROPOSED SYSTEM

The advancement in Information Technology and internet penetration has greatly enhanced various business processes and communication between companies (services provider) and their customers of which car rental industry is not left out.

The proposed system focuses on creating a web-based application for a company to run a car rental business. It would be designed as an online portal where customers can view available vehicles based on their make, model, and cost, register for an account with the company, and book a suitable car.

Our project aims at creating a user-friendly interface where users can easily look for the desired cars availability and rent them based on their required dates.

It will break all traditional limitations with a web-based system that will help the customers view and rent the desired vehicles on the go. The company owned cars are parked and made available to the customers at hotspots of the city like malls, showrooms, service centers. The customers can book the car near their location or at their preferred location and access them. The company facilitates the customers with the following four easy steps (Register, Reserve, Unlock&Go, Return). The service provided helps the company go global and increase its income by attracting more customers.

2.3 SOFTWARE TOOLS USED

2.3.1 SERVER SIDE

- MySQL: Database to store, retrieve, update and delete customer details accordingly.
- Xamp, PHPMyAdmin: Tool written in PHP, intended to handle MySQL administration over the web.
- Java: For the UI design and the functionalities to develop an interactive webpage.
- CSS: For styling the web pages, including colors, layout, and fonts.
- JDBC: Java Database Connectivity, a standard Java API for database-independent connectivity between the Java programming language and the database.
- IDE: NetBeans is used for the development.

2.3.2 CLIENT SIDE

- A fully functional laptop/desktop or any handheld device like smartphones, tablet, iPad, etc. with an active internet connection with windows operating system or any other GUI based operating system.
- Any web-browser that supports HTML, CSS, and PHP web pages with internet connectivity to access the companys website.

3. SYSTEM REQUIREMENTS

3.1 SYSTEM MODEL

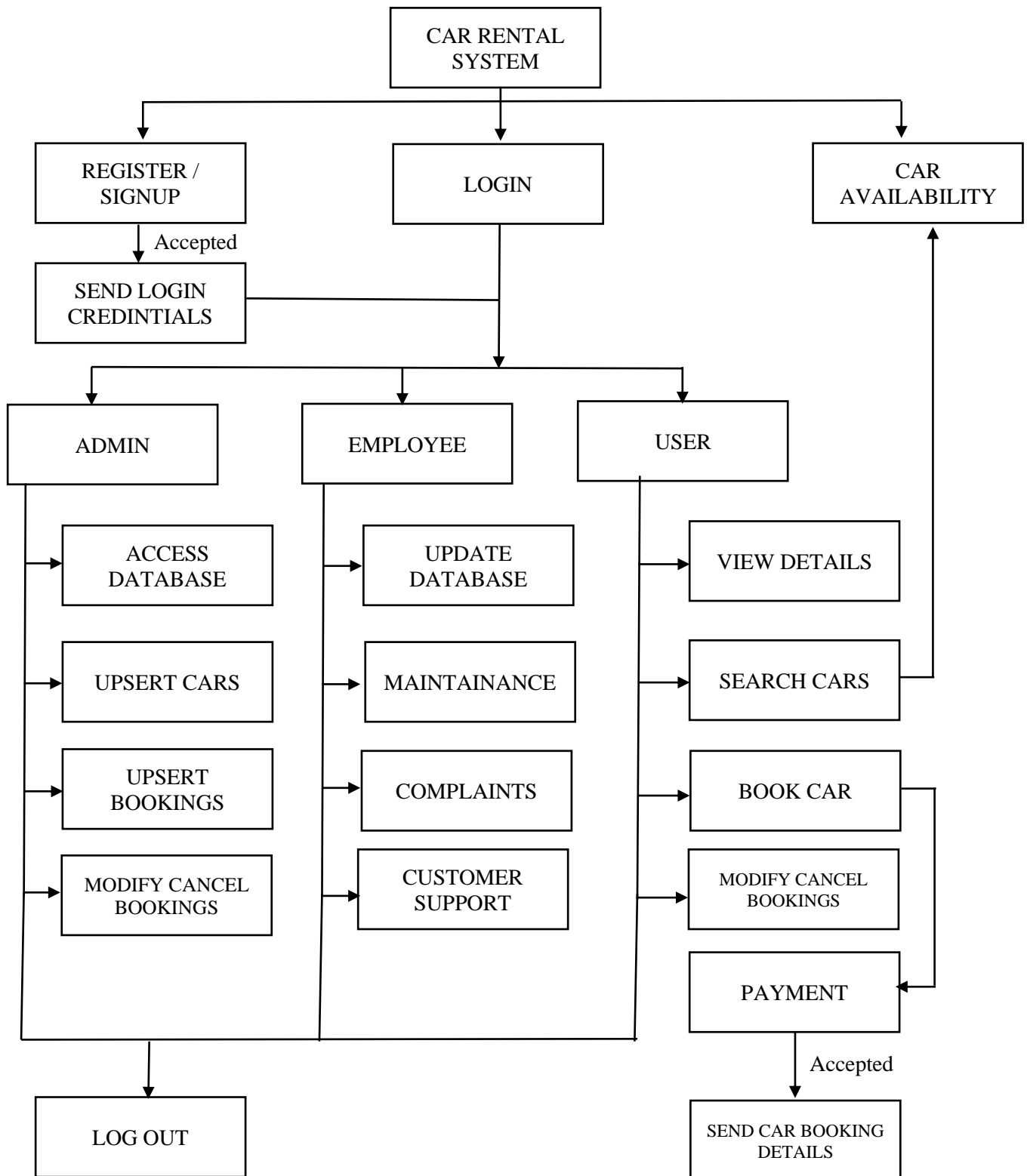


Figure 3.1: System Model

3.2 FUNCTIONAL REQUIREMENTS

Illustrates the working nature of the system:

- The system is available for registered and non-registered customers to view the available cars and their details like make, model, cost, availability time, etc.
- The customers need to register with the system to be able to rent a car. The registration includes verification of the age and driver's license of the customer. The customer can optionally opt for the yearly membership.
- Registered customers can rent a car through online reservations of the system.
- Successful reservation updates the database, thus changing that particular car's availability based on the reservation details.
- Customers should be able to edit or cancel a booking one day before the booked date.
- The cost should be calculated and made available to the customer at the time of booking with the appropriate discounts applied for customers with memberships.
- Any additional cost like damage fee or late return fee should be added and billed upon return of the car.
- Customers will be able to apply for cleaning and sanitizing through the cleaning service provided in the system.
- The availability of cars will be shown only if the vehicle has been returned and marked as available for use.
- Customers have to add a fixed amount of money to the wallet for booking and renting cars. The total fee is payable within 48 hours after the completion of the trip. The payment can be made using a debit card, credit card, or other online transaction modes are available as well.
- Cars will be available in nearby hotspots areas like malls, showrooms, or service centers.

3.3 HARDWARE REQUIREMENTS

Both Mac and Windows operating systems can be used for the development. Use of a standard android system that provides computational power for all the required processing.

Server Side:

- Hard Disk: Minimum 500GB Disk Space
- RAM: Minimum 16GB
- OS: Windows 10, Mac

Client Side:

- Hard Disk: Minimum 80GB Disk Space
- RAM: Minimum 2GB
- OS: Windows 10, Mac

Besides, the system will use the standard hardware and data communication resources. This includes, but is not limited to, general network connection at the server/hosting site, network server, and network management tools.

3.4 SOFTWARE REQUIREMENTS

- Frontend: HTML, CSS, Java.
- Database: MySQL, Xamp, PHPMyAdmin.
- IDE: NetBeans.
- Others: Apache Server.

4. DESIGN SPECIFICATIONS

4.1 ARCHITECTURAL DESIGN

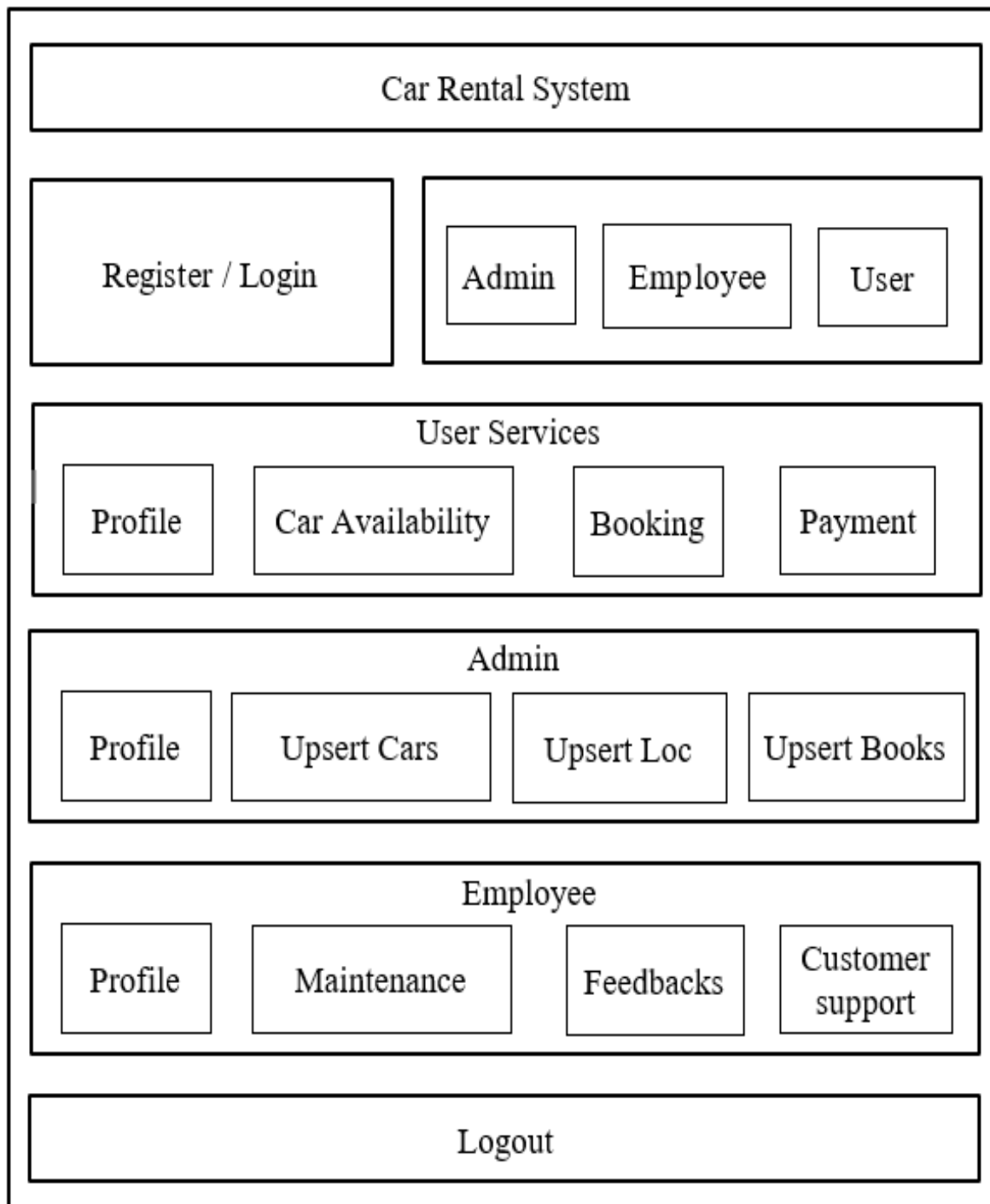


Figure 4.1: Architectural Design

4.2 DATABASE DESIGN

TABLE DESIGN

4.2.1 Table: Customer

Primary Key : UserID

S.NO	FIELD NAME	DATA TYPE	CONSTRAINTS
1	UserID	Integer(7)	Unique
2	FisrtName	Varchar(30)	Not Null
3	LastName	Varchar(20)	Nullable
4	Email	Varchar(70)	Not Null
5	PhoneNumber	Varchar(10)	Not Null
6	DOB	date	Not Null
7	DLnumber	Varchar(15)	Unique
8	Street	Varchar(10)	Not Null
9	City	Varchar(10)	Not Null
10	State	Varchar(10)	Not Null
11	Zipcode	Integer(6)	Not Null

4.2.2 Table: Login Credentials

Primary, Foreign Key : CustomerUserID

S.NO	FIELD NAME	DATA TYPE	CONSTRAINTS
1	CustomerUserID	Integer(7)	Unique
2	Password	Varchar(12)	Not Null

4.2.3 Table: Location

Primary Key : LocID

S.NO	FIELD NAME	DATA TYPE	CONSTRAINTS
1	LocID	Integer(7)	Unique
2	Street	Varchar(10)	Not Null
3	City	Varchar(10)	Not Null
4	State	Varchar(10)	Not Null
5	Zipcode	Integer(6)	Not Null

4.2.4 Table: Car

Primary Key : CarID

Foreign Key : LocID

S.NO	FIELD NAME	DATA TYPE	CONSTRAINTS
1	CarID	Integer(4)	Unique
2	LocID	Integer(7)	Unique
3	CarCategory	Varchar(8)	Unique
4	ModelName	Varchar(30)	Not Null
5	ModelYear	Integer(5)	Not Null
6	PersonIncudable	Integer(2)	Not Null
7	CostPerDay	Integer(5)	Not Null

4.2.5 Table: Car Availability

Primary, Foreign Key : CarID

S.NO	FIELD NAME	DATA TYPE	CONSTRAINTS
1	CarID	Integer(7)	Unique
2	Availflag	Tinyint	Not Null

4.2.6 Table: Booking

Primary Key : BookingID

S.NO	FIELD NAME	DATA TYPE	CONSTRAINTS
1	BookingID	Integer(4)	Unique
2	PickDt	Timestamp	Not Null
3	ReturnDt	Timestamp	Null
4	Status	Varchar(20)	Not Null
5	CarDetails	Varchar(20)	Not Null
6	Amount	Integer(10)	Not Null
7	CustomerDetails	Integer(7)	Not Null

4.2.7 Table: Customer_Rent_Car

Primary, Foreign Key : CustomerUserID, CarID

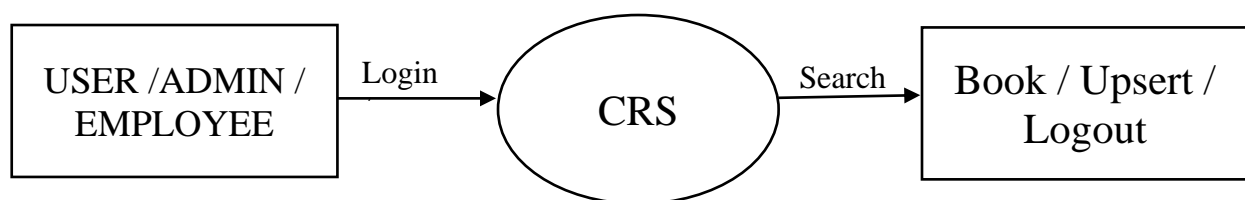
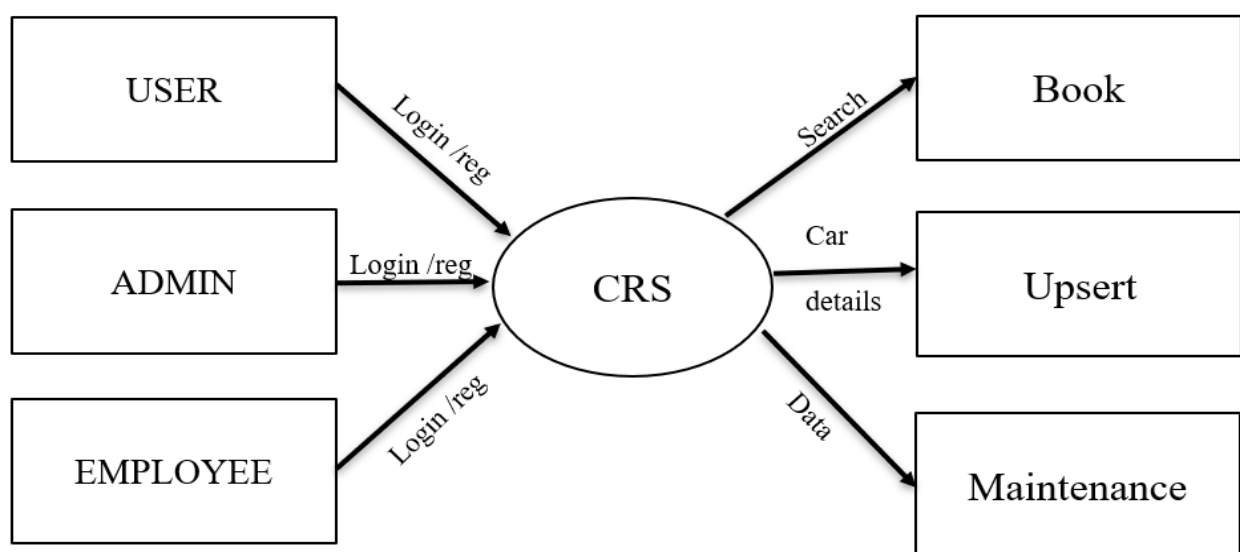
S.NO	FIELD NAME	DATA TYPE	CONSTRAINTS
1	CustomerUserID	Integer(7)	Unique
2	CarID	Integer(4)	Unique
3	BookingID	Integer(12)	Unique

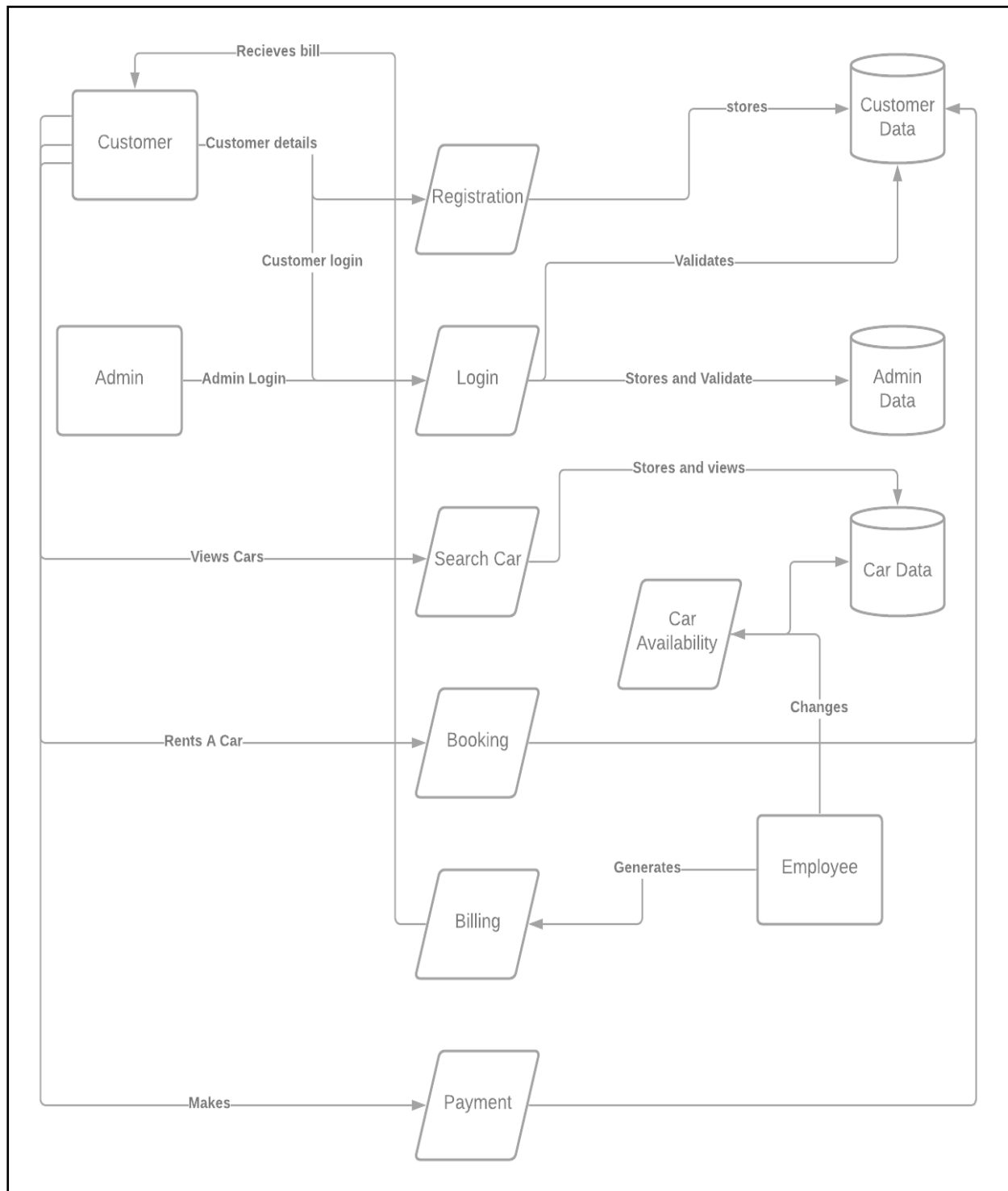
4.2.8 Table: Billing

Primary Key : BillID

Foreign Key : BookingID

S.NO	FIELD NAME	DATA TYPE	CONSTRAINTS
1	BillID	Integer(12)	Unique
2	BookingID	Integer(12)	Unique
3	BillDate	timestamp	Not Null
4	BillStatus	Varchar(20)	Not Null
5	TotalAmt	Integer(10)	Not Null

4.2.2 DATA FLOW DIAGRAM**4.2 DATA FLOW DIAGRAM : LEVEL 0***Figure 4.2.2 (A): Data Flow Diagram – Level 0***4.3 DATA FLOW DIAGRAM : LEVEL 1***Figure 4.2.2 (B): Data Flow Diagram – Level 1*

4.4 DATA FLOW DIAGRAM : LEVEL 2*Figure 4.2.2 (C): Data Flow Diagram – Level 2*

4.5 ENTITY RELATIONSHIP DIAGRAM

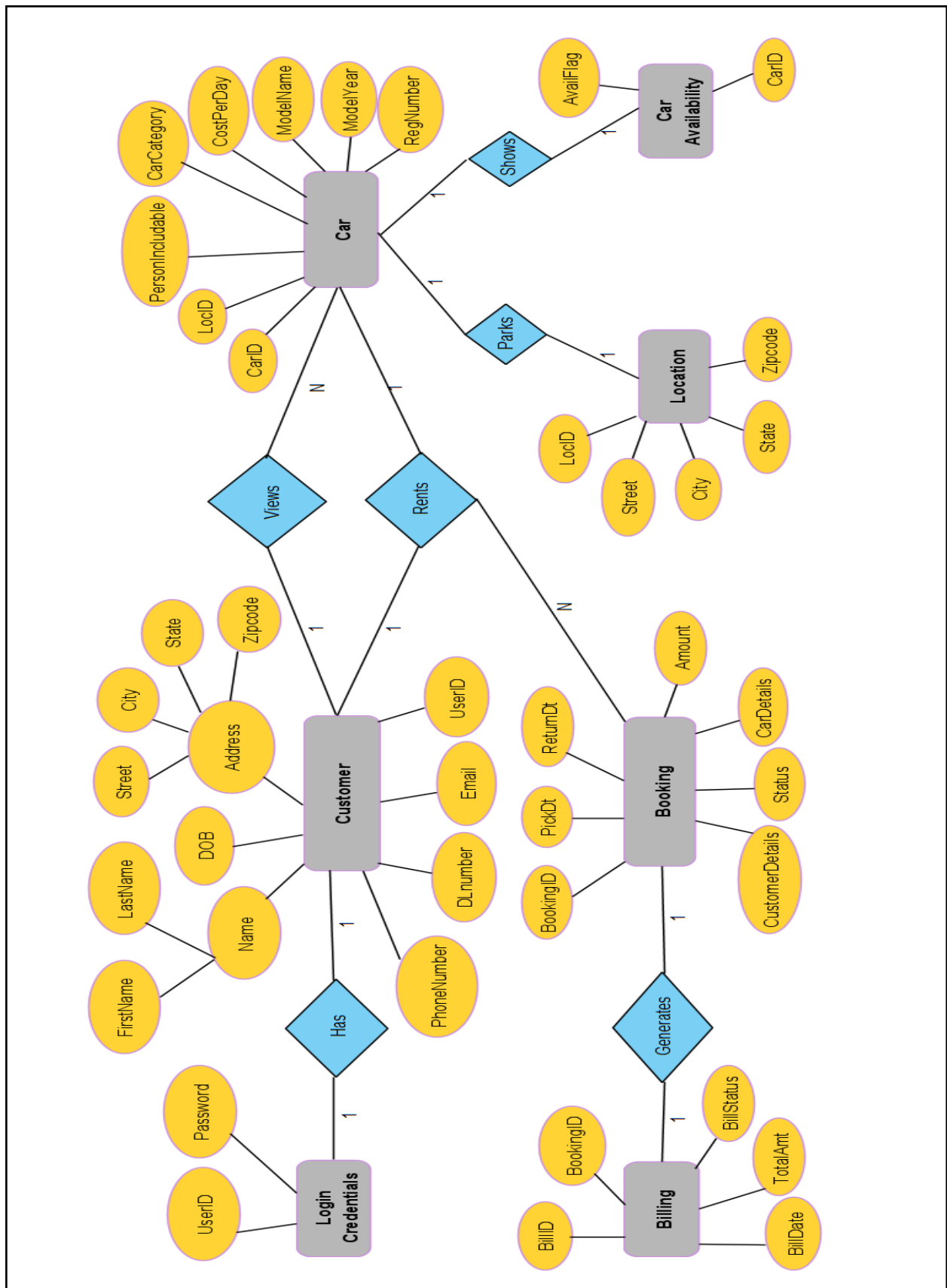


Figure 4.2.3: Entity Relationship Diagram

4.6 USE CASE DIAGRAM

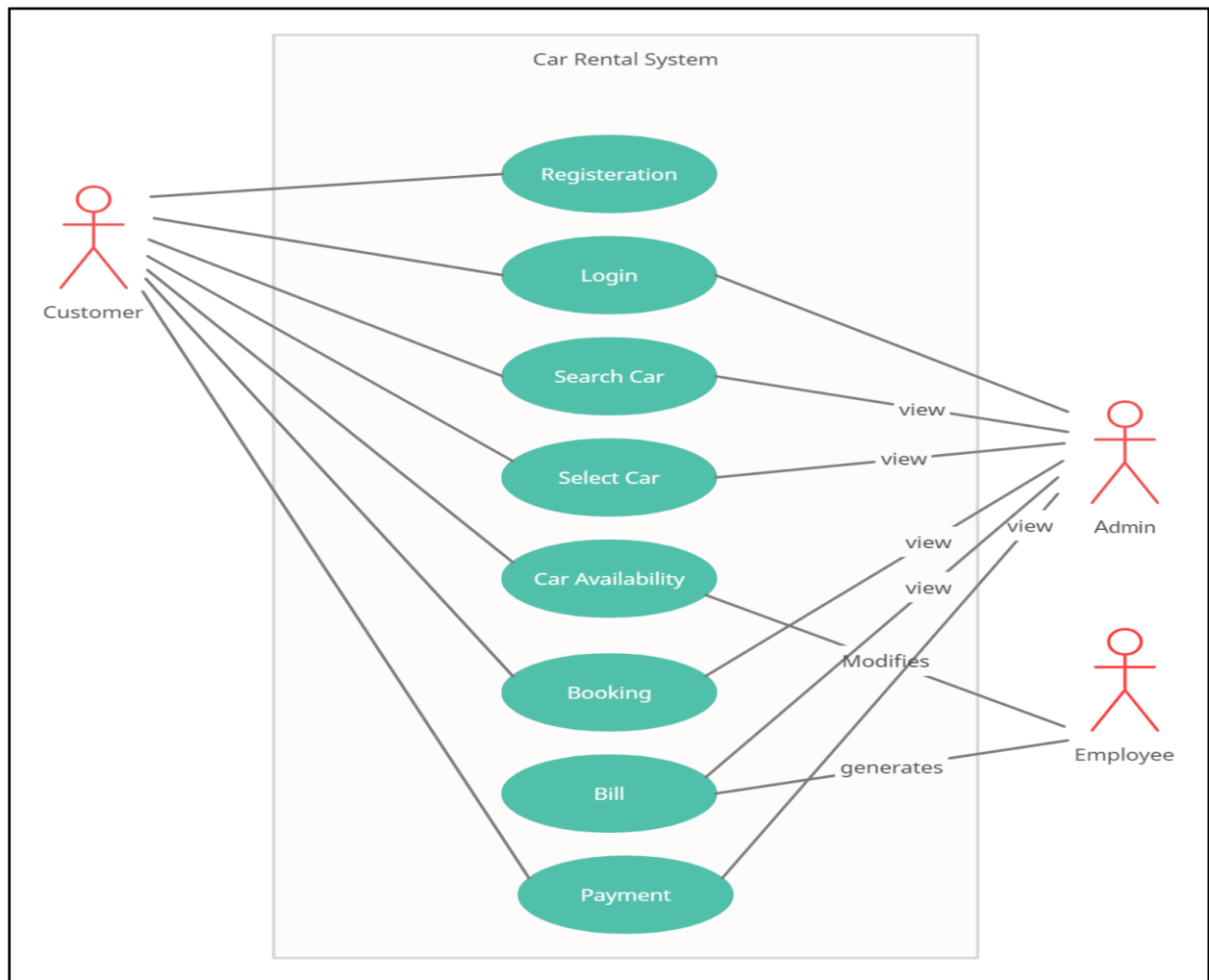


Figure 4.2.4: Use Case Diagram

4.7 CLASS DIAGRAM

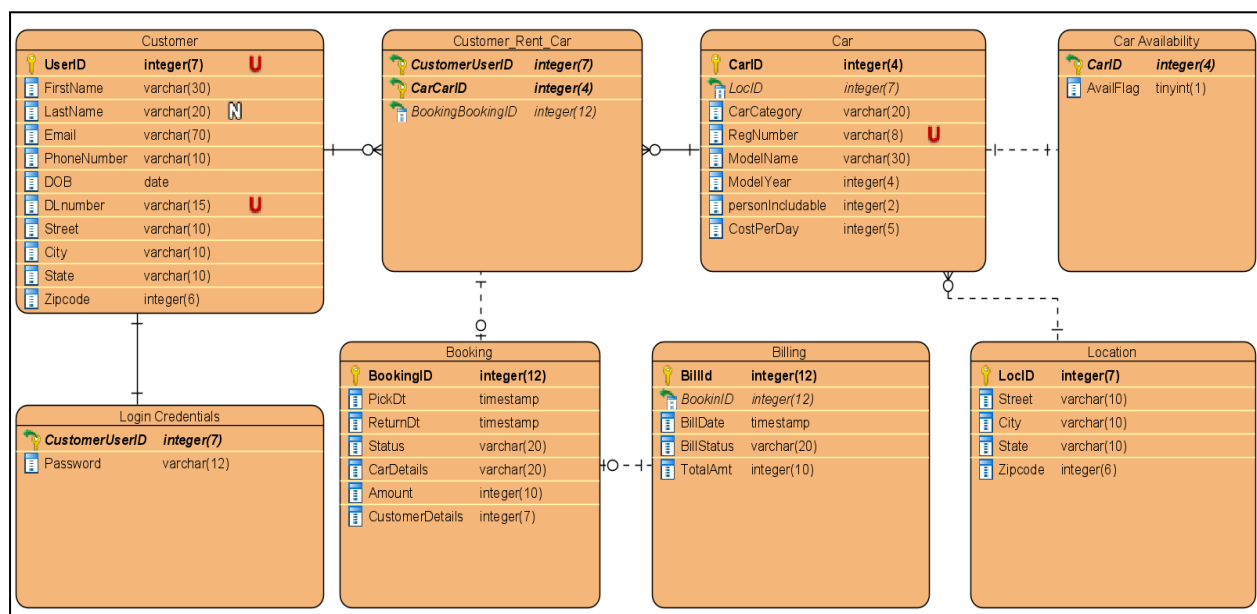


Figure 4.2.5: Class Diagram

4.8 OBJECT DIAGRAM

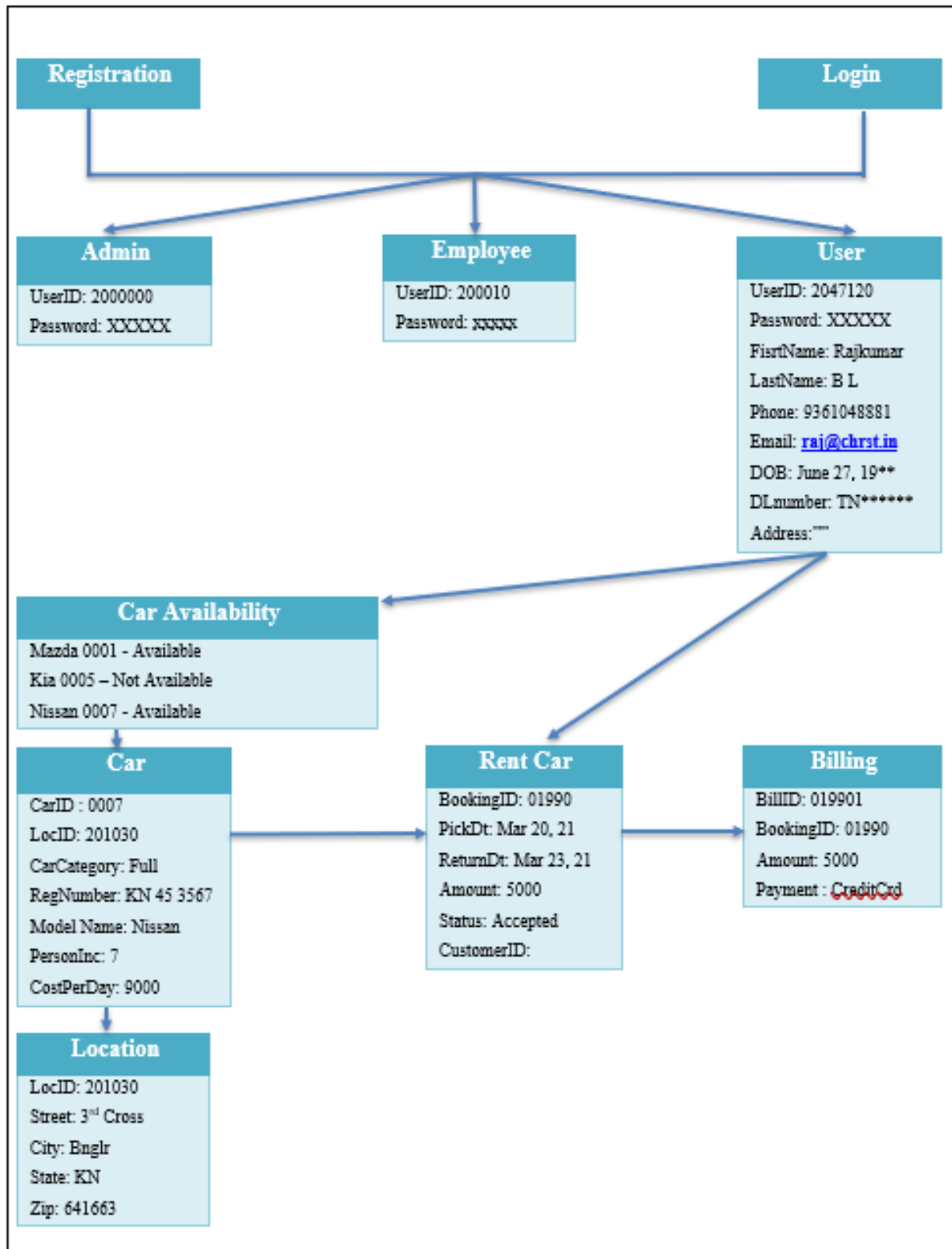
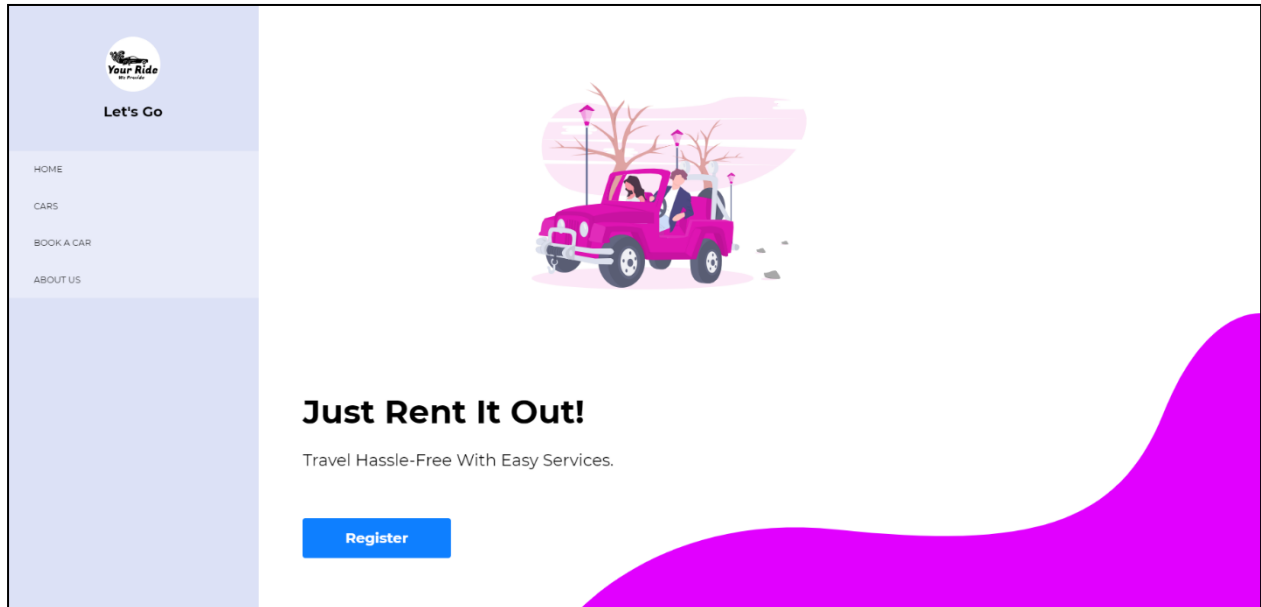


Figure 4.2.6: Object Diagram

4.3 USER INTERFACE DESIGN

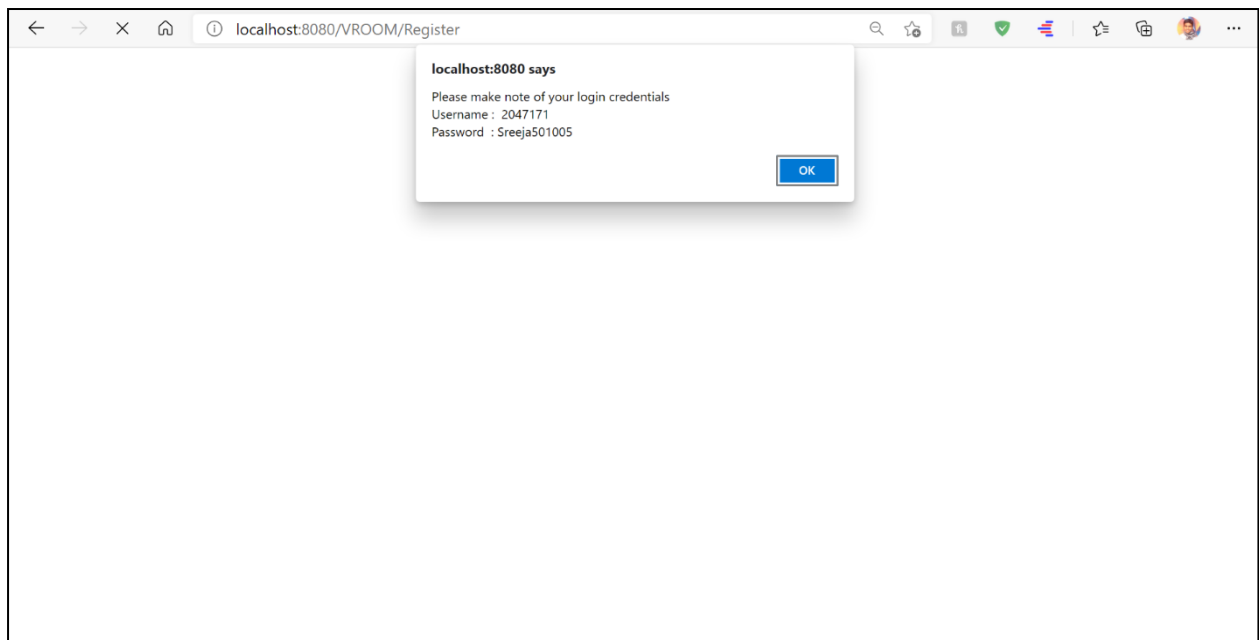
HOME PAGE



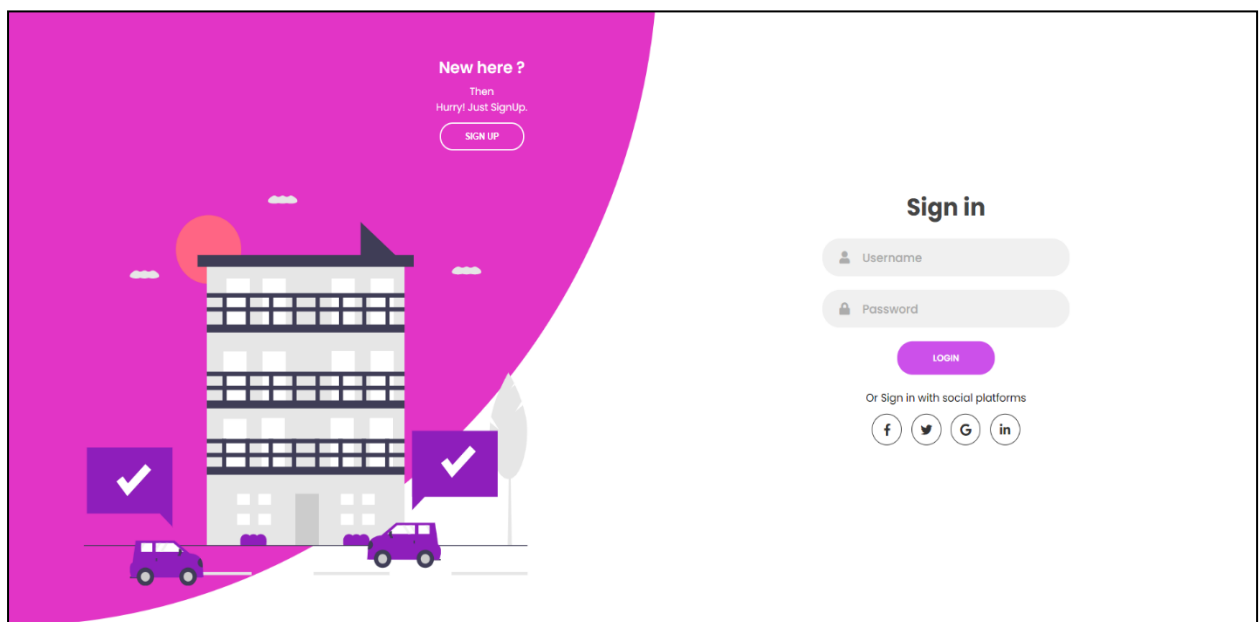
REGISTER

The register page has a white background with a pink sidebar on the right. The sidebar contains the text 'One of us?' and 'Then start Booking And Enjoy Your Journey.' with a 'LOGIN' button. The main content area is titled 'Register' and contains a form with the following fields: Name (Sreeja), Surname (C S), Email (Sreeja@crs.in), Phone (9361048881), Date of Birth (1998-06-18), License Number (KA12386769), Full Name (S G PALYA), City (Bengaluru), State (KA), and Pincode (501005). A pink 'SIGN UP' button is at the bottom of the form. The sidebar also features an illustration of a person sitting on a large black box with white lines, set against a pink background with stylized trees.

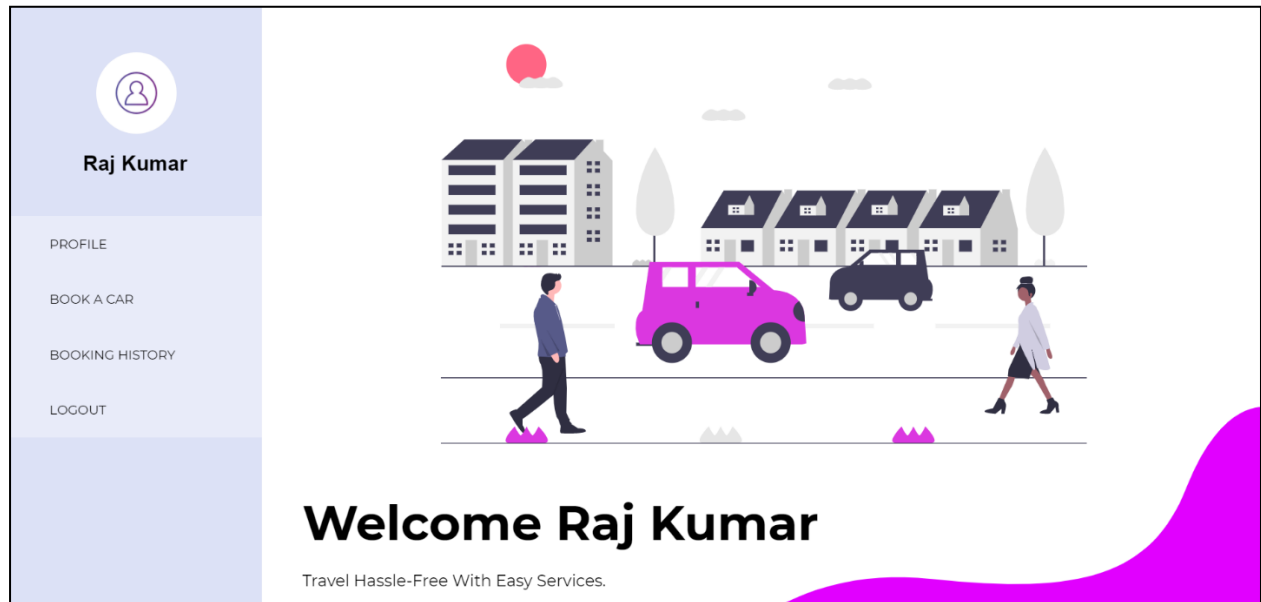
REGISTRATION CONFIRMATION



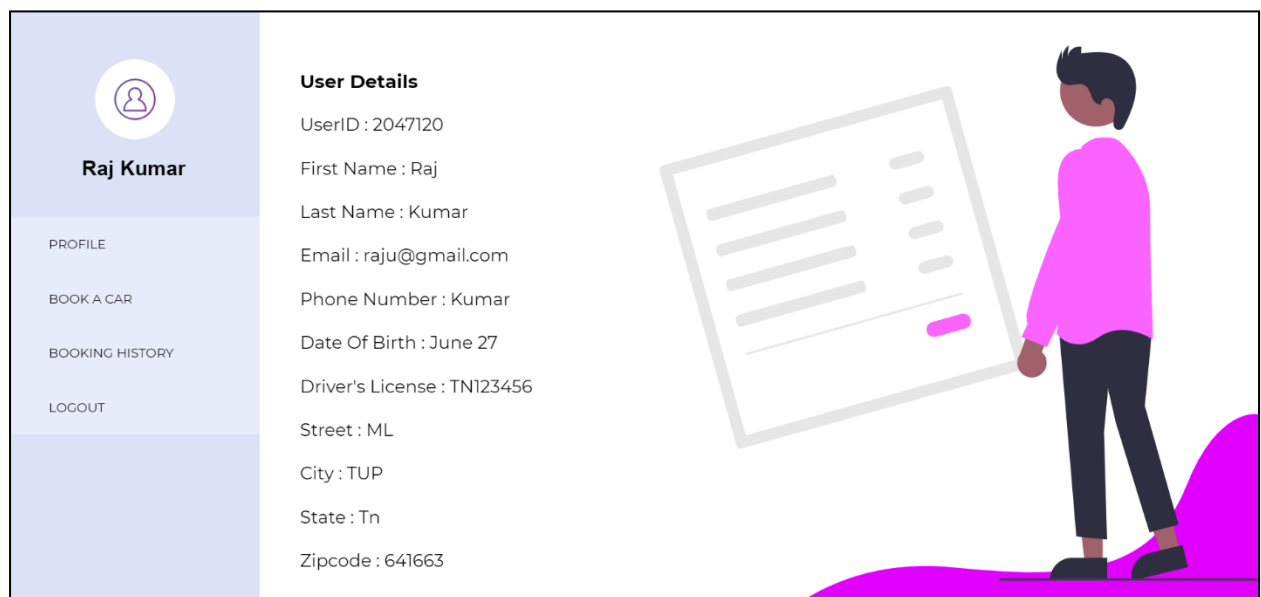
LOGIN



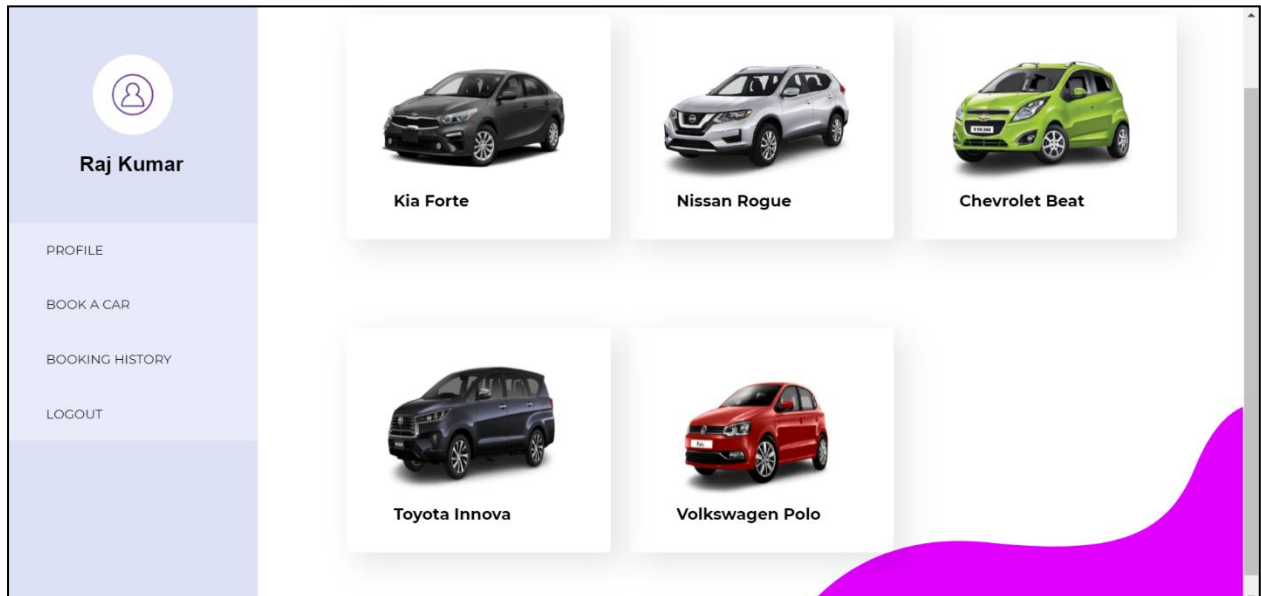
USER WELCOME PAGE



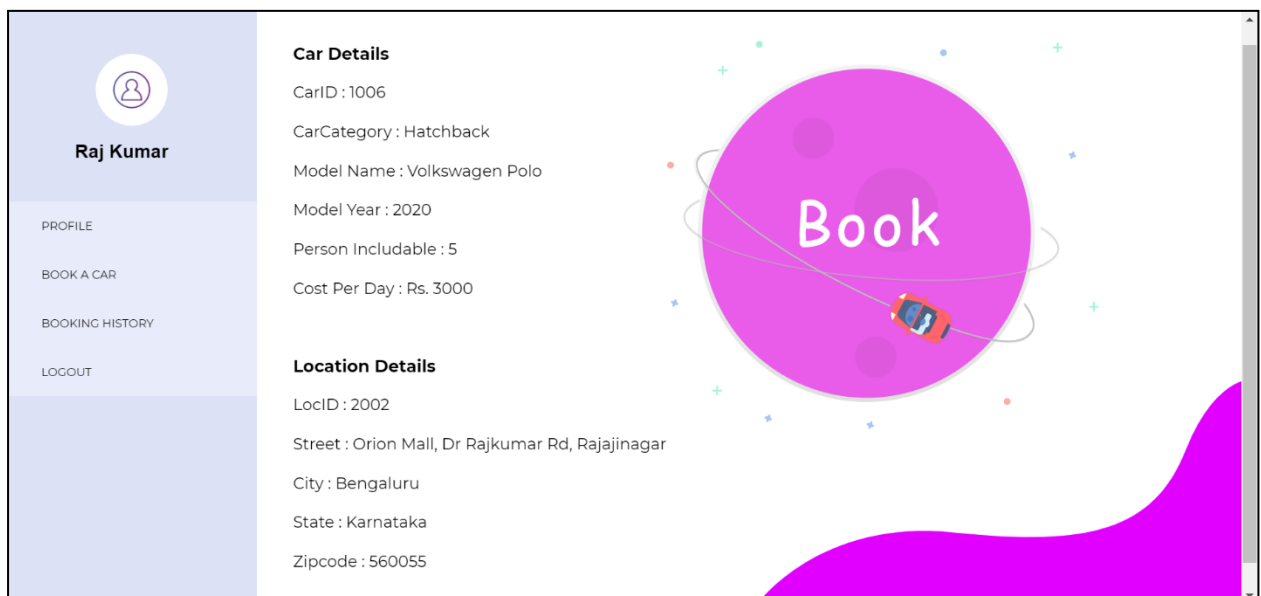
USER PROFILE



CAR AVAILABILITY



SELECTIVE CAR DETAILS



4.4 REPORT DESIGN

VROOM CAR SHARE - BOOKING HISTORY

Name: RajKumar **User-ID:** 2047120

Booking : 1
Booking ID : 51794598
Billing ID : 656998589
Car Model : Nissan Rogue
Total Amount : 5500
Pick Up Date : 2021-03-04
Return Date : 2021-03-04
Location : The Forum Mall, CL Layout



Booking : 2
Booking ID : 81634136
Billing ID : 459838343
Car Model : Kia Forte
Total Amount : 56000
Pick Up Date : 2021-03-12
Return Date : 2021-03-27
Location : The Forum Mall, CL Layout



Booking : 3
Booking ID : 276353068
Billing ID : 322097492
Car Model : Chevrolet Beat
Total Amount : 2500
Pick Up Date : 2021-03-05
Return Date : 2021-03-06
Location : The Forum Mall, CL Layout



5. IMPLEMENTATION

5.1 SOURCE CODE

5.1.1 HTML CODE

Index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <link rel="stylesheet" href="css/main.css">
    <script
src="https://kit.fontawesome.com/64d58efce2.js"
crossorigin="anonymous"
></script>
    <title>CRS</title>
  </head>
  <body>
    <aside>
      <figure>
        <div id="avatar"></div>
        <figcaption>Let's Go</figcaption>
      </figure>
      
      <nav>
        <ul>
          <li><a href="index.html">Home</a></li>
          <li><a href="/CarAvail">Cars</a></li>
          <li><a href="booking.html">Book A Car</a></li>
          <li><a href="#">About Us</a></li>
          <li><a href="#">Contact Us</a></li>
        </ul>
      </nav>
    </aside>
    <main>
      <div class="logosec">
      </div>
      <h1>Just Rent It Out!</h1>
      <p>Travel Hassle-Free With Easy Services.</p>
      <a href="signin-signup.html" class="cta">Login / Register</a>
    </main>
```

```
<script>

(function() {
  var menu = document.querySelector('ul'),
      menulink = document.querySelector('img');

  menulink.addEventListener('click', function(e) {
    menu.classList.toggle('active');
    e.preventDefault();
  });
})();
</script>
</body>
</html>
```

Signin-signup.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <script
    src="https://kit.fontawesome.com/64d58efce2.js"
    crossorigin="anonymous"
  ></script>
  <link rel="stylesheet" href="css/style.css" />
  <title>SignIn&SignUp</title>
</head>
<body>
  <div class="container">
    <div class="forms-container">
      <div class="signin-signup">
        <form method="post" action="Login" class="sign-in-form">
          <h2 class="title">Login</h2>
          <div class="input-field">
            <i class="fas fa-user"></i>
            <input type="text" placeholder="UserId" name="userid" required/>
          </div>
          <div class="input-field">
            <i class="fas fa-lock"></i>
            <input type="password" placeholder="Password" name="password" required/>
          </div>
          <input type="submit" value="Login" class="btn solid" />
          <p class="social-text">Or Sign in with social platforms</p>
          <div class="social-media">
            <a href="#" class="social-icon">
              <i class="fab fa-facebook-f"></i>
            </a>
          </div>
        </form>
      </div>
    </div>
  </div>
```

```

    </a>
    <a href="#" class="social-icon">
      <i class="fab fa-twitter"></i>
    </a>
    <a href="#" class="social-icon">
      <i class="fab fa-google"></i>
    </a>
    <a href="#" class="social-icon">
      <i class="fab fa-linkedin-in"></i>
    </a>
  </div>
</form>
<form method="post" action="Register" class="sign-up-form">
  <h2 class="title">Register</h2>
  <div class="input-field">
    <i class="fas fa-user"></i>
    <input type="text" placeholder="FirstName" name="firstname" required />
  </div>
  <div class="input-field">
    <i class="fas fa-user"></i>
    <input type="text" placeholder="LastName" name="lastname" required/>
  </div>
  <div class="input-field">
    <i class="fas fa-envelope"></i>
    <input type="text" placeholder="Email" name="email" required/>
  </div>
  <div class="input-field">
    <i class="fas fa-mobile-alt"></i>
    <input type="text" placeholder="Phone Number" name="phonenumber" required/>
  </div>
  <div class="input-field">
    <i class="fas fa-calendar-alt"></i>
    <input type="text" id="dobpicker" placeholder="Date Of Birth" name="dob"
required onfocus="focusFunction()" onblur="(this.type='placeholder')">
    <!--<input type="text" placeholder="Date Of Birth" name="dob" required/> -->
  </div>
  <div class="input-field">
    <i class="fas fa-address-card"></i>
    <input type="text" placeholder="Aadhaar Number" name="adnumber" required/>
  </div>
  <div class="input-field">
    <i class="fas fa-address-card"></i>
    <input type="text" placeholder="Driver License Number" name="dlnumber"
required/>
  </div>
  <div class="input-field">
    <i class="fas fa-map-marker-alt"></i>
    <input type="text" placeholder="Street" name="street" required/>
  </div>
  <div class="input-field">

```

```

        <i class="fas fa-map-marker-alt"></i>
        <input type="text" placeholder="City" name="city" required/>
    </div>
    <div class="input-field">
        <i class="fas fa-map-marker-alt"></i>
        <input type="text" placeholder="State" name="state" required/>
    </div>
    <div class="input-field">
        <i class="fas fa-map"></i>
        <input type="text" placeholder="Zipcode" name="zipcode" required/>
    </div>

    <input type="submit" class="btn" value="Sign up" />
    <p class="social-text">Or Sign up with social platforms</p>
    <div class="social-media">
        <a href="#" class="social-icon">
            <i class="fab fa-facebook-f"></i>
        </a>
        <a href="#" class="social-icon">
            <i class="fab fa-twitter"></i>
        </a>
        <a href="#" class="social-icon">
            <i class="fab fa-google"></i>
        </a>
        <a href="#" class="social-icon">
            <i class="fab fa-linkedin-in"></i>
        </a>
    </div>
</form>
</div>
</div>

<div class="panels-container">
    <div class="panel left-panel">
        <div class="content">
            <h3>New here ?</h3>
            <p>
                Then <br> Hurry! Just SignUp.
            </p>
            <button class="btn transparent" id="sign-up-btn">
                Register
            </button>
        </div>
        
    </div>
    <div class="panel right-panel">
        <div class="content">
            <h3>One of us ?</h3>
            <p>
                Then start Booking And Enjoy Your Journey.
            </p>
        </div>
    </div>
</div>

```

```

    </p>
    <button class="btn transparent" id="sign-in-btn">
      Login
    </button>
  </div>
  
</div>
</div>
</div>
<script src="app.js"></script>
<script>
  function focusFunction() {
    document.getElementById("dobpicker").type="date";
    document.getElementById("dobpicker").min = "1940-01-01";
    document.getElementById("dobpicker").max = "2002-12-31";
  }
</script>
</body>
</html>

```

Caravailability.html

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <link rel="stylesheet" href="css/ca.css">
    <script
      src="https://kit.fontawesome.com/64d58efce2.js"
      crossorigin="anonymous"
    ></script>
    <title>CRS</title>
  </head>
  <body>
    <aside>
      <figure>
        <div id="avatar"></div>
        <figcaption>Let's Go</figcaption>
      </figure>
      
      <nav>
        <ul>
          <li><a href="index.html">Home</a></li>

          <li><a href="caravailability.html">Cars</a></li>
          <li><a href="booking.html">Book A Car</a></li>
          <li><a href="#">About Us</a></li>

```

```
</ul>
</nav>
</aside>
<main>
  <div class="container">
    <div class="carspics">
      
      <span>Carname</span>
    </div>
    <div class="carspics">
      
      <span>Carname</span>
    </div>
    <div class="carspics">
      
      <span>Carname</span>
    </div>
    <div class="carspics">
      
      <span>Carname</span>
    </div>
    <div class="carspics">
      
      <span>Carname</span>
    </div>
    <div class="carspics">
      
      <span>Carname</span>
    </div>
    <div class="carspics">
      
      <span>Carname</span>
    </div>
  </div>
</main>
<script>

(function() {
  var menu = document.querySelector('ul'),
      menulink = document.querySelector('img');

  menulink.addEventListener('click', function(e) {
    menu.classList.toggle('active');
    e.preventDefault();
  });
})();

</script>
</body>
</html>
```


5.1.2 JSP Code

Prfl_Car_Avail_Date.jsp

```
<% @page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>JSP Page</title>
    <link rel="stylesheet" href="css/ca_1.css" />
    <script src="https://kit.fontawesome.com/64d58efce2.js"
      crossorigin="anonymous"></script>
  </head>
  <body>
    <aside>
      <figure>
        <div id="avatar"></div>
        <a href="/Profile">
          <button style="background:none;border:none;font-size: 1.3em;font-weight: bold;
margin-left: 15%;">
            ${FirstName} ${LastName}</button></a>
        </figure>
        
        <nav>
          <ul>
            <li><a href="/Prfl_Data">Profile</a></li>
            <li><a href="/Prfl_CarAvailDate">Book A Car</a></li>
            <li><a href="/Prfl_Booking_Hist">Booking History</a></li>
            <li><a href="/Logout">Logout</a></li>
          </ul>
        </nav>
      </aside>
      <main>
        <form method="post" action="Prfl_CarAvail" class="date-form">
          <div class="headline">
            <h1 class="title1"> START BOOKING YOUR RIDE </h1>
          </div>
          <div class="container">
            <div>
              
            </div>
            <div class="input-field" style="top:30px; right:320px; bottom:150px;">
              <i class="fas fa-calendar-alt"></i>
              <input type="text" id="fromdate" placeholder="From date" name="fromdate"
required onfocus="focusFunction()" onblur="(this.type='placeholder')">
            </div>
          </div>
        </form>
      </main>
    </body>
  </html>
```

```

    <div class="input-field" style="top:140px; right:605px; bottom:150px;">
    <i class="fas fa-calendar-alt"></i>
    <input type="text" id="todate" placeholder="To date" name="todate" required
onfocus="focusFunction()" onblur="(this.type='placeholder')">
    </div>
    <input style="position:absolute; height:55px; width:165px; top:410px; left:360px;
bottom:150px;" type="submit" class="btn" value="Submit"/>
    </div>
</form>
</main>
<script>
    (function() {
        var menu = document.querySelector('ul'), menulink =
document.querySelector('img');
        menulink.addEventListener('click', function(e) {
            menu.classList.toggle('active');
            e.preventDefault();});
    })();
</script>
<script>
    function focusFunction() {
        document.getElementById("fromdate").type="date";
        document.getElementById("fromdate").min = "2021-03-04";
        document.getElementById("fromdate").max = "2021-03-30";

        document.getElementById("todate").type="date";
        document.getElementById("todate").min = "2021-03-04";
        document.getElementById("todate").max = "2021-03-30";
    }
</script>
</body>
</html>

```

Prfl_Bill_Book_Id.jsp

```

<% @page contentType="text/html" pageEncoding="UTF-8"% >
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <link rel="stylesheet" href="css/car_data.css">
    <script src="https://kit.fontawesome.com/64d58efce2.js"
crossorigin="anonymous"></script>
    <title>CRS</title>
</head>
<body>
    <aside>
    <figure>

```

```

<div id="avatar"></div>
<a href="/Profile">
  <button style="background:none;border:none;font-size: 1.3em;font-weight: bold;
margin-left: 15%;">
    ${FirstName} ${LastName}</button></a>
</figure>

<nav>
<ul>
  <li><a href="/Prfl_Data">Profile</a></li>
  <li><a href="/Prfl_CarAvailDate">Book A Car</a></li>

<li><a href="/Prfl_Booking_Hist">Booking History</a></li>
  <li><a href="/Logout">Logout</a></li>
</ul>
</nav>
</aside>
<main>
  <a href="#">
    
  </a>
  <p></br></br></br></br></br></br></br></br></br></br></br></br></br></p>
  <h1>You are all set to vroom now ${FirstName} !!!</h1>
  <h3>Booking Confirmed</h3>
  <p>BookingID : ${bookid}</p>
  <p>BillID : ${billid}</p>
</main>
<script>
  (function() {
    var menu = document.querySelector('ul'), menulink =
document.querySelector('img');
    menulink.addEventListener('click', function(e) {
      menu.classList.toggle('active');
      e.preventDefault();});
  })();
</script>
</body>
</html>

```

App.js

```

const sign_in_btn = document.querySelector("#sign-in-btn");
const sign_up_btn = document.querySelector("#sign-up-btn");
const container = document.querySelector(".container");
sign_up_btn.addEventListener("click", () => {
  container.classList.add("sign-up-mode");});
sign_in_btn.addEventListener("click", () => {
  container.classList.remove("sign-up-mode");});

```

5.1.3 CSS CODE

Ca.css

```
@import url('https://fonts.googleapis.com/css?family=Montserrat:400,700');
@import
url("https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;500;600;700;800
&display=swap");

body, html {
    height: calc(100% - 2em);
    margin: 0;
    font-family: 'Montserrat';
}

body {
    /* Permalink - use to edit and share this gradient: http://colorzilla.com/gradient-
editor/#4a00fb+0,e600ff+100 */
    background: #cc2b5e; /* Old browsers */
    background: -moz-linear-gradient(45deg,#cc2b5e 0%, #753a88 100%); /* FF3.6-15 */
    background: -webkit-linear-gradient(45deg, #cc2b5e 0%,#753a88 100%); /* Chrome10-
25,Safari5.1-6 */
    background: linear-gradient(45deg, #cc2b5e 0%,#753a88 100%); /* W3C, IE10+, FF16+,
Chrome26+, Opera12+, Safari7+ */
    filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#cc2b5e',
endColorstr='#753a88',GradientType=1 ); /* IE6-9 fallback on horizontal gradient */
    background-repeat: no-repeat;
    background-attachment: fixed;
}

aside {
    color: #fff;
    display: grid;
    grid-template-columns: auto 40px;
    padding: 2em;
}

.img {
    width: 40px;
    cursor: pointer;
}

.figure {
    margin: 0;
}

figcaption {
    font-size: 1.3em;
    font-weight: bold;
```

```
}

nav {
  background: #fff;
}

.active {
  position: absolute;
  display: block;
  background: #fff;
  width: 50%;
  left: 0;
  top: 0;
  height: 100%;
}

ul {
  display: none;
  list-style-type: none;
  margin: 0;
  padding: 0;
}

ul li a {
  text-decoration: none;
  text-transform: uppercase;
  font-size: .8em;
  display: block;
  padding: 1.5em 3em;
  background-color: rgba(255,255,255,.35);
  color: #000;
}

main {
  background: #fff;
  padding: 1em 1.4em;
}

h1 {
  margin: 0;
}

p {
  font-size: 1.1em;
  line-height: 1.5em;
}

.cta {
  display: block;
```

```
color: #fff;
background: #0E7FFF;
text-decoration: none;
padding: 8px 50px;
font-size: 1.2em;
text-align: center;
font-weight: bold;
border-radius: 4px;
margin: 2em 0;
}

.container {
padding: 3em;
display: grid;
grid-gap: 2em;
grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
}

.carspics {
padding: 2em;
border-radius: .3em;
box-shadow: 10px 10px 30px rgba(0,0,0,0.1);
}

#crp {
margin-top: 0;
height: 90%;
width: 95%;
}

span {
font-weight: bold;
position: relative;
margin-left: 15px;
}

/* :before {
content: "";
position: absolute;
height: 1px;
width: 10px;
border-bottom: 1px solid black;
top: 10px;
left: -15px;
}*/

@media (min-width: 550px) {
/*
```

```
.span-2 {
  grid-column: auto / span 2;
}
*/
}

@media only screen and (min-width: 768px) {

  body {
    display: grid;
    grid-template-columns: 20% auto;
    padding: 1em;
    overflow: hidden;
  }

  .img {
    display: none;
  }

  aside {
    background: #DCE1F6;
    grid-template-columns: auto;
    grid-template-rows: 20% auto;
    padding: 0;
  }

  main {
    padding: 2.5em;
    padding-bottom: 3em;
    background: white url('../images/bg.svg') no-repeat bottom right;
    background-size: 65%;
  }

  #avatar {
    border-radius: 50%;
    background: #fff;
    width: 80px;
    height: 80px;
    margin: 2em auto 1em auto;
  }

  figcaption {
    text-align: center;
    color: #000;
  }

  ul {
    display: block;
    margin-top: 5em;
    position: relative !important;
  }
}
```

```
    background: none !important;
    width: 100% !important;
}

ul li a:hover {
    background-color: #fff;
}

nav {
    background: none;
}

.cta {
    display: inline-block;
    font-size: 1.3em;
    padding: .8em 3em;
}

h1 {
    font-size: 3em;
    margin-top: 1em;
}

p {
    font-size: 1em;
}

.logo{
    max-width: 70%;
    max-height: 70%;
    display: block;
    margin-bottom: 2em;
    margin-left: 15%;
}

logosec{
    height:250px;
    width:500px;
}

.logo2{
    height: 100%;
    width: 100%;
}

}

@media (min-width: 550px) {
    /*
    .span-2 {
        grid-column: auto / span 2;
    }
    */
}
```


User_main.css

```
@import url('https://fonts.googleapis.com/css?family=Montserrat:400,700');
@import
url("https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;500;600;700;800
&display=swap");

body, html {
    height: calc(100% - 2em);
    margin: 0;
    font-family: 'Montserrat';
}

body {
    /* Permalink - use to edit and share this gradient: http://colorzilla.com/gradient-
editor/#4a00fb+0,e600ff+100 */
    background: #cc2b5e; /* Old browsers */
    background: -moz-linear-gradient(45deg,#cc2b5e 0%, #753a88 100%); /* FF3.6-15 */
    background: -webkit-linear-gradient(45deg, #cc2b5e 0%,#753a88 100%); /* Chrome10-
25,Safari5.1-6 */
    background: linear-gradient(45deg, #cc2b5e 0%,#753a88 100%); /* W3C, IE10+, FF16+,
Chrome26+, Opera12+, Safari7+ */
    filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#cc2b5e',
endColorstr='#753a88',GradientType=1 ); /* IE6-9 fallback on horizontal gradient */
    background-repeat: no-repeat;
    background-attachment: fixed;
}

aside {
    color: #fff;
    display: grid;
    grid-template-columns: auto 40px;
    padding: 2em;
}

.img {
    width: 40px;
    cursor: pointer;
}

.figure {
    margin: 0;
}

figcaption {
    font-size: 1.3em;
    font-weight: bold;
}
```

```
nav {
  background: #fff;
}

.active {
  position: absolute;
  display: block;
  background: #fff;
  width: 50%;
  left: 0;
  top: 0;
  height: 100%;
}

ul {
  display: none;
  list-style-type: none;
  margin: 0;
  padding: 0;
}

ul li a {
  text-decoration: none;
  text-transform: uppercase;
  font-size: .8em;
  display: block;
  padding: 1.5em 3em;
  background-color: rgba(255,255,255,.35);
  color: #000;
}

main {
  background: #fff;
  padding: 1em 1.4em;
}

h1 {
  margin: 0;
}

p {
  font-size: 1.1em;
  line-height: 1.5em;
}

.cta {
  display: block;
  color: #fff;
  background: #0E7FFF;
```

```
text-decoration: none;
padding: 8px 50px;
font-size: 1.2em;
text-align: center;
font-weight: bold;
border-radius: 4px;
margin: 2em 0;
}
@media only screen and (min-width: 768px) {

  body {
    display: grid;
    grid-template-columns: 20% auto;
    padding: 1em;
    overflow: hidden;
  }

  .img {
    display: none;
  }

  aside {
    background: #DCE1F6;
    grid-template-columns: auto;
    grid-template-rows: 20% auto;
    padding: 0;
  }

  main {
    padding: 2.5em;
    padding-bottom: 3em;
    background: white url('../images/bg.svg') no-repeat bottom right;
    background-size: 55%;
  }

  #avatar {
    border-radius: 50%;
    background: #fff;
    width: 80px;
    height: 80px;
    margin: 2em auto 1em auto;
  }

  figcaption {
    text-align: center;
    color: #000;
  }

  ul {
    display: block;
    margin-top: 5em;
    position: relative !important;
```

```
        background: none !important;
        width: 100% !important;
    }
    ul li a:hover {
        background-color: #fff;
    }
    nav {
        background: none;
    }
    .cta {
        display: inline-block;
        font-size: 1.3em;
        padding: .8em 3em;
    }
    h1 {
        font-size: 3em;
        margin-top: 1em;
    }
    p {
        font-size: 1em;
    }
    .logo{
        max-width: 70%;
        max-height: 70%;
        display: block;
        margin-bottom: 2em;
        margin-left: 15%;
    }

    logosec{
        height:250px;
        width:500px;
    }

    .logo2{
        height: 100%;
        width: 100%;
    }
}
```

5.1.4 JAVA CODE

AdminPrflData.java

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

/**
 *
 * @author blraj
 */
@WebServlet(urlPatterns = {"/AdminPrflData"})
public class AdminPrflData extends HttpServlet {

    /**
     * Processes requests for both HTTP GET and POST
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            HttpSession session = request.getSession(false);
            response.setHeader("Cache-Control", "no-cache, no-store, must-revalidate");
            response.setHeader("Pragma", "no-cache");
            response.setHeader("Expires", "0");
            if(session != null)
            {
                String Userid = (String) session.getAttribute("userid");
                String FirstName = (String) session.getAttribute("firstname");
                String LastName = (String) session.getAttribute("lastname");
                String Email = (String) session.getAttribute("email");
                String Phnum = (String) session.getAttribute("phnum");
                String DOB = (String) session.getAttribute("dob");
                String DLnum = (String) session.getAttribute("dlnum");
                String ADRnum = (String) session.getAttribute("adnum");
                String Street = (String) session.getAttribute("street");
                String City = (String) session.getAttribute("city");
            }
        }
    }
}
```

```

String State = (String) session.getAttribute("state");
String Zipcode = (String) session.getAttribute("zip");

out.println("<html lang=\"en\">\n" +
    "    <head>\n" +
    "        <meta charset=\"UTF-8\">\n" +
    "        <meta name=\"viewport\" content=\"width=device-width, initial-
scale=1.0\">\n" +
    "        <meta http-equiv=\"X-UA-Compatible\" content=\"ie=edge\">\n" +
    "        <link rel=\"stylesheet\" href=\"css/prfl_data.css\">\n" +
    "        <script\n" +
    "            src=\"https://kit.fontawesome.com/64d58efce2.js\" \n" +
    "            crossorigin=\"anonymous\" \n" +
    "        ></script>\n" +
    "        <title>CRS</title>\n" +
    "    </head>\n" +
    "    <body>\n" +
    "        <aside>\n" +
    "            <figure>\n" +
    "                <div id=\"avatar\"><img class=\"logo2\"
src=\"images/person_logo.png\"></div>\n" +
    "                <a href=\"./AdminProfile\">\n" +
    "                    <button style=\"background:none;border:none;font-size:
1.3em;font-weight: bold;margin-left: 15%;\">
+FirstName+ \" \" + LastName+</button></a>\n" +
    "                </div>\n" +
    "                <img class=\"img\" src=\"images/menu.svg\" alt=\"menu\">\n" +
    "            <nav>\n" +
    "                <ul>\n" +
    "                    <li><a href=\"./AdminPrflData\">Profile</a></li>\n" +
    "                    \n" +
    "                    <li><a href=\"#\">Upsert Cars</a></li>\n" +
    "                    <li><a href=\"#\">Upsert Bookings</a></li>\n" +
    "                    <li><a href=\"#\">Booking History</a></li>\n" +
    "                    <li><a href=\"./Logout\">Logout</a></li>\n" +
    "                </ul>\n" +
    "            </nav>\n" +
    "        </aside>\n" +
    "        <main> \n" +
    "    </div>"+
    "<img src=\"images/prfldata.svg\" style=\"position:absolute; top:50px;
left:665px; bottom: 150px\">"+
    "<h3>Admin Details</h3>"+
    "<p>UserID      : "+ Userid +"</p>"+
    "<p>Name        : "+ FirstName + \" \" + LastName +"</p>"+
    "<p>Email       : "+ Email +"</p>"+
    "<p>Phone Number : "+ Phnum +"</p>"+
    "<p>Date Of Birth : "+ DOB +"</p>"+
    "<p>Driver's License : "+ DLnum +"</p>"+

```

```

        "<p>Aaadhaar Number : "+ ADRnum + "</p>" +
        "<p>Street      : "+ Street + "</p>" +
        "<p>City        : "+ City + "</p>" +
        "<p>State        : "+ State + "</p>" +
        "<p>Zipcode      : "+ Zipcode + "</p>" +
        "</div>" +
        "    </main>\n" +
        "    <script>\n" +
        "        \n" +
        "        (function() {\n" +
        "            var menu = document.querySelector('ul'),\n" +
        "                menulink = document.querySelector('img');\n" +
        "            \n" +
        "            menulink.addEventListener('click', function(e) {\n" +
        "                menu.classList.toggle('active');\n" +
        "                e.preventDefault();\n" +
        "            });\n" +
        "        })();\n" +
        "        \n" +
        "    </script>\n" +
        "    </body>\n" +
        "</html>");
    }
    else
    {
        response.sendRedirect("signin-signup.html");
        response.setHeader("Cache-Control", "no-cache, no-store, must-revalidate");
        response.setHeader("Pragma", "no-cache");
        response.setHeader("Expires", "0");
    }
}

// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on
the left to edit the code.">
/**
 * Handles the HTTP <code>GET</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

/**

```

```

    * Handles the HTTP <code>POST</code> method.
    *
    * @param request servlet request
    * @param response servlet response
    * @throws ServletException if a servlet-specific error occurs
    * @throws IOException if an I/O error occurs
    */
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Returns a short description of the servlet.
     *
     * @return a String containing servlet description
     */
    @Override
    public String getServletInfo() {
        return "Short description";
    } // </editor-fold>
}

```

AdminProfile.java

```

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

/**
 *
 * @author blraj
 */
@WebServlet(urlPatterns = {"/AdminProfile"})
public class AdminProfile extends HttpServlet {

    /**
     * Processes requests for both HTTP <code>GET</code> and <code>POST</code>
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs

```



```

* @throws IOException if an I/O error occurs
*/
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    try (PrintWriter out = response.getWriter()) {
        HttpSession session = request.getSession(false);
        response.setHeader("Cache-Control", "no-cache, no-store, must-revalidate");
        response.setHeader("Pragma", "no-cache");
        response.setHeader("Expires", "0");
        if(session != null)
        {
            String FirstName = (String) session.getAttribute("firstname");
            String LastName = (String) session.getAttribute("lastname");

            out.println("<html lang=\"en\">\n" +
                "    <head>\n" +
                "        <meta charset=\"UTF-8\">\n" +
                "        <meta name=\"viewport\" content=\"width=device-width, initial-
scale=1.0\">\n" +
                "        <meta http-equiv=\"X-UA-Compatible\" content=\"ie=edge\">\n" +
                "        <link rel=\"stylesheet\" href=\"css/user_main.css\">\n" +
                "        <script\n" +
                "            src=\"https://kit.fontawesome.com/64d58efce2.js\"\n" +
                "            crossorigin=\"anonymous\"\n" +
                "        ></script>\n" +
                "        <title>CRS</title>\n" +
                "    </head>\n" +
                "    <body>\n" +
                "        <aside>\n" +
                "            <figure>\n" +
                "                <div id=\"avatar\"><img class=\"logo2\"
src=\"images/person_logo.png\"></div>\n" +
                "                <a href=\"./AdminProfile\">\n" +
                "                    <button style=\"background:none;border:none;font-size:
1.3em;font-weight: bold;margin-left: 15%;\">
+LastName+FirstName+</button></a>\n" +
                "            </figure>\n" +
                "            <img class=\"img\" src=\"images/menu.svg\" alt=\"menu\">\n" +
                "            <nav>\n" +
                "                <ul>\n" +
                "                    <li><a href=\"./AdminPrflData\">Profile</a></li>\n" +
                "                    \n" +
                "                    <li><a href=\"#\">Upsert Cars</a></li>\n" +
                "                    <li><a href=\"#\">Upsert Bookings</a></li>\n" +
                "                    <li><a href=\"#\">Booking History</a></li>\n" +
                "                    <li><a href=\"./Logout\">Logout</a></li>\n" +
                "                </ul>\n" +
                "            </nav>\n" +
                "        </aside>\n" +

```

```

        "    <main> \n" +
        "    <div class=\"logosec\"><img class=\"logo\"
src=\"images/Welcome.svg\" style=\"height:380px\" alt=\"logo\">\n" +
        "    \n" +
        "    </div>\n" +
        "    <h1>Hello, "+FirstName+" "+LastName+"</h1> \n" +
        "    <p> Being happy is being productive!</p>\n" +
        "\n" +
        "    </main>\n" +
        "    <script>\n" +
        "    \n" +
        "    (function() {\n" +
        "        var menu = document.querySelector('ul'),\n" +
        "        menulink = document.querySelector('img');\n" +
        "        \n" +
        "        menulink.addEventListener('click', function(e) {\n" +
        "            menu.classList.toggle('active');\n" +
        "            e.preventDefault();\n" +
        "        });\n" +
        "    })();\n" +
        "    \n" +
        "    </script>\n" +
        "    </body>\n" +
        "</html>");
    }
    else
    {
        response.sendRedirect("signin-signup.html");
        response.setHeader("Cache-Control", "no-cache, no-store, must-revalidate");
        response.setHeader("Pragma", "no-cache");
        response.setHeader("Expires", "0");

    }
}
}
}

```

// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">

/**

* Handles the HTTP <code>GET</code> method.

*

* @param request servlet request

* @param response servlet response

* @throws ServletException if a servlet-specific error occurs

* @throws IOException if an I/O error occurs

*/

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response)

```

        throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Handles the HTTP <code>POST</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Returns a short description of the servlet.
 *
 * @return a String containing servlet description
 */
@Override
public String getServletInfo() {
    return "Short description";
} // </editor-fold>

}

```

Gnrt_Pdf.java

```

import java.io.IOException;
import java.io.PrintWriter;
import java.io.*;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import com.itextpdf.text.*;
import com.itextpdf.text.Font.FontFamily;
import com.itextpdf.text.pdf.PdfWriter;
import com.itextpdf.text.Image;
import com.itextpdf.text.pdf.PdfPCell;
import com.itextpdf.text.pdf.PdfPTable;
import java.awt.Color;
import java.util.logging.Level;

```

```
import java.util.logging.Logger;

/**
 *
 * @author blraj
 */
@WebServlet(urlPatterns = {"/Gnrt_PDF"})
public class Gnrt_PDF extends HttpServlet {

    /**
     * Processes requests for both HTTP <code>GET</code> and <code>POST</code>
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            HttpSession session = request.getSession(false);
            String uid= (String) session.getAttribute("userid");

            String FirstName = (String) session.getAttribute("firstname");
            String LastName = (String) session.getAttribute("lastname");

            Vroom vr = new Vroom();
            int userid = Integer.parseInt(uid);
            vr.pop_usr_bkng(userid);
            int no_of_bkngs = vr.user_bookings.get(userid).size();

            if(session == null)
            {
                RequestDispatcher rd = request.getRequestDispatcher("Login");
                rd.forward(request,response);
            }
            else
            {
                try
                {
                    Document bk_hist = new Document();
                    PdfWriter.getInstance(bk_hist, new
                    FileOutputStream("D://Booking_History.pdf"));

                    bk_hist.open();
```

```

        Font blue = new Font(FontFamily.COURIER, 26, Font.BOLD,
BaseColor.BLUE);
        Chunk heading = new Chunk("VROOM CAR SHARE - BOOKING HISTORY",
blue);
        Paragraph p = new Paragraph(30);
        p.add(heading);
        //p.setAlignment(Element.ALIGN_CENTER);
        bk_hist.add(p);

        Font nm = new Font(FontFamily.COURIER, 20, Font.BOLD,
BaseColor.BLACK);
        String nmhead = "\nName: " + FirstName + LastName + "  User-ID: " + uid;
        Chunk nmid = new Chunk(nmhead, nm);
        Paragraph nmidp = new Paragraph(30);
        nmidp.add(nmid);
        //p.setAlignment(Element.ALIGN_CENTER);
        bk_hist.add(nmidp);

        float img_y=500f;
        float yRed = 200f;

        if(no_of_bkngs>0)
        {
            for(int i=0; i < no_of_bkngs; i++)
            {
                String carid =
vr.car_details.get(Integer.parseInt(vr.user_bookings.get(userid).get(i).get(2))).get(0);

                String filename =
"C:\\Users\\blraj\\Documents\\NetBeansProjects\\VROOM\\web\\images\\"+carid+".jpg";
                String text = "\nBooking : " + (i+1) + "\n"
                    + "Booking ID : " + vr.user_bookings.get(userid).get(i).get(0) +
"\n"
                    + "Billing ID : " + vr.user_bookings.get(userid).get(i).get(8) + "\n"
                    + "Car Model : "+
vr.car_details.get(Integer.parseInt(vr.user_bookings.get(userid).get(i).get(2))).get(4) + "\n"
                    + "Total Amount : "+ vr.user_bookings.get(userid).get(i).get(6) +
"\n"
                    + "Pick Up Date : "+ vr.user_bookings.get(userid).get(i).get(3) +
"\n"
                    + "Return Date : "+ vr.user_bookings.get(userid).get(i).get(4) +
"\n"
                    + "Location : "+
vr.carloc.get(Integer.parseInt(vr.user_bookings.get(userid).get(i).get(5))).get(1);

                PdfPTable table = new PdfPTable(2);
                table.setWidthPercentage(100);
                table.setWidths(new int[] {3, 2});
                table.addCell(createTextCell(text));
                table.addCell(createImageCell(filename));

```

```

        bk_hist.add(table);
    }
}

bk_hist.close();

String altmsg = "Pdf Generated successfully!";
response.setContentType("text/html");
out.println("<script type=\"text/javascript\">");
out.println("alert(\"" + altmsg + "\");");
out.println("</script>");
RequestDispatcher rd = request.getRequestDispatcher("./Prfl_Booking_Hist");
rd.forward(request,response);

} catch (Exception ex) {
    Logger.getLogger(Gnrt_PDF.class.getName()).log(Level.SEVERE, null, ex);
}
}
}
}

public static PdfPCell createImageCell(String path) throws DocumentException,
IOException {
    Image img = Image.getInstance(path);
    //img.scalePercent(40);
    img.setAlignment(Image.MIDDLE);
    PdfPCell cell = new PdfPCell();
    cell.addElement(new Paragraph(" "));
    cell.addElement(new Paragraph(" "));
    cell.addElement(img);
    cell.setBorder(Rectangle.NO_BORDER);

    cell.setHorizontalAlignment(Element.ALIGN_CENTER); //cell.setVerticalAlignment(Elleme
nt.ALIGN_CENTER);
    return cell;
}

public static PdfPCell createTextCell(String text) throws DocumentException,
IOException {
    PdfPCell cell = new PdfPCell();
    Paragraph p = new Paragraph(23);
    Font blk = new Font(FontFamily.HELVETICA, 18, Font.NORMAL, BaseColor.BLACK);
    Chunk ech_bkn = new Chunk(text, blk);
    p.add(ech_bkn);
    cell.addElement(p);
    //cell.setVerticalAlignment(Element.ALIGN_BOTTOM);
    cell.setBorder(Rectangle.NO_BORDER);
    return cell;
}
}

```

```
// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on
the left to edit the code.">
```

```
/**
 * Handles the HTTP <code>GET</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Handles the HTTP <code>POST</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Returns a short description of the servlet.
 *
 * @return a String containing servlet description
 */
@Override
public String getServletInfo() {
    return "Short description";
} // </editor-fold>

}
```

Login.java

```
import java.io.IOException;
import java.io.PrintWriter;
import java.io.*;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
```

```
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import com.itextpdf.text.*;
import com.itextpdf.text.Font.FontFamily;
import com.itextpdf.text.pdf.PdfWriter;
import com.itextpdf.text.Image;
import com.itextpdf.text.pdf.PdfPCell;
import com.itextpdf.text.pdf.PdfPTable;
import java.awt.Color;
import java.util.logging.Level;
import java.util.logging.Logger;

/**
 *
 * @author blraj
 */
@WebServlet(urlPatterns = {"/Gnrt_PDF"})
public class Gnrt_PDF extends HttpServlet {

    /**
     * Processes requests for both HTTP <code>GET</code> and <code>POST</code>
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            HttpSession session = request.getSession(false);
            String uid= (String) session.getAttribute("userid");

            String FirstName = (String) session.getAttribute("firstname");
            String LastName = (String) session.getAttribute("lastname");

            Vroom vr = new Vroom();
            int userid = Integer.parseInt(uid);
            vr.pop_usr_bkng(userid);
            int no_of_bkngs = vr.user_bookings.get(userid).size();

            if(session == null)
            {
                RequestDispatcher rd = request.getRequestDispatcher("Login");
                rd.forward(request,response);
            }
        }
    }
}
```

```

else
{
    try
    {
        Document bk_hist = new Document();
        PdfWriter.getInstance(bk_hist, new
FileOutputStream("D://Booking_History.pdf"));

        bk_hist.open();

        Font blue = new Font(FontFamily.COURIER, 26, Font.BOLD,
BaseColor.BLUE);
        Chunk heading = new Chunk("VROOM CAR SHARE - BOOKING HISTORY",
blue);
        Paragraph p = new Paragraph(30);
        p.add(heading);
        //p.setAlignment(Element.ALIGN_CENTER);
        bk_hist.add(p);

        Font nm = new Font(FontFamily.COURIER, 20, Font.BOLD,
BaseColor.BLACK);
        String nmhead = "\nName: " + FirstName + LastName + "  User-ID: " + uid;
        Chunk nmid = new Chunk(nmhead, nm);
        Paragraph nmidp = new Paragraph(30);
        nmidp.add(nmid);
        //p.setAlignment(Element.ALIGN_CENTER);
        bk_hist.add(nmidp);

        float img_y=500f;
        float yRed = 200f;

        if(no_of_bkngs>0)
        {
            for(int i=0; i < no_of_bkngs; i++)
            {
                String carid =
vr.car_details.get(Integer.parseInt(vr.user_bookings.get(userid).get(i).get(2))).get(0);

                String filename =
"C:\\Users\\blraj\\Documents\\NetBeansProjects\\VROOM\\web\\images\\"+carid+".jpg";
                String text = "\nBooking : " + (i+1) + "\n"
                    + "Booking ID : " + vr.user_bookings.get(userid).get(i).get(0) +
"\n"
                    + "Billing ID : " + vr.user_bookings.get(userid).get(i).get(8) + "\n"
                    + "Car Model : "+
vr.car_details.get(Integer.parseInt(vr.user_bookings.get(userid).get(i).get(2))).get(4) + "\n"
                    + "Total Amount : "+ vr.user_bookings.get(userid).get(i).get(6) +
"\n"
                    + "Pick Up Date : "+ vr.user_bookings.get(userid).get(i).get(3) +
"\n"

```

```

        +"Return Date : "+ vr.user_bookings.get(userid).get(i).get(4) +
"\n"
        +"Location : "+
vr.carloc.get(Integer.parseInt(vr.user_bookings.get(userid).get(i).get(5))).get(1);

        PdfPTable table = new PdfPTable(2);
        table.setWidthPercentage(100);
        table.setWidths(new int[]{3, 2});
        table.addCell(createTextCell(text));
        table.addCell(createImageCell(filename));
        bk_hist.add(table);
    }
}

bk_hist.close();

String altmsg = "Pdf Generated successfully!";
response.setContentType("text/html");
out.println("<script type=\"text/javascript\">");
out.println("alert(\"" + altmsg + "\");");
out.println("</script>");
RequestDispatcher rd = request.getRequestDispatcher("./Prfl_Booking_Hist");
rd.forward(request,response);

    } catch (Exception ex) {
        Logger.getLogger(Gnrt_PDF.class.getName()).log(Level.SEVERE, null, ex);
    }
}

}
}

```

```

    public static PdfPCell createImageCell(String path) throws DocumentException,
IOException {
        Image img = Image.getInstance(path);
        //img.scalePercent(40);
        img.setAlignment(Image.MIDDLE);
        PdfPCell cell = new PdfPCell();
        cell.addElement(new Paragraph(" "));
        cell.addElement(new Paragraph(" "));
        cell.addElement(img);
        cell.setBorder(Rectangle.NO_BORDER);

        cell.setHorizontalAlignment(Element.ALIGN_CENTER);//cell.setVerticalAlignment(Elleme
nt.ALIGN_CENTER);
        return cell;
    }

```

```

    public static PdfPCell createTextCell(String text) throws DocumentException,
IOException {

```

```
PdfPCell cell = new PdfPCell();
Paragraph p = new Paragraph(23);
Font blk = new Font(FontFamily.HELVETICA, 18, Font.NORMAL, BaseColor.BLACK);
Chunk ech_bkn = new Chunk(text, blk);
p.add(ech_bkn);
cell.addElement(p);
//cell.setVerticalAlignment(Element.ALIGN_BOTTOM);
cell.setBorder(Rectangle.NO_BORDER);
return cell;
}

// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on
the left to edit the code.">
/**
 * Handles the HTTP <code>GET</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Handles the HTTP <code>POST</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Returns a short description of the servlet.
 *
 * @return a String containing servlet description
 */
@Override
public String getServletInfo() {
    return "Short description";
}
</editor-fold>
```

```
}
```

Logout.java

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

/**
 *
 * @author blraj
 */
@WebServlet(urlPatterns = {"/Logout"})
public class Logout extends HttpServlet {

    /**
     * Processes requests for both HTTP GET and POST
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            HttpSession session = request.getSession(false);
            session.invalidate();
            String altmsg = "You have been logged out successfully!";
            response.setContentType("text/html");
            out.println("<script type=\"text/javascript\">");
            out.println("alert(\"" + altmsg + "\");");
            out.println("window.location.href = \"signin-signup.html\";");
            out.println("</script>");
        }
    }

    // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on
    the left to edit the code.">
    /**
     * Handles the HTTP GET method.
     *
     * @param request servlet request
     * @param response servlet response

```

```

    * @throws ServletException if a servlet-specific error occurs
    * @throws IOException if an I/O error occurs
    */
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Handles the HTTP <code>POST</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Returns a short description of the servlet.
     *
     * @return a String containing servlet description
     */
    @Override
    public String getServletInfo() {
        return "Short description";
    } // </editor-fold>
}

```

Prfl_Bk_Confirm_Pay.java

```

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;

```

```
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

/**
 *
 * @author blraj
 */
@WebServlet(urlPatterns = {"/Prfl_Payment"})
public class Prfl_Bill_ID extends HttpServlet {

    /**
     * Processes requests for both HTTP <code>GET</code> and <code>POST</code>
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            HttpSession session = request.getSession(false);
            String cid= (String) session.getAttribute("cid");
            String uid= (String) session.getAttribute("userid");
            Vroom vr = new Vroom();
            int carid = Integer.parseInt(cid);
            int userid = Integer.parseInt(uid);
            int locid = Integer.parseInt(vr.caravail.get(carid).get(1))-2000;

            String FirstName = (String) session.getAttribute("firstname");
            String LastName = (String) session.getAttribute("lastname");

            String fromdate = (String) session.getAttribute("fromdate");
            String todate = (String) session.getAttribute("todate");

            int todt = Integer.parseInt(todate.substring(8,10));
            int frmdt = Integer.parseInt(fromdate.substring(8,10));
            int total =(1+todt)-frmdt;
            total = total * Integer.parseInt(vr.caravail.get(carid).get(7));

            if(session != null)
            {

                Connection con=null;
```

```
Statement stm=null;
ResultSet rs=null;
int bookid = vr.bookID;
int billid = vr.billID;

try {
    Class.forName("com.mysql.jdbc.Driver");
    con=DriverManager.getConnection("jdbc:mysql://localhost/crs","root","");
    PreparedStatement bill;
    PreparedStatement book;
    book=con.prepareStatement("insert into booking values(?, ?, ?, ?, ?, ?, ?, ?, ?)");
    book.setInt(1, bookid);
    book.setInt(2, userid);
    book.setInt(3, carid);
    book.setString(4, fromdate);
    book.setString(5, todate);
    book.setInt(6, Integer.parseInt(vr.caravail.get(carid).get(1)));
    book.setInt(7, Integer.parseInt(vr.caravail.get(carid).get(1)));
    book.setInt(8, total);
    book.setString(9, "Approved");
    book.executeUpdate();

    bill=con.prepareStatement("insert into billing values(?, ?, ?, ?, ?)");
    bill.setInt(1, billid);
    bill.setInt(2, bookid);
    bill.setString(3, fromdate);
    bill.setString(4, "Approved");
    bill.setInt(5, total);
    bill.executeUpdate();

} catch (ClassNotFoundException ex) {out.println("Invalid1");
    Logger.getLogger(Prfl_Bill_ID.class.getName()).log(Level.SEVERE, null, ex);
} catch (SQLException ex) {
    Logger.getLogger(Prfl_Bill_ID.class.getName()).log(Level.SEVERE, null, ex);
}

request.setAttribute("LastName", LastName);
request.setAttribute("FirstName", FirstName);

request.setAttribute("Amount", Integer.toString(total*100));

request.setAttribute("bookid", bookid);
request.setAttribute("billid", billid);

request.getRequestDispatcher("/Prfl_Bill_Book_ID.jsp").forward(request, response);
}
}
```

// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">

```
/**
 * Handles the HTTP <code>GET</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Handles the HTTP <code>POST</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Returns a short description of the servlet.
 *
 * @return a String containing servlet description
 */
@Override
public String getServletInfo() {
    return "Short description";
} // </editor-fold>
}
```

Prfl_CarAvail.java

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
```

```

import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

/**
 *
 * @author blraj
 */
@WebServlet(urlPatterns = {"/Prfl_CarAvail"})
public class Prfl_CarAvail extends HttpServlet {

    /**
     * Processes requests for both HTTP <code>GET</code> and <code>POST</code>
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {

            HttpSession session = request.getSession(false);
            Vroom vr = new Vroom();
            if(session != null)
            {
                String FirstName = (String) session.getAttribute("firstname");
                String LastName = (String) session.getAttribute("lastname");

                session.setAttribute("todate", request.getParameter("todate"));
                session.setAttribute("fromdate", request.getParameter("fromdate"));

                vr.pop_availcar(request.getParameter("fromdate"), request.getParameter("todate"));

                int no_of_avail_cars = vr.caravail.size();

                out.println("<html lang=\"en\">\n" +
                    "    <head>\n" +
                    "        <meta charset=\"UTF-8\">\n" +
                    "        <meta name=\"viewport\" content=\"width=device-width, initial-
scale=1.0\">\n" +
                    "        <meta http-equiv=\"X-UA-Compatible\" content=\"ie=edge\">\n" +
                    "        <link rel=\"stylesheet\" href=\"css/ca_1.css\">\n" +
                    "        <script\n" +
                    "            src=\"https://kit.fontawesome.com/64d58efce2.js\"\n" +
                    "            crossorigin=\"anonymous\"\n" +
                    "        ></script>\n" +
                    "        <title>CRS</title>\n" +

```

```

"    </head>\n" +
"    <body>\n" +
"        <aside>\n" +
"            <figure>\n" +
"                <div id=\"avatar\"><img class=\"logo2\"
src=\"images/person_logo.png\"></div>\n" +
"                <a href=\"./Profile\">\n" +
"                    <button style=\"background:none;border:none;font-size: 1.3em;font-
weight: bold; margin-left: 15%;\">
                        +FirstName+" "+ LastName+</button></a>\n" +
"                </figure>\n" +
"                <img class=\"img\" src=\"images/menu.svg\" alt=\"menu\">\n" +
"                <nav>\n" +
"                    <ul>\n" +
"                        <li><a href=\"./Prfl_Data\">Profile</a></li>\n" +
"                        \n" +
"                        <li><a href=\"./Prfl_CarAvailDate\">Book A Car</a></li>\n" +
"                        <li><a href=\"./Prfl_Booking_Hist\">Booking History</a></li>\n"
+
"                        <li><a href=\"./Logout\">Logout</a></li>\n" +
"                    </ul>\n" +
"                </nav>\n" +
"            </aside>\n" +
"            <main> \n");

out.println("<div class=\"container\">\n");
for(Integer i : vr.caravail.keySet())
{
    out.println("<a href=\"./Prfl_CarData?xxx="+vr.caravail.get(i).get(0)+"\">"
        + "<div class=\"carspics\">\n" +
"        <img src=\"images/\"+vr.caravail.get(i).get(0)+\".jpg\" id=\"crp\">\n" +
"        <span>"+ vr.caravail.get(i).get(4)+"</span>\n" +
"        </div>\n" +
"        </a>");
}
out.println("</div>");

out.println("    </main>\n" +
"    <script>\n" +
"        \n" +
"        (function() {\n" +
"            var menu = document.querySelector('ul'),\n" +
"            menulink = document.querySelector('img');\n" +
"            \n" +
"            menulink.addEventListener('click', function(e) {\n" +
"                menu.classList.toggle('active');\n" +
"                e.preventDefault();\n" +
"            });\n" +
"        })();\n" +
"        \n" +
"    ");

```

```
        "</script>\n" +
        "</body>\n" +
        "</html>");
    }
}

// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on
the left to edit the code.">
/**
 * Handles the HTTP <code>GET</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Handles the HTTP <code>POST</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Returns a short description of the servlet.
 *
 * @return a String containing servlet description
 */
@Override
public String getServletInfo() {
    return "Short description";
} // </editor-fold>
}
```

Profile.java

```

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

/**
 *
 * @author blraj
 */
@WebServlet(urlPatterns = {"/Profile"})
public class Profile extends HttpServlet {

    /**
     * Processes requests for both HTTP GET and POST
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            HttpSession session = request.getSession(false);
            response.setHeader("Cache-Control", "no-cache, no-store, must-revalidate");
            response.setHeader("Pragma", "no-cache");
            response.setHeader("Expires", "0");
            if(session != null)
            {
                String FirstName = (String) session.getAttribute("firstname");
                String LastName = (String) session.getAttribute("lastname");

                out.println("<html lang=\"en\">\n" +
                    "    <head>\n" +
                    "        <meta charset=\"UTF-8\">\n" +
                    "        <meta name=\"viewport\" content=\"width=device-width, initial-
scale=1.0\">\n" +
                    "        <meta http-equiv=\"X-UA-Compatible\" content=\"ie=edge\">\n" +
                    "        <link rel=\"stylesheet\" href=\"css/user_main.css\">\n" +
                    "        <script\n" +
                    "            src=\"https://kit.fontawesome.com/64d58efce2.js\"\n" +
                    "            crossorigin=\"anonymous\">\n" +

```

```

" ></script>\n" +
" <title>CRS</title>\n" +
" </head>\n" +
" <body>\n" +
" <aside>\n" +
" <figure>\n" +
" <div id=\"avatar\"><img class=\"logo2\"
src=\"images/person_logo.png\"></div>\n" +
" <a href=\"./Profile\">\n" +
" <button style=\"background:none;border:none;font-size:
1.3em;font-weight: bold;margin-left: 15%;\">
+FirstName+\" \" + LastName+\"</button></a>\n" +
" </figure>\n" +
" <img class=\"img\" src=\"images/menu.svg\" alt=\"menu\">\n" +
" <nav>\n" +
" <ul>\n" +
" <li><a href=\"./Prfl_Data\">Profile</a></li>\n" +
" \n" +
" <li><a href=\"./Prfl_CarAvailDate\">Book A Car</a></li>\n" +
" <li><a href=\"./Prfl_Booking_Hist\">Booking
History</a></li>\n" +
" <li><a href=\"./Logout\">Logout</a></li>\n" +
" </ul>\n" +
" </nav>\n" +
" </aside>\n" +
" <main> \n" +
" <div class=\"logosec\"><img class=\"logo\"
src=\"images/town_r6pc.svg\" alt=\"logo\">\n" +
" \n" +
" </div>\n" +
" <h1>Welcome "+FirstName+" "+LastName+"</h1> \n" +
" <p> Travel Hassle-Free With Easy Services.</p>\n" +
" \n" +
" </main>\n" +
" <script>\n" +
" \n" +
" (function() {\n" +
" var menu = document.querySelector('ul'),\n" +
" menulink = document.querySelector('img');\n" +
" \n" +
" menulink.addEventListener('click', function(e) {\n" +
" menu.classList.toggle('active');\n" +
" e.preventDefault();\n" +
" });\n" +
" })();\n" +
" \n" +
" </script>\n" +
" </body>\n" +
" </html>");
}

```

```
        else
        {
            response.sendRedirect("signin-signup.html");
            response.setHeader("Cache-Control", "no-cache, no-store, must-revalidate");
            response.setHeader("Pragma", "no-cache");
            response.setHeader("Expires", "0");

        }
    }
}

// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on
the left to edit the code.">
/**
 * Handles the HTTP <code>GET</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Handles the HTTP <code>POST</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Returns a short description of the servlet.
 *
 * @return a String containing servlet description
 */
@Override
public String getServletInfo() {
    return "Short description";
}
```

```
    } // </editor-fold>
}
```

Register.java

```
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 *
 * @author blraj
 */
@WebServlet(urlPatterns = {"/Register"})
public class Register extends HttpServlet {

    /**
     * Processes requests for both HTTP <code>GET</code> and <code>POST</code>
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            /* TODO output your page here. You may use following sample code. */
            Connection con=null;
            Statement stm=null;
            ResultSet rs=null;
            String fname = request.getParameter("firstname");
            String lname = request.getParameter("lastname");
            String email = request.getParameter("email");
            String phnumber = request.getParameter("phonenummer");
            String dob = request.getParameter("dob");
```

```

String adrnumber = request.getParameter("adrnumber");
String dlnumber = request.getParameter("dlnumber");
String street = request.getParameter("street");
String city = request.getParameter("city");
String state = request.getParameter("state");
String zipcode = request.getParameter("zipcode");
String password = fname+zipcode;
Vroom person = new Vroom();
int id = person.memID;
/* TODO output your page here. You may use following sample code. */
try {
    Class.forName("com.mysql.jdbc.Driver");
    con=DriverManager.getConnection("jdbc:mysql://localhost/crs","root","");
    PreparedStatement login;
    PreparedStatement reg;
    reg=con.prepareStatement("insert into customer values(?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)");

    reg.setInt(1, id);
    reg.setString(2, fname);
    reg.setString(3, lname);
    reg.setString(4, email);
    reg.setString(5, phnumber);
    reg.setString(6, dob);
    reg.setString(7, dlnumber);
    reg.setString(8, adrnumber);
    reg.setString(9, street);
    reg.setString(10, city);
    reg.setString(11, state);
    reg.setString(12, zipcode);
    reg.executeUpdate();
    String altmsg = "Please make note of your login credentials\\nUsername : "+id+"\\nPassword : "+password;

    login=con.prepareStatement("insert into logincredentials values(?, ?)");
    login.setInt(1, id);
    login.setString(2, password);
    login.executeUpdate();

    response.setContentType("text/html");
    out.println("<script type='text/javascript'>");
    out.println("alert('"+ altmsg +"');");
    out.println("window.location.href = \"signin-signup.html\";");
    out.println("</script>");
} catch (ClassNotFoundException ex) { out.println("Invalid1");
    Logger.getLogger(Register.class.getName()).log(Level.SEVERE, null, ex);
} catch (SQLException ex) {
    Logger.getLogger(Register.class.getName()).log(Level.SEVERE, null, ex);
}
}

```



```

    }
    // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on
the left to edit the code.">
    /**
     * Handles the HTTP <code>GET</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Handles the HTTP <code>POST</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Returns a short description of the servlet.
     *
     * @return a String containing servlet description
     */
    @Override
    public String getServletInfo() {
        return "Short description";
    } // </editor-fold>
}

```

Vroom.java

```

import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

```

```
import java.sql.SQLException;
import java.sql.Statement;
import java.text.ParseException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.http.HttpSession;
import java.text.SimpleDateFormat;
import java.util.Date;
/**
 *
 * @author blraj
 */
public class Vroom {

    int    memID;
    int    bookID;
    int    billID;
    private static ArrayList<Integer> list_cust_ids = new ArrayList<Integer>();
    private static ArrayList<Integer> list_book_ids = new ArrayList<Integer>();
    private static ArrayList<Integer> list_bill_ids = new ArrayList<Integer>();
    private static ArrayList<Integer> list_carid_notavail = new ArrayList<Integer>();
    public HashMap<Integer,ArrayList<String>> caravail=new
HashMap<Integer,ArrayList<String>>();
    public HashMap<Integer,ArrayList<String>> car_details=new
HashMap<Integer,ArrayList<String>>();
    public HashMap<Integer,ArrayList<String>> carloc = new
HashMap<Integer,ArrayList<String>>();
    public HashMap<Integer,ArrayList<ArrayList<String>>> user_bookings=new
HashMap<Integer,ArrayList<ArrayList<String>>>();

    public Vroom() {
        memIdGenr();
        bookIdGenr();
        billIdGenr();
        poplcars();
        poploc();
    }

    private void memIdGenr()
    {
        try{
            /* TODO output your page here. You may use following sample code. */
            Connection con=null;
            Statement stm=null;
            ResultSet rs=null;
            Class.forName("com.mysql.jdbc.Driver");
            con=DriverManager.getConnection("jdbc:mysql://localhost/crs","root","");
            PreparedStatement id_fetcher;
```

```
String query=("select *from customer");
stm=con.createStatement();
rs=stm.executeQuery(query);
if(rs.next())
{
    list_cust_ids.add(Integer.parseInt(rs.getString("UserID")));
}
this.memID = (int)(Math.random() * (2047300 - 2047100 + 1) + 2047100);
while(list_cust_ids.contains(memID))
{
    this.memID = (int)(Math.random() * (2047300 - 2047100 + 1) + 2047100);
}
} catch (ClassNotFoundException ex) {
    Logger.getLogger(Register.class.getName()).log(Level.SEVERE, null, ex);
} catch (SQLException ex) {
    Logger.getLogger(Register.class.getName()).log(Level.SEVERE, null, ex);
}
}

private void bookIdGenr()
{
    try{
        /* TODO output your page here. You may use following sample code. */
        Connection con=null;
        Statement stm=null;
        ResultSet rs=null;
        Class.forName("com.mysql.jdbc.Driver");
        con=DriverManager.getConnection("jdbc:mysql://localhost/crs","root","");
        PreparedStatement id_fetcher;
        String query=("select *from booking");
        stm=con.createStatement();
        rs=stm.executeQuery(query);
        if(rs.next())
        {
            list_book_ids.add(Integer.parseInt(rs.getString("BookingID")));
        }

        this.bookID = (int)(Math.random() * (000000000 - 999999999 + 1) + 999999999);
        while(list_book_ids.contains(bookID))
        {
            this.bookID = (int)(Math.random() * (000000000 - 999999999 + 1) + 999999999);
        }
        } catch (ClassNotFoundException ex) {
            Logger.getLogger(Register.class.getName()).log(Level.SEVERE, null, ex);
        } catch (SQLException ex) {
            Logger.getLogger(Register.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

private void billIdGenr()
```

```

{
    try{
        /* TODO output your page here. You may use following sample code. */
        Connection con=null;
        Statement stm=null;
        ResultSet rs=null;
        Class.forName("com.mysql.jdbc.Driver");
        con=DriverManager.getConnection("jdbc:mysql://localhost/crs","root","");
        PreparedStatement id_fetcher;
        String query=("select *from billing");
        stm=con.createStatement();
        rs=stm.executeQuery(query);
        if(rs.next())
        {
            list_bill_ids.add(Integer.parseInt(rs.getString("BillID")));
        }

        this.billID = (int)(Math.random() * (0000000000 - 999999999 + 1) + 999999999);
        while(list_bill_ids.contains(billID))
        {
            this.billID = (int)(Math.random() * (0000000000 - 999999999 + 1) + 999999999);
        }
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(Register.class.getName()).log(Level.SEVERE, null, ex);
    } catch (SQLException ex) {
        Logger.getLogger(Register.class.getName()).log(Level.SEVERE, null, ex);
    }
}

private void poplcars()
{
    Connection con=null;
    Statement stm=null;
    ResultSet rs=null;
    ResultSet rs2=null;
    int avail=1;
    try {
        Class.forName("com.mysql.jdbc.Driver");
        con=DriverManager.getConnection("jdbc:mysql://localhost/crs","root","");
        String query="select *from caravailability where AvailFlag='"+avail+"'";
        stm=con.createStatement();
        rs=stm.executeQuery(query);
        while(rs.next())
        {
            String query2="select *from car where
CarID='"+Integer.parseInt(rs.getString("CarID"))+"'";
            stm=con.createStatement();
            rs2=stm.executeQuery(query2);
            if(rs2.next())
            {

```

```

        ArrayList<String> car = new ArrayList<String>();
        car.add(rs2.getString("CarID"));
        car.add(rs2.getString("LocID"));
        car.add(rs2.getString("CarCategory"));
        car.add(rs2.getString("RegNumber"));
        car.add(rs2.getString("ModelName"));
        car.add(rs2.getString("ModelYear"));
        car.add(rs2.getString("PersonIncludable"));
        car.add(rs2.getString("CostPerDay"));

        if(!car_details.containsKey(Integer.parseInt(rs2.getString("CarID"))))
        {
            car_details.put(Integer.parseInt(rs2.getString("CarID")), car);
        }
        insert_caravail(Integer.parseInt(rs2.getString("CarID")),car);
    }
} catch (ClassNotFoundException ex) {
    Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null, ex);
} catch (SQLException ex) {
    Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null, ex);
}
}

private void poploc()
{
    Connection con=null;
    Statement stm=null;
    ResultSet rs=null;
    int id=1;
    try {
        Class.forName("com.mysql.jdbc.Driver");
        con=DriverManager.getConnection("jdbc:mysql://localhost/crs","root","");
        String query="select *from location";
        stm=con.createStatement();
        rs=stm.executeQuery(query);
        while(rs.next())
        {
            ArrayList<String> loc = new ArrayList<String>();
            loc.add(rs.getString("LocID"));
            loc.add(rs.getString("Street"));
            loc.add(rs.getString("City"));
            loc.add(rs.getString("State"));
            loc.add(rs.getString("Zipcode"));
            insert_carloc(Integer.parseInt(rs.getString("LocID")),loc);
        }
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null, ex);
    } catch (SQLException ex) {
        Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null, ex);
    }
}

```

```
    }  
}  
  
private void insert_carloc(int id, ArrayList<String> loc)  
{  
    carloc.put(id, loc);  
}  
  
private void delete_carloc(int id)  
{  
    if(carloc.containsKey(id))  
    {  
        carloc.remove(id);  
    }  
}  
  
private void insert_caravail(int id, ArrayList<String> car)  
{  
    caravail.put(id, car);  
}  
  
private void delete_caravail(int id)  
{  
    if(caravail.containsKey(id))  
    {  
        caravail.remove(id);  
    }  
}  
  
public void deleteall_caravail()  
{  
    caravail.clear();  
}  
  
private void insert_car_notavail(int id)  
{  
    if(!list_carid_notavail.contains(id))  
    {  
        list_carid_notavail.add(id);  
    }  
}  
private void delete_car_notavail(int id)  
{  
    if(list_carid_notavail.contains(id))  
    {  
        list_carid_notavail.remove(id);  
    }  
}  
public void pop_usr_bkng(int userid)  
{
```

```

String result="yes";
ArrayList<ArrayList<String>> all_bkng = new ArrayList<ArrayList<String>>();
Connection con=null;
Statement stm=null;
ResultSet booking=null;
ResultSet billing=null;
try {
    Class.forName("com.mysql.jdbc.Driver");
    con=DriverManager.getConnection("jdbc:mysql://localhost/crs","root","");
    String query="select *from booking where CustomerID='"+userid+"'";
    stm=con.createStatement();
    booking=stm.executeQuery(query);
    while(booking.next())
    {
        ArrayList<String> ech_bk = new ArrayList<String>();
        ech_bk.add(booking.getString("BookingID"));
        ech_bk.add(booking.getString("CustomerID"));
        ech_bk.add(booking.getString("CarID"));
        ech_bk.add(booking.getString("PickDt"));
        ech_bk.add(booking.getString("ReturnDt"));
        ech_bk.add(booking.getString("PickLocID"));
        ech_bk.add(booking.getString("Amount"));
        ech_bk.add(booking.getString("Status"));

        String query2="select *from billing where
BookingID='"+Integer.parseInt(booking.getString("BookingID"))+"'";
        stm=con.createStatement();
        billing=stm.executeQuery(query2);
        if(billing.next())
        {
            ech_bk.add(billing.getString("BillID"));
        }
        all_bkng.add(ech_bk);
    }

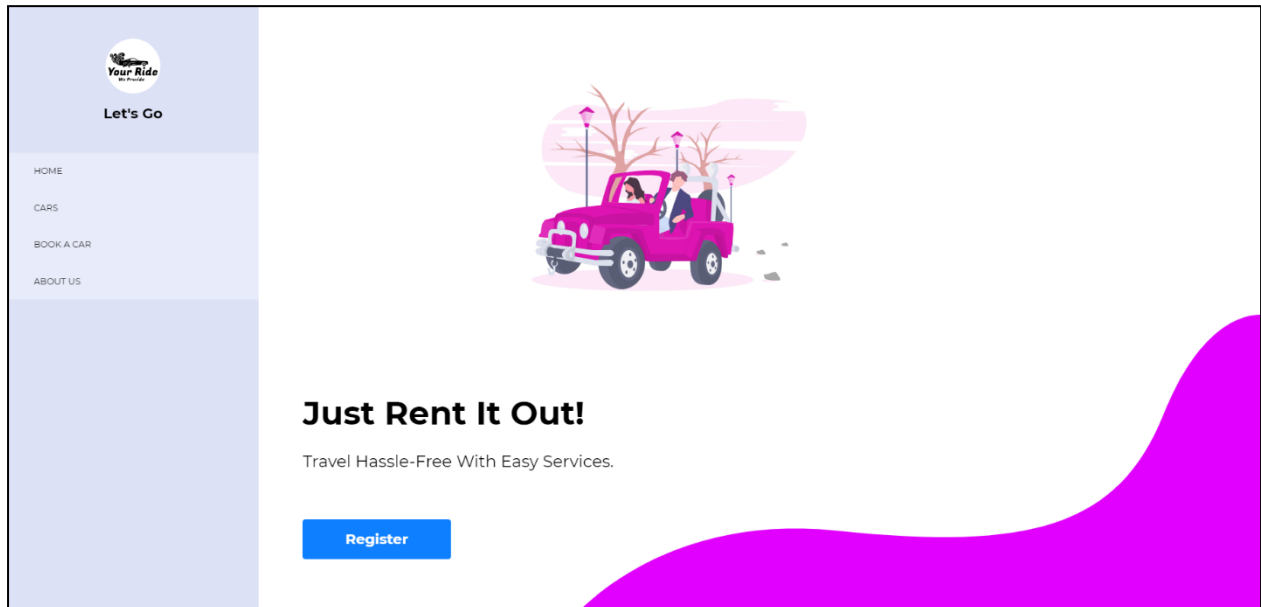
    if(!user_bookings.containsKey(userid))
    {
        user_bookings.put(userid, all_bkng);
    }

} catch (ClassNotFoundException ex) {
    Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null, ex);
} catch (SQLException ex) {
    Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null, ex);
}
}
}

```

5.2 SCREEN SHOTS

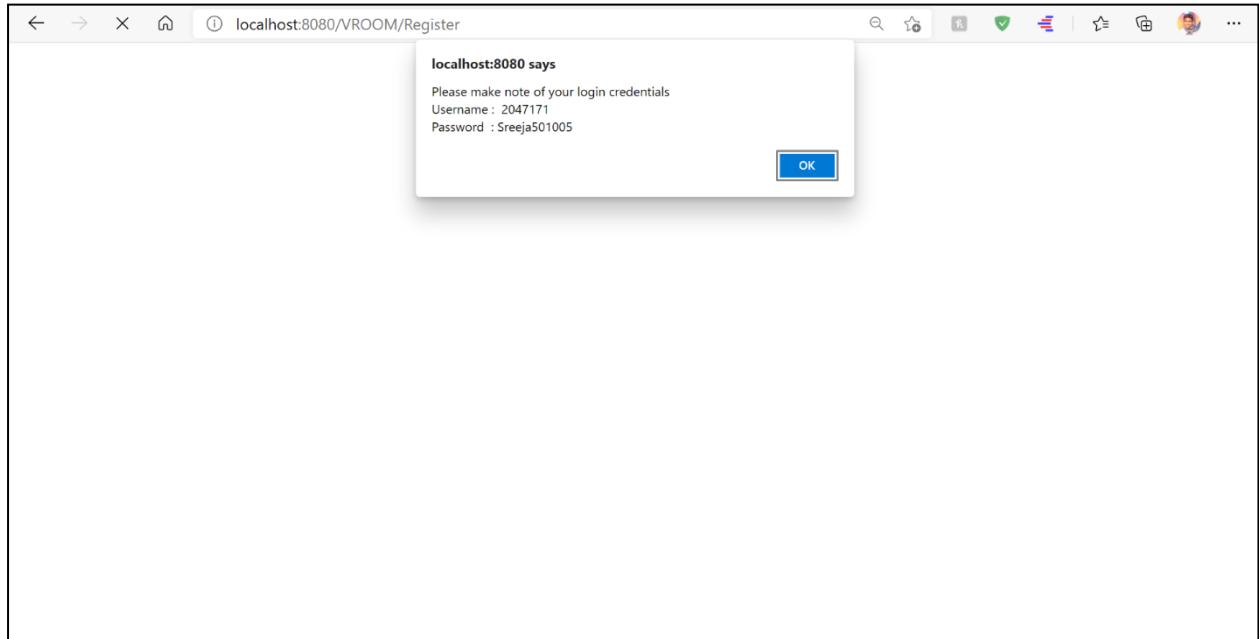
HOME PAGE



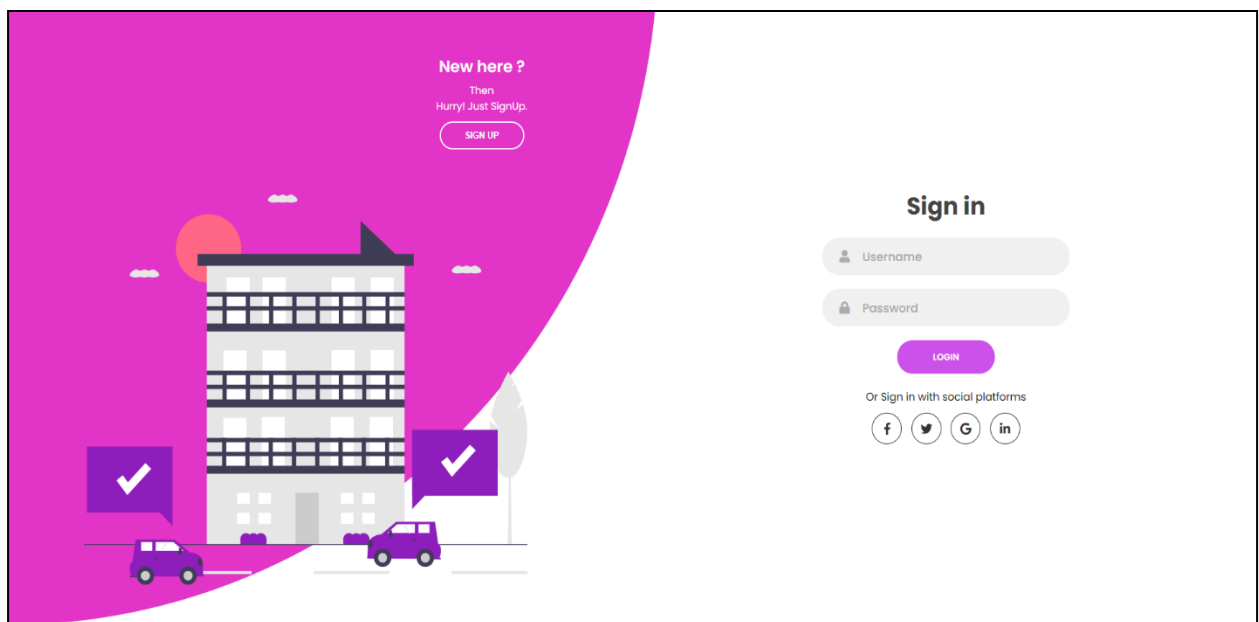
REGISTER

The Register page has a light blue sidebar on the left with the title 'Register'. The sidebar contains a list of input fields for user registration: Name (Sreeja), Surname (C S), Email (Sreeja@crs.in), Phone Number (9361048881), Date of Birth (1998-06-18), License Number (KA12386789), Address (S G PALYA), City (Bengaluru), State (KA), and Pin Code (501005). A blue 'SIGN UP' button is at the bottom of the sidebar. The main content area has a large, wavy pink shape on the right. On the left, there is a dark blue box with three white horizontal lines, and a person is sitting on top of it. The text 'One of us ?' is displayed in bold, followed by the subtitle 'Then start Booking And Enjoy Your Journey.' and a blue 'LOGIN' button.

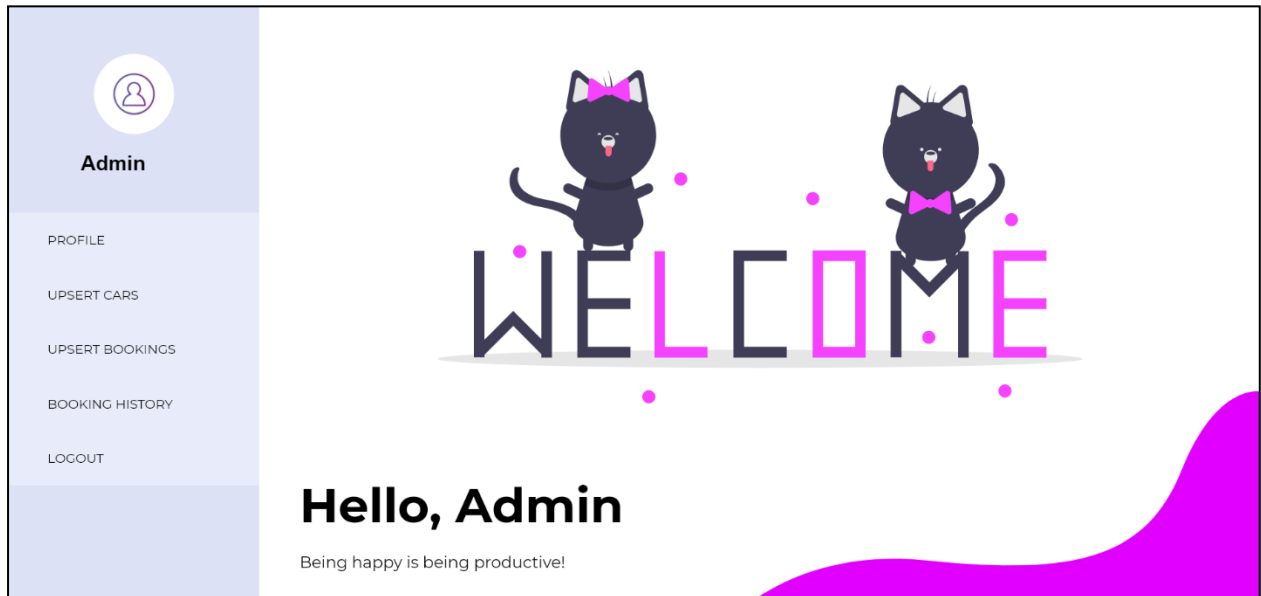
REGISTRATION CONFIRMATION



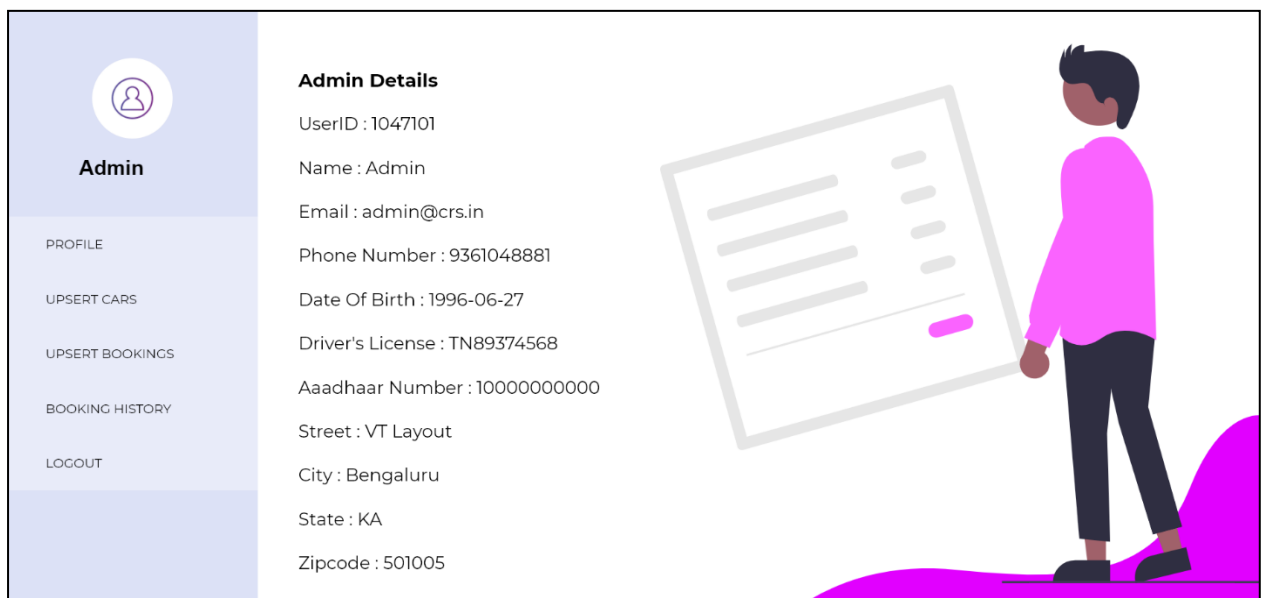
LOGIN



ADMIN LOGIN - WELCOME PAGE



ADMIN PROFILE



ADMIN CAR INSERT / UPDATE

PROFILE

UPSERT CARS

UPSERT LOCATION

BOOKING HISTORY

LOGOUT

Enter the Car Details

Car ID

Choose the Location ID

Choose Your Car Type

Car Registration Number


Model Name

Model Year

Person Includable

Cost Per Day

INSERT / UPDATE



ADMIN LOCATION INSERT / UPDATE

Admin

PROFILE

UPSERT CARS

UPSERT LOCATION

BOOKING HISTORY

LOGOUT

Enter the Location Details

Location ID

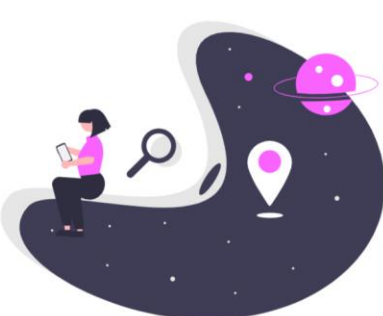
Street

City

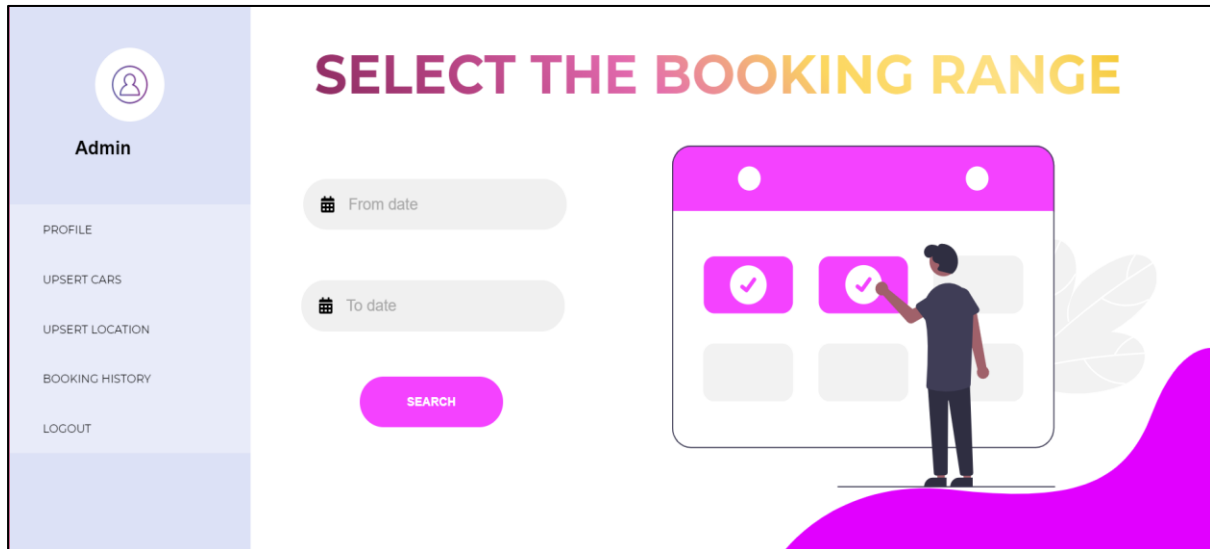
State

Zipcode

INSERT / UPDATE

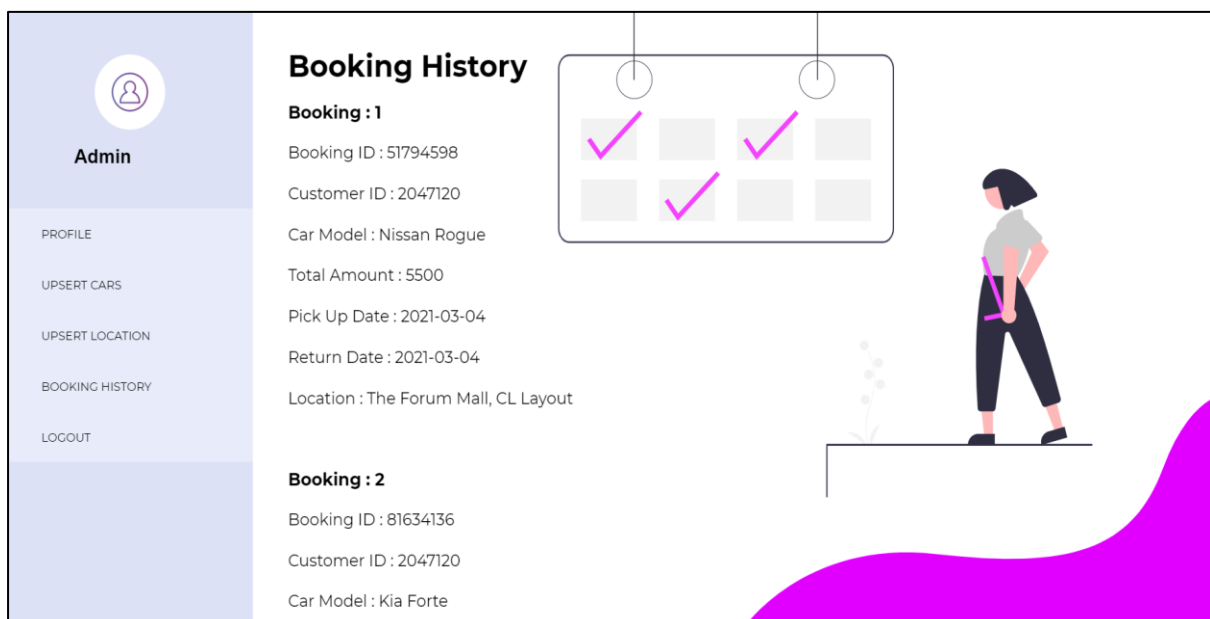


ADMIN BOOKING HISTORY DATE PICKER




The interface features a sidebar on the left with a user profile icon and the name 'Admin'. Below this, a list of menu items includes 'PROFILE', 'UPSET CARS', 'UPSET LOCATION', 'BOOKING HISTORY', and 'LOGOUT'. The main content area is titled 'SELECT THE BOOKING RANGE' in large, bold, multi-colored letters. It contains two date selection fields: 'From date' and 'To date', each with a calendar icon. A prominent pink 'SEARCH' button is positioned below these fields. To the right, an illustration shows a person interacting with a large digital calendar that has several dates marked with checkmarks. The background includes a stylized pink wave at the bottom right.

ADMIN BOOKING HISTORY



The interface features a sidebar on the left with a user profile icon and the name 'Admin'. Below this, a list of menu items includes 'PROFILE', 'UPSET CARS', 'UPSET LOCATION', 'BOOKING HISTORY', and 'LOGOUT'. The main content area is titled 'Booking History' in bold. It displays two booking entries, each with a set of details and a corresponding illustration.
Booking : 1
Booking ID : 51794598
Customer ID : 2047120
Car Model : Nissan Rogue
Total Amount : 5500
Pick Up Date : 2021-03-04
Return Date : 2021-03-04
Location : The Forum Mall, CL Layout
Booking : 2
Booking ID : 81634136
Customer ID : 2047120
Car Model : Kia Forte
To the right of the text, there is an illustration of a person standing next to a large calendar grid. The calendar grid has several dates marked with checkmarks. The background includes a stylized pink wave at the bottom right.

ADMIN LOCATION INSERT / UPDATE

**Admin**

PROFILE

UPERT CARS

UPERT LOCATION

BOOKING HISTORY

LOGOUT

Enter the Location Details

Location ID

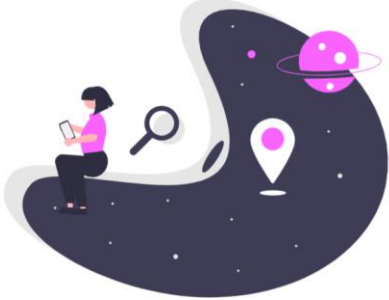
Street

City


State

Zipcode

INSERT / UPDATE



ADMIN PROFILE

**Admin**

PROFILE

UPERT CARS

UPERT BOOKINGS

BOOKING HISTORY

LOGOUT

Admin Details

UserID : 1047101

Name : Admin

Email : admin@crs.in

Phone Number : 9361048881

Date Of Birth : 1996-06-27

Driver's License : TN89374568


Aaadhaar Number : 10000000000

Street : VT Layout

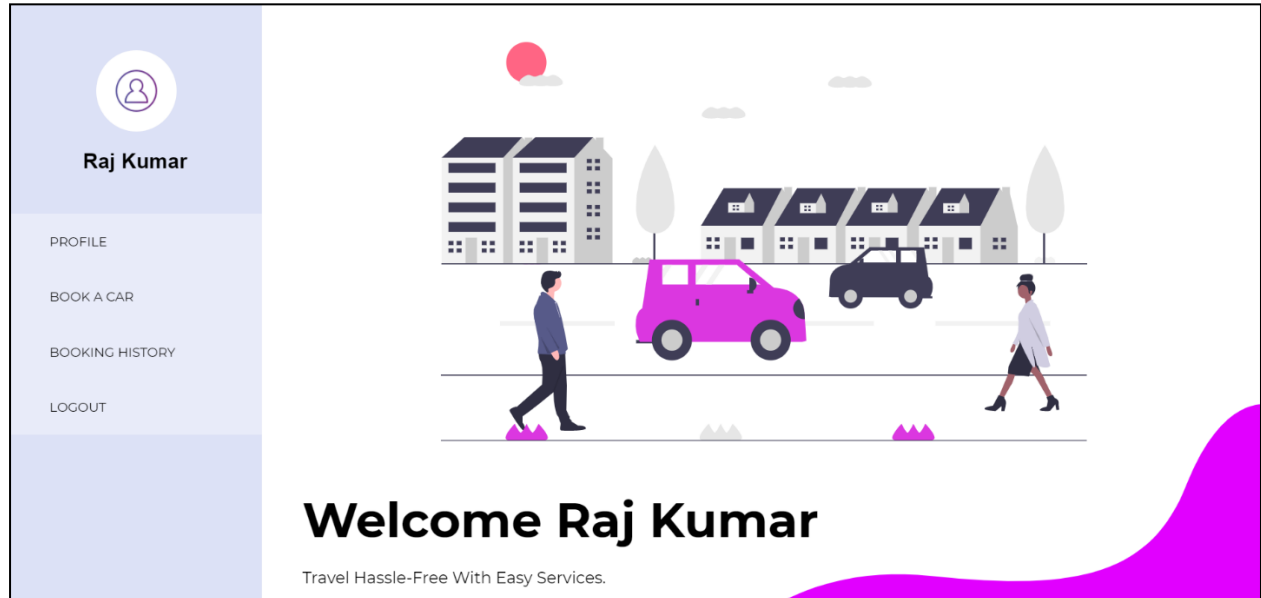
City : Bengaluru

State : KA

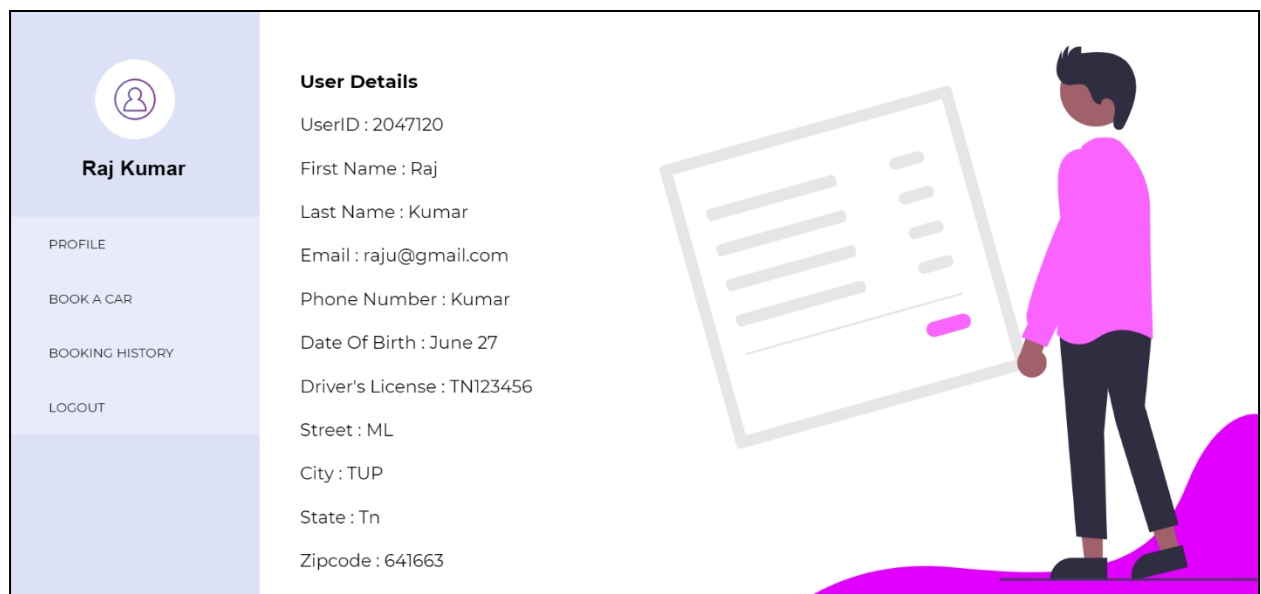
Zipcode : 501005




USER LOGIN - WELCOME PAGE



USER PROFILE



CAR AVAILABILITY – DATE PICKER



Raj Kumar


PROFILE


BOOK A CAR

BOOKING HISTORY

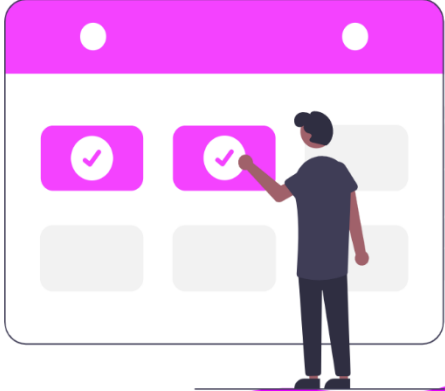
LOGOUT

START BOOKING YOUR RIDE

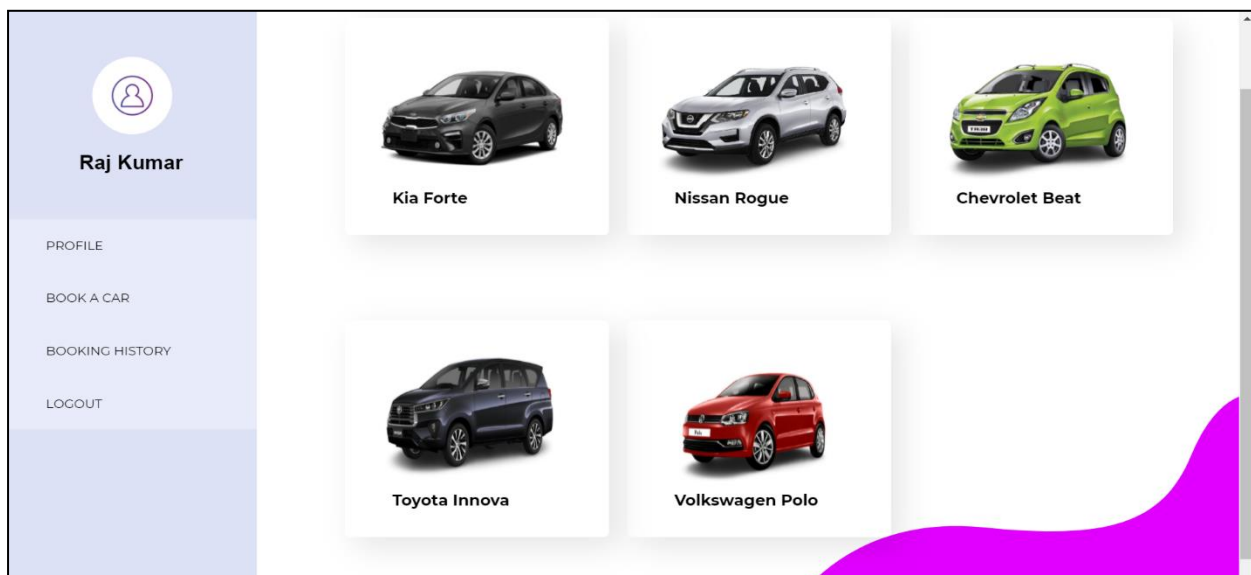
 From date

 To date


SUBMIT



CAR AVAILABILITY



SELECTIVE CAR DETAILS



Raj Kumar

PROFILE

BOOK A CAR

BOOKING HISTORY

LOGOUT

Car Details

CarID : 1003

CarCategory : Hatchback

Model Name : Chevrolet Beat

Model Year : 2018

Person Includable : 5

Cost Per Day : Rs. 2500

Location Details

LocID : 2001


Street : The Forum Mall, CL Layout

City : Bengaluru


State : Karnataka

Zipcode : 560095

BOOK NOW



CAR AVAILABILITY



Raj Kumar

PROFILE

BOOK A CAR

BOOKING HISTORY

LOGOUT

Booking Details

Model Name : Chevrolet Beat

Model Year : 2018

CarCategory : Hatchback

Person Includable : 5

Cost Per Day : Rs. 2500

PickUp Date : 2021-03-04

Return Date : 2021-03-05


Pick-Up Location :-

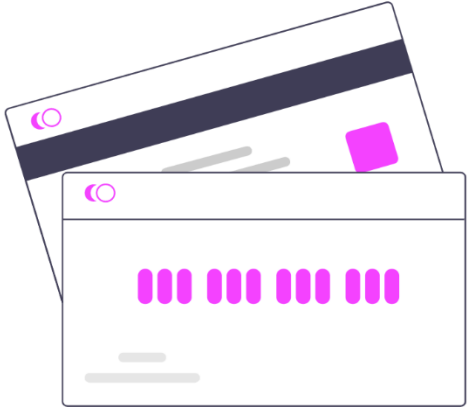
The Forum Mall, CL Layout

Bengaluru, Karnataka

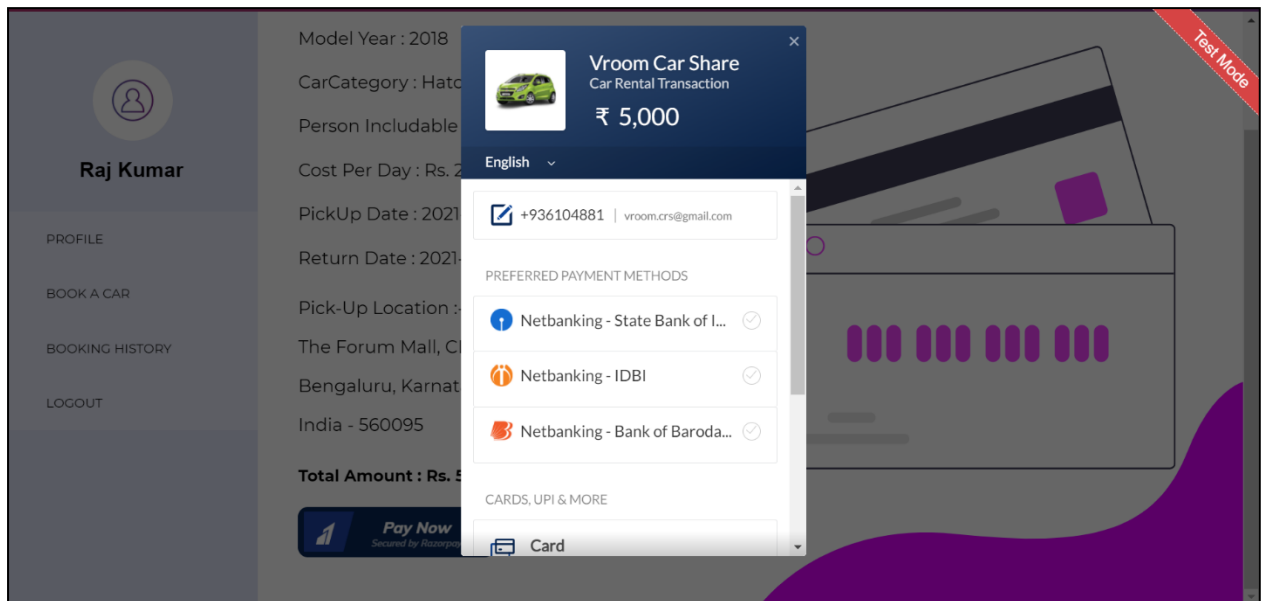
India - 560095

Total Amount : Rs. 5000

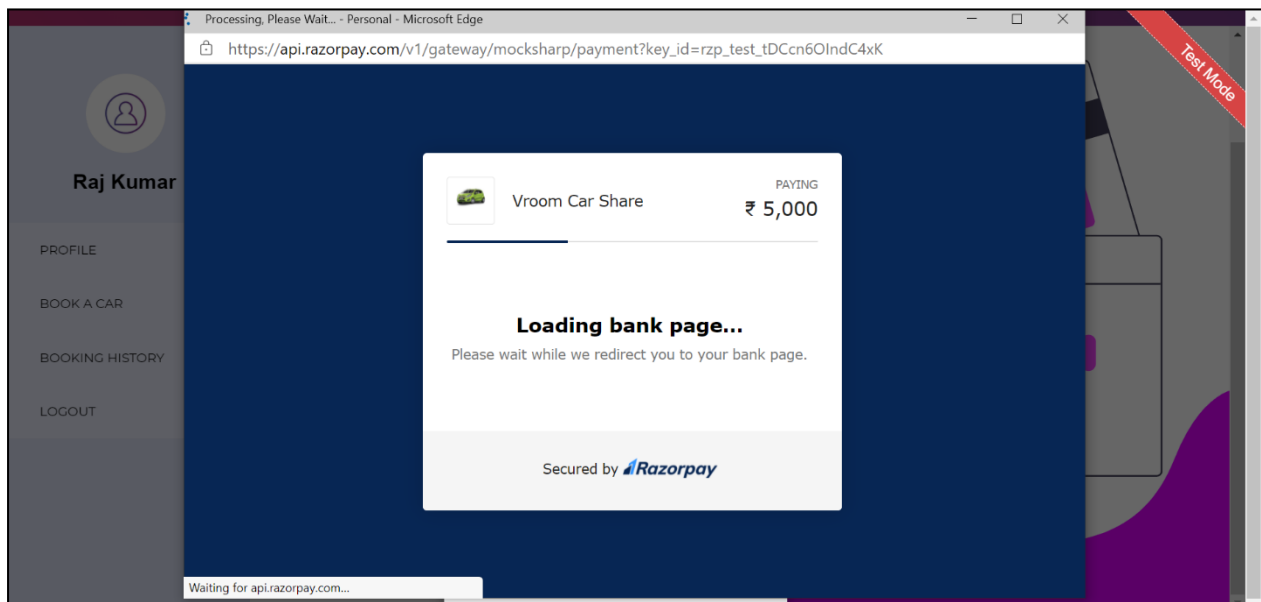
 **Pay Now**
Secured by Razorpay



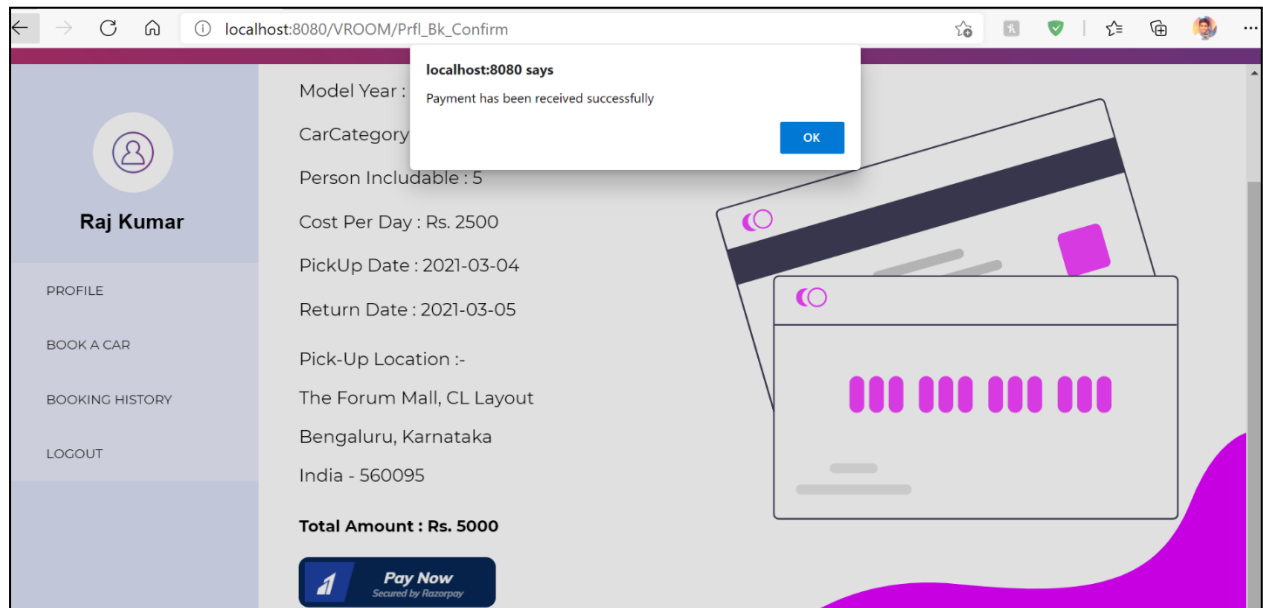
PAYMENT PAGE - 01



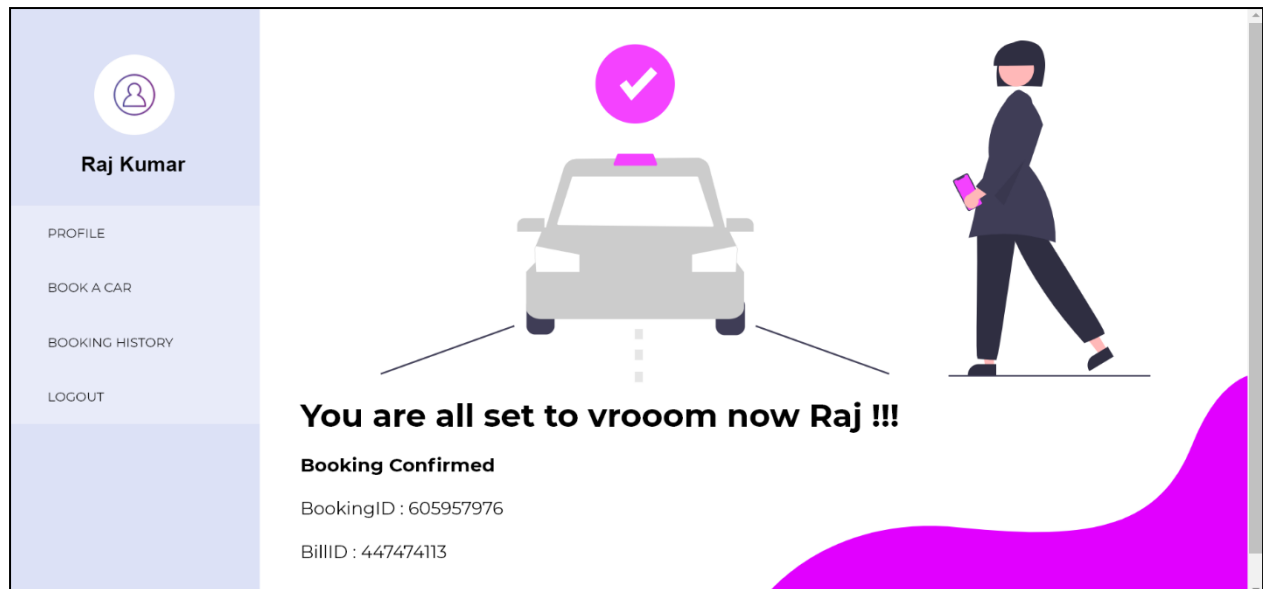
PAYMENT PAGE - 02



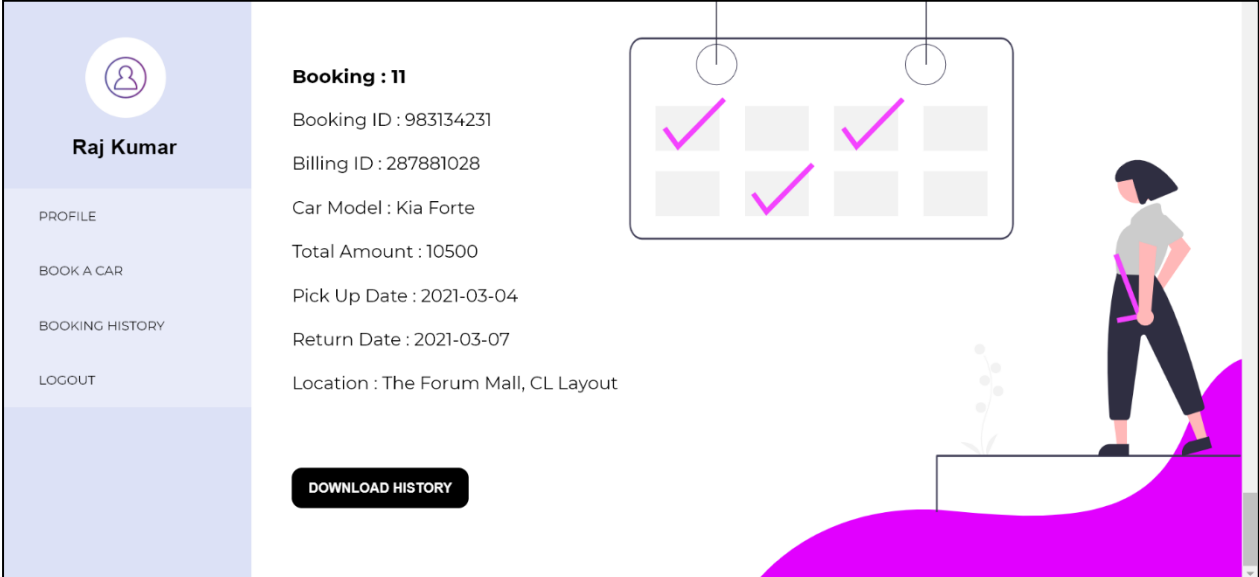
PAYMENT SUCCESS



BOOKING CONFIRMATION



BOOKING HISTORY



The interface shows a user profile for Raj Kumar on the left with navigation links: PROFILE, BOOK A CAR, BOOKING HISTORY, and LOGOUT. The main area displays booking details for Booking ID: 983134231, Billing ID: 287881028, Car Model: Kia Forte, Total Amount: 10500, Pick Up Date: 2021-03-04, Return Date: 2021-03-07, and Location: The Forum Mall, CL Layout. A 'DOWNLOAD HISTORY' button is present. A calendar icon shows a grid with three dates marked with checkmarks. An illustration of a person standing on a pink wave is on the right.

Raj Kumar

PROFILE

BOOK A CAR

BOOKING HISTORY

LOGOUT

Booking : 11

Booking ID : 983134231

Billing ID : 287881028

Car Model : Kia Forte

Total Amount : 10500

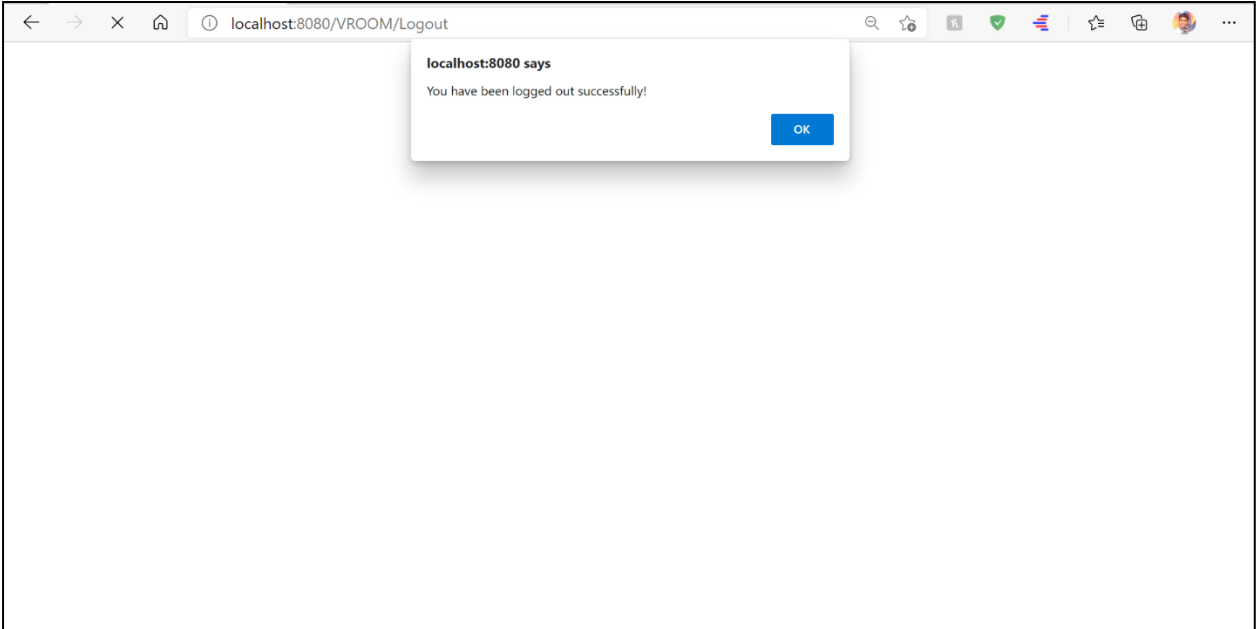
Pick Up Date : 2021-03-04

Return Date : 2021-03-07

Location : The Forum Mall, CL Layout

DOWNLOAD HISTORY

SESSION LOGOUT



A browser window at localhost:8080/VROOM/Logout shows a confirmation dialog box. The dialog box has a title 'localhost:8080 says' and a message 'You have been logged out successfully!' with an 'OK' button.

localhost:8080/VROOM/Logout

localhost:8080 says

You have been logged out successfully!

OK

6. TESTING

6.1 TEST PLAN

6.1.1 OBJECTIVE

The main objective of unit testing is to verify whether every unit operates as intended. A function, procedure, method, or even the entire module can be considered a separate unit. Unit testing can be conducted manually, but automated testing is a more common practice.

6.1.2 LOGGING TESTS AND REPORTING

Developers fix the defects found in each unit. If these defects are related to other modules and units, they must be reported.

6.1.3 OBJECTIVE

System testing is generally conducted after Unit Testing. The objective of System Testing is to evaluate compliance of an integrated application with its requirements.

6.1.4 PERFORMANCE TESTING

Performing testing is conducted to detect issues related to:

- memory consumption
- power utilization
- network connectivity
- operating in the background
- switching between applications
- memory leakage

6.1.3.2 INTERRUPT TESTING

This testing type examines how an application reacts to interruption and resumes to its previous state. There are numerous reasons that can potentially interrupt the operation of an app, such as getting a phone call, messages, notifications, battery low, etc.

6.1.5 USABILITY TESTING

Usability testing is performed to check whether the application is easy to use and understand for the end-user.

6.1.6 INSTALLATION AND LAUNCH TESTING

Installation testing aims to detect whether there are any issues during the installation, uninstallation, and updating process. Once the application has been installed, a QA engineer also checks the launching process.

6.1.7 FUNCTIONAL TESTING

All the functions and features of the application are tested to verify whether they operate according to the specification.

6.1.8 PASS/FAIL CONDITIONS

All the conditions when tests pass or fail are defined and described.

6.2 TEST CASES AND TEST REPORTS

SI no	Test Case	Description	Expected Output	Actual Output	Result
1	Launch site	Site url: http://localhost:8080/VROOM/	Launches CRS site	Site launched	PASS
2	SignUp	User can enter details to register successfully	User enters correct details as first name,last name,email,DOB,address,aadhar number,DL number	User enters correct details and successfully created login credentials and alert box pops up and redirects to login page	PASS
3	Login	User enter userid and password to login successfully	User enters correct userid and password	Upon entering correct credentials user logs in and gets redirect to user profile page	PASS

4	User Profile Navigation	User can navigate through profile, book a car, booking history and logout	User can view profiles, can book a car , can view booking history and upon completion can logout and go to home page.	User views details , books a car for a particular period, completes payment, views booking history and finally logs out. And redirects to the home page.	PASS
5	Duplicate entries to sign up	User enters duplicate entries to register	Upon entering duplicated details user gets invalid message	The page shows invalid messages upon duplicate details entered.	PASS
6	Incorrect login with incorrect login credentials	User enters incorrect userid or password to login	Upon entering incorrect userid or password user gets invalid user or password alert message	Alert box pops up showing invalid user or password message.	PASS
7	Car availability	Users can view the cars available.	Upon submitting the trip dates user can view the available cars	User booked the car successfully on the required dates.	PASS
8	Log Out	User can log out of user profile	Upon clicking on logout user logs out of the session	User logs out and gets an alert message with logged out successfully and redirects to the login page.	PASS
9	Home page navigation	User can navigate through contact us, about us and book car	Redirects about us and contact us page.	User views about us and contact us page	PASS
10	Admin Login	Admin enters correct details to login	Admin logs in and redirects to admin profile page	Admin lands on profile page upon successful login	PASS
11	Admin Profile navigation	Admin can navigate through upsert car , upsert location , view bookings and log out	After landing on admin profile page admin can upsert new cars , new locations and can view booking history of users.	Admin upserts new car , new locations and views booking history of users. On log out admin get redirected to login page.	PASS
12	Admin login with incorrect credentials	Admin enter incorrect credentials	incorrect credentials pops up alert box with invalid credentials message	Admin gets alert box indicating invalid user ID or password message	PASS

7. CONCLUSION

The project entitled “Car Rental System” was completed successfully. The system has been developed with much care and free of errors and at the same time it is efficient and less time consuming.

Car rental business has emerged with a new goody compared to the experience where every activity concerning car rental business is limited to a physical location only. Even though the physical location has not been totally eradicated⁶ the nature of functions and how these functions are achieved has been reshaped by the power of internet. Nowadays, customers can

reserve cars online, rent car online, and have the car brought to their doorstep once the customer is a registered member or go to the office to pick the car.

The purpose of this project was to develop a web application and an android application for users to rent cars with use on the go. This project helped us in gaining valuable information and practical knowledge on several topics like designing UI for the HTML pages using ADOBE XD, usage of responsive templates, development of interactive web application, and management of database using MySQL.

The entire system is secured. Also, the project helped us understand about the development phases of a project and software development life cycle. We learned how to test different features of a project. This project has given us great satisfaction in designing an web application which can be implemented by any company wanting to take their car rental business to web portal and provide users with the best service.

7.1 ADVANTAGES

The web based car rental system has offered an advantage to both customers as well as Car Rental Company to efficiently and effectively manage the business and satisfies customers need at the click of a button.

- The online car rental solution is fully functional and flexible.
- It is very easy to use.
- The online car rental system helps in back-office administration by streamlining and

standardizing the procedures.

- It saves a lot of time, money, and labour.
- Eco-friendly: The monitoring of the vehicle activity and the overall business becomes easy and includes the least of paperwork.
- The software acts as an office that is open 24/7.
- It increases the efficiency of the management at offering quality services to the customers.
- It provides custom features development and support with the software.

7.2 LIMITATIONS

- Only limited cars number of cars are available.
- The car rent price is comparatively higher.
- A stable internet connection is needed for the customer to book cars and also manipulate the booking when required
- There can be exceptional cases where the previous car renter not being able to return the car at the destined time, leading to car not being available for next booked person.

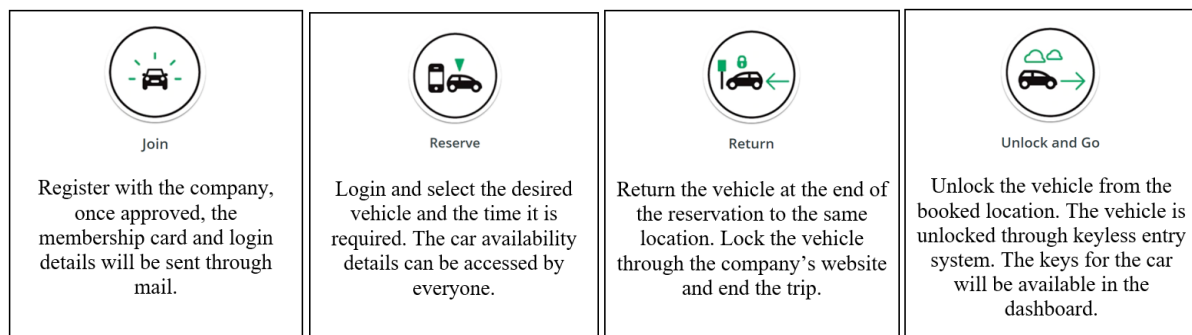
7.2 FUTURE ENHANCEMENT

There is a scope for further development in our project to a great extent. A number of features can be added to this system in future like providing users more features and optimizing the system to ease the booking process. A mobile application can also be developed for the car rental system thus providing the user with additional handy feature over web application.

One of the main implementations that was proposed initially for our rental system was to provide the user with the four easy steps: (Register, Reserve, Unlock&Go, Return). The company owned cars will be parked and made available to the customers at hotspots of the city like malls, showrooms, service centers. The customers can book the car near their location or at their preferred location and access them through a keyless entry system with OTP or E-card tapping.

The keyless entry system relies on digital, IoT-enabled technology. Technological solution integration aims to optimize customer service and minimize waiting times during vehicle rental. The rental of vehicles should thus be possible outside business hours. At the same time,

vehicle error messages are to be automatically recorded and transmitted to digitally map vehicle maintenance processes and intervals and trigger immediate corrections. A telematics box installed in the vehicle interior is connected to the vehicle's on-board computer for external communication. The solution operates via a mobile network combined with M2M technologies. A GPS and a GSM module, as well as gyro sensors, are integrated into the box. The box is used for data retrieval and communication, virtual handling of customer interaction, vehicle condition monitoring, and predictive maintenance applications.



At the same time, vehicle error messages are to be automatically recorded and transmitted to digitally map vehicle maintenance processes and intervals and trigger immediate corrections. A GPS and a GSM module, as well as gyro sensors, are integrated into the box. The box is used for data retrieval and communication, virtual handling of customer interaction, vehicle condition monitoring, and predictive maintenance applications. These features could have implemented unless the time did not limit us.

REFERENCE

[1] Octo Telematics. 2021. *OCTO SmartKeyLess* | Octo Telematics. [online] Available at: <<https://www.octotelematics.com/connected-vehicles/real-life-applications/octo-keyless/>> [Accessed 9 February 2021].