CAR RENTAL

MANAGEMENT SYSTEM

**MINI PROJECT REPORT**

|  |  |  |
| --- | --- | --- |
| **Submitted** | **by** |  |
| **SHYAM PRASAD S** |  | **231801166** |
| **VIGNESH A** |  | **231801187** |
| **SIVA MURUGAN B** |  | **231801167** |

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING IN

ARTIFICAL INTELLIGENCE OF DATA SCIENCE RAJALAKSHMI ENGINEERING COLLEGE (AUTONOMOUS) THANDALAM

CHENNAI-602105

2024 - 2025

## BONAFIDE CERTIFICATE

Certified that this project report “**CAR RENTAL MANAGEMENT SYSTEM**” is the bonafide work of **“SHYAM PRASAD S(231801166),VIGNESH A(231801187), SIVA MURUGAN B(231801167)”**

who carried out the project work under my supervision.

#### Submitted for the Practical Examination held on

**INTERNAL EXAMINER EXTERNAL EXAMINER**

**ABSTRACT**

The Car Rental Management System is an efficient, user-friendly software solution designed to streamline and automate the operations of car rental businesses. This system addresses the challenges of manual booking processes, vehicle inventory management, customer data handling, and payment tracking. By leveraging modern technologies, it ensures a seamless experience for both the rental service providers and their customers.

The system features modules for user registration, vehicle search, booking, and payment, supported by a dynamic database that maintains up-to-date records of vehicles, availability, and customer details. It offers robust functionality for managing vehicle categories, pricing models, and maintenance schedules. Advanced search filters enable users to find vehicles that match their preferences, such as model, type, or price range.

The platform integrates with secure payment gateways to facilitate online transactions, ensuring reliability and customer trust. For administrators, it provides insightful dashboards for tracking revenue, bookings, and operational metrics. Additionally, it supports automated notifications for booking confirmations, reminders, and maintenance alerts.

This system enhances efficiency by minimizing human error, reducing administrative workload, and delivering superior customer service. It is scalable and customizable, making it suitable for businesses of all sizes. The Car Rental Management System is an essential tool for modernizing the car rental industry, ensuring sustainability and competitiveness in a fast-paced digital world.

**TABLE OF CONTENTS**

#### INTRODUCTION:

* 1. INTRODUCTION
  2. OBJECTIVES
  3. MODULES

#### SURVEY OF TECHNOLOGIES:

* 1. SOFTWARE DESCRIPTION
  2. LANGUAGES:
     1. Sql
     2. Python

#### REQUIREMENTS AND ANALYSIS:

* 1. REQUIREMENT SPECIFICATION
  2. HARDWARE AND SOFTWARE REQUIRE
  3. ARCHITECTURE DIAGRAM
  4. ER DIAGRAM

#### PROGRAM CODE

1. **RESULTS SCREENSHOT**

#### CONCLUSION

1. **REFERENCES**

# INTRODUCTION:

The Car Rental Management System is a comprehensive software application designed to modernize and simplify the operations of car rental businesses. In today’s fast-paced world, the demand for convenient and efficient transportation services is higher than ever. Traditional methods of managing vehicle rentals, which often rely on manual record-keeping, can lead to inefficiencies, errors, and poor customer experiences. This system addresses these challenges by automating the entire rental process, from vehicle selection and booking to payment and return.The system provides an intuitive interface for customers to browse available vehicles, check pricing, and make reservations seamlessly. For business administrators, it offers tools to manage vehicle inventory, monitor bookings, track vehicle maintenance schedules, and generate performance reports. Its integration with secure payment gateways ensures reliable and hassle-free transactions.By digitizing operations, the Car Rental Management System reduces operational overhead, enhances accuracy, and improves service quality. This system is not only a solution to immediate operational challenges but also a step towards the digital transformation of the car rental industry, catering to the needs of businesses and customers alike.

**Objectives**

 **Streamline Rental Processes**  
Automate vehicle booking, return, and payment procedures to enhance efficiency and reduce manual errors.

 **Enhance Customer Experience**  
Provide an intuitive platform for customers to search, compare, and reserve vehicles with ease.

 **Optimize Vehicle Utilization**  
Effectively manage vehicle inventory, ensuring maximum availability and usage of the fleet.

 **Enable Real-time Data Access**  
Maintain an up-to-date database of vehicles, bookings, and customer information for seamless operations.

### MODULES:

.

 **User Management Module**

* Handles user registration, login, and profile management.
* Supports different user roles, such as customers and administrators.

 **Vehicle Management Module**

* Manages vehicle inventory, including details such as model, type, availability, and pricing.
* Tracks vehicle status for maintenance, repairs, and usage.

 **Booking and Reservation Module**

* Facilitates vehicle search based on customer preferences (e.g., type, price, location).
* Manages bookings, cancellations, and modifications in real-time.

 **Payment and Billing Module**

* Integrates secure payment gateways for online transactions.
* Generates invoices and tracks payment history for transparency.

 **Admin Dashboard Module**

* Provides a centralized interface for monitoring bookings, revenue, and fleet performance.
* Allows administrators to set pricing models, discounts, and promotional offers.

 **Notification Module**

* Sends automated notifications for booking confirmations, reminders, and payment receipts.
* Alerts administrators for maintenance schedules or vehicle unavailability.

 **Report Generation Module**

* Generates detailed reports on revenue, fleet utilization, and customer trends.
* Supports data-driven decision-making for business growth.

 **Feedback and Support Module**

* Allows customers to provide feedback on their rental experience.
* Provides support through FAQs and contact forms for issue resolution.

# SURVEY OF TECHNOLOGIES:

## Software Description :

This project utilizes a combination of software tools to create a comprehensive and efficient car rental management system:

* **Database Management System (DBMS):** Sql is chosen as the DBMS for the car rental management system due to its flexibility with unstructured data, ease of scalability, and ability to handle a diverse range of data types efficiently.
* **Integrated Development Environment (IDE):** PyCham is selected as the IDE for its Python-specific features, code completion, debugging tools, and seamless integration with Sql.

## Languages :

* **SQL:** SQL plays a crucial role in managing and interacting with data related to vehicles, customers, bookings, payments, and other essential components. Below are some key SQL queries used for the operations of such a system, assuming a relational database structure with common tables like customers, vehicles, bookings, and payments.
* **FLASK**: Flask, a lightweight and flexible Python web framework, serves as the backend framework for building the system's web interface. Flask facilitates rapid development by offering simplicity and scalability, Flask’s integration with SQL databases enables seamless data flow, making it possible for users to track inventory, manage orders, and generate reports in real-time. Together, SQL and Flask create a powerful, user-friendly, and responsive system for effective car rental management.
  + 1. **SQL:**

Sql plays a crucial role in the car rental management system by:

 **High Performance**

* SQL efficiently manages large datasets, handling complex queries and transactions with minimal response time.

 **Data Integrity**

* SQL ensures accuracy and reliability of data with constraints and relational integrity checks, preventing data duplication and inconsistency.

 **Scalability**

* SQL databases can scale vertically, accommodating increasing amounts of data as a business grows.

### 2.2.2 JAVA PROGRAMMING:

### 

 **Object-Oriented**: Java follows the principles of object-oriented programming (OOP), such as encapsulation, inheritance, and polymorphism.

 **Platform Independence**: Java code is compiled into bytecode, which runs on the Java Virtual Machine (JVM). This enables Java applications to run on any system with a JVM installed.

 **Simple**: Java has a straightforward syntax inspired by C++, making it relatively easy to learn, especially for those familiar with other programming languages

###  Secure: It features bytecode verification, exception handling, robust security manager to prevent malicious.

# REQUIREMENTS AND ANALYSIS :

**Requirement Specification**

**Functional Requirements :**

These are the core functionalities the system must provide to fulfill the objectives of a Car Rental Management System.

**a. User Management**

* **Customer Registration and Authentication:**
  + Users should be able to create an account by entering personal details like name, email, phone number, etc.
  + Provide a secure login system with user authentication.
  + Admins should have a separate login to access management features.
  + Customers should be able to edit their personal details (name, contact information, etc.).
  + Admins can manage both customer and staff profiles.

**b. Vehicle Management**

* + Admins should be able to add new vehicles, update vehicle details (e.g., model, type, price, status), and remove vehicles from the fleet.
  + Each vehicle must have a status (e.g., available, rented, under maintenance).
  + Customers can search for vehicles by type, price range, model, location, etc.

**c. Reservation and Booking**

* + Customers should be able to reserve a vehicle for a specified period.
  + Customers should be able to modify or cancel their reservations before the booking date.
  + Customers should be able to view their past and upcoming bookings.

**Non-Functional Requirements**

Non-functional requirements are criteria that judge the system's operation, performance, and user experience.

**a. Performance**

* **Scalability**:
  + The system should handle increased loads as the business grows, managing more customers, vehicles, and bookings.
* **Response Time**:
  + The system should respond to user actions (e.g., search, booking) in less than 2 seconds.

**b. Usability**

* **User Interface (UI)**:
  + The UI should be intuitive and easy to navigate for both customers and administrators.
* **Mobile Compatibility**:
  + The system should be mobile-responsive, allowing users to make bookings and manage their accounts from smartphones and tablets.

**c. Security**

* **Data Protection**:
  + Sensitive data, such as personal information and payment details, should be encrypted and securely stored.
* **Authentication and Authorization**:
  + The system must ensure that only authorized users (admins, customers) can access specific sections.
* **Secure Payment Integration**:
  + The system must use secure payment gateways to process customer payments and prevent fraud.

**Hardware and Software Requirements :**

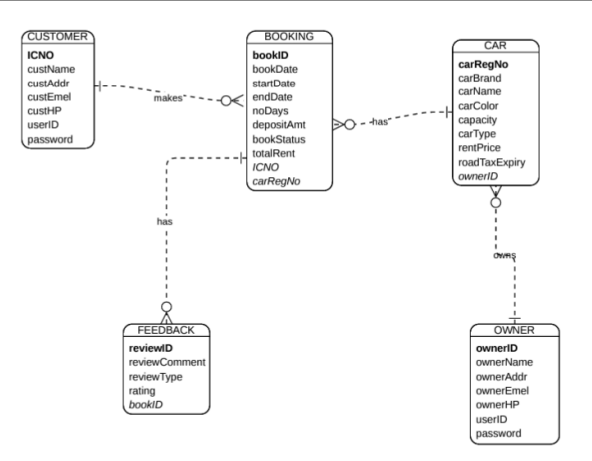
## Hardware :

* **Server:** A powerful server with sufficient CPU, RAM, and storage to handle the database and application workload.
* **Network:** A reliable network connection to allow access to the system from different locations.
* **POS Terminals:** Point-of-sale terminals for processing sales transactions.

## Software :

* **Database Management System (DBMS):** MySQL, PostgreSQL, or SQL Server or Mongodb.
* **Integrated Development Environment (IDE):** PyCharm or Visual Studio Code.
* **Operating System:** Linux or Windows.
* **Web Server:** Apache or Nginx.

**3.4 ER DIAGRAM** :



# PROGRAM CODE:

# /\*

# \* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

# \* Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template

# \*/

# package getingearfinal;

# import java.awt.Image;

# import java.io.ByteArrayOutputStream;

# import java.io.File;

# import java.io.FileInputStream;

# import java.sql.Statement;

# import java.sql.Connection;

# import java.sql.DriverManager;

# import java.sql.PreparedStatement;

# import java.sql.ResultSet;

# import java.sql.SQLException;

# import java.util.logging.Level;

# import java.util.logging.Logger;

# import javax.swing.ImageIcon;

# import javax.swing.JFileChooser;

# import javax.swing.JOptionPane;

# import javax.swing.table.DefaultTableModel;

# import net.proteanit.sql.DbUtils;

# /\*\*

# \*

# \* @author Akshay

# \*/

# public class Car extends javax.swing.JFrame {

# 

# public Car() {

# initComponents();

# DisplayCars();

# autoCarRegNum();

# 

# 

# 

# }

# /\*\*

# \* This method is called from within the constructor to initialize the form.

# \* WARNING: Do NOT modify this code. The content of this method is always

# \* regenerated by the Form Editor.

# \*/

# @SuppressWarnings("unchecked")

# // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents

# private void initComponents() {

# jPanel1 = new javax.swing.JPanel();

# jLabel1 = new javax.swing.JLabel();

# jLabel2 = new javax.swing.JLabel();

# jLabel3 = new javax.swing.JLabel();

# jLabel4 = new javax.swing.JLabel();

# jLabel14 = new javax.swing.JLabel();

# jLabel15 = new javax.swing.JLabel();

# jLabel16 = new javax.swing.JLabel();

# jLabel20 = new javax.swing.JLabel();

# jLabel17 = new javax.swing.JLabel();

# jPanel2 = new javax.swing.JPanel();

# jLabel8 = new javax.swing.JLabel();

# jLabel5 = new javax.swing.JLabel();

# jLabel6 = new javax.swing.JLabel();

# jLabel7 = new javax.swing.JLabel();

# jLabel9 = new javax.swing.JLabel();

# jLabel10 = new javax.swing.JLabel();

# jLabel11 = new javax.swing.JLabel();

# RegNumTb = new javax.swing.JTextField();

# PriceTb = new javax.swing.JTextField();

# BrandCb = new javax.swing.JComboBox<>();

# ResetBtn = new javax.swing.JButton();

# DeleteBtn = new javax.swing.JButton();

# EditBtn = new javax.swing.JButton();

# SaveBtn = new javax.swing.JButton();

# jLabel12 = new javax.swing.JLabel();

# jScrollPane1 = new javax.swing.JScrollPane();

# CarsTable = new javax.swing.JTable();

# ModelCb = new javax.swing.JComboBox<>();

# StatusCb = new javax.swing.JComboBox<>();

# setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

# setBackground(new java.awt.Color(250, 250, 250));

# setUndecorated(true);

# jPanel1.setBackground(new java.awt.Color(27, 38, 79));

# jLabel1.setFont(new java.awt.Font("Dubai", 0, 24)); // NOI18N

# jLabel1.setForeground(new java.awt.Color(255, 255, 255));

# jLabel1.setText("Customer");

# jLabel1.addMouseListener(new java.awt.event.MouseAdapter() {

# public void mouseClicked(java.awt.event.MouseEvent evt) {

# jLabel1MouseClicked(evt);

# }

# });

# jLabel2.setFont(new java.awt.Font("Dubai", 0, 24)); // NOI18N

# jLabel2.setForeground(new java.awt.Color(255, 255, 255));

# jLabel2.setText("Return Car");

# jLabel2.addMouseListener(new java.awt.event.MouseAdapter() {

# public void mouseClicked(java.awt.event.MouseEvent evt) {

# jLabel2MouseClicked(evt);

# }

# });

# jLabel3.setFont(new java.awt.Font("Dubai", 0, 24)); // NOI18N

# jLabel3.setForeground(new java.awt.Color(255, 255, 255));

# jLabel3.setText("Rent Car");

# jLabel3.addMouseListener(new java.awt.event.MouseAdapter() {

# public void mouseClicked(java.awt.event.MouseEvent evt) {

# jLabel3MouseClicked(evt);

# }

# });

# jLabel4.setFont(new java.awt.Font("Dubai", 0, 24)); // NOI18N

# jLabel4.setForeground(new java.awt.Color(255, 255, 255));

# jLabel4.setText("Logout");

# jLabel4.addMouseListener(new java.awt.event.MouseAdapter() {

# public void mouseClicked(java.awt.event.MouseEvent evt) {

# jLabel4MouseClicked(evt);

# }

# });

# jLabel14.setIcon(new javax.swing.ImageIcon(getClass().getResource("/getingearfinal/Untitled design (5) (2).png"))); // NOI18N

# jLabel15.setIcon(new javax.swing.ImageIcon(getClass().getResource("/getingearfinal/Untitled design (3) (1).png"))); // NOI18N

# jLabel16.setIcon(new javax.swing.ImageIcon(getClass().getResource("/getingearfinal/unknown3 (1).png"))); // NOI18N

# jLabel20.setIcon(new javax.swing.ImageIcon(getClass().getResource("/getingearfinal/unknown6 (1).jpg"))); // NOI18N

# jLabel17.setIcon(new javax.swing.ImageIcon(getClass().getResource("/getingearfinal/unknown1 (1).png"))); // NOI18N

# javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

# jPanel1.setLayout(jPanel1Layout);

# jPanel1Layout.setHorizontalGroup(

# jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

# .addGroup(jPanel1Layout.createSequentialGroup()

# .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

# .addGroup(jPanel1Layout.createSequentialGroup()

# .addContainerGap()

# .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

# .addGroup(jPanel1Layout.createSequentialGroup()

# .addGap(7, 7, 7)

# .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

# .addComponent(jLabel17, javax.swing.GroupLayout.Alignment.TRAILING)

# .addComponent(jLabel16, javax.swing.GroupLayout.Alignment.TRAILING)

# .addComponent(jLabel15)))

# .addComponent(jLabel14))

# .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

# .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

# .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 109, javax.swing.GroupLayout.PREFERRED\_SIZE)

# .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED\_SIZE, 109, javax.swing.GroupLayout.PREFERRED\_SIZE)

# .addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED\_SIZE, 109, javax.swing.GroupLayout.PREFERRED\_SIZE)

# .addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED\_SIZE, 109, javax.swing.GroupLayout.PREFERRED\_SIZE)))

# .addGroup(jPanel1Layout.createSequentialGroup()

# .addGap(68, 68, 68)

# .addComponent(jLabel20)))

# .addContainerGap(38, Short.MAX\_VALUE))

# );

# jPanel1Layout.setVerticalGroup(

# jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

# .addGroup(jPanel1Layout.createSequentialGroup()

# .addGap(93, 93, 93)

# .addComponent(jLabel20)

# .addGap(77, 77, 77)

# .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

# .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 28, javax.swing.GroupLayout.PREFERRED\_SIZE)

# .addComponent(jLabel16))

# .addGap(69, 69, 69)

# .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

# .addGroup(jPanel1Layout.createSequentialGroup()

# .addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED\_SIZE, 28, javax.swing.GroupLayout.PREFERRED\_SIZE)

# .addGap(77, 77, 77)

# .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

# .addGroup(jPanel1Layout.createSequentialGroup()

# .addComponent(jLabel15)

# .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

# .addComponent(jLabel14))

# .addGroup(jPanel1Layout.createSequentialGroup()

# .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED\_SIZE, 28, javax.swing.GroupLayout.PREFERRED\_SIZE)

# .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

# .addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED\_SIZE, 28, javax.swing.GroupLayout.PREFERRED\_SIZE)))

# .addGap(34, 34, 34))

# .addGroup(jPanel1Layout.createSequentialGroup()

# .addComponent(jLabel17)

# .addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))))

# );

# jLabel8.setFont(new java.awt.Font("Arial", 1, 18)); // NOI18N

# jLabel8.setForeground(new java.awt.Color(87, 108, 168));

# jLabel8.setText("X");

# javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);

# jPanel2.setLayout(jPanel2Layout);

# jPanel2Layout.setHorizontalGroup(

# jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

# .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel2Layout.createSequentialGroup()

# .addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

# .addComponent(jLabel8)

# .addContainerGap())

# );

# jPanel2Layout.setVerticalGroup(

# jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

# .addGroup(jPanel2Layout.createSequentialGroup()

# .addComponent(jLabel8)

# .addGap(0, 7, Short.MAX\_VALUE))

# );

# jLabel5.setFont(new java.awt.Font("Dubai", 0, 36)); // NOI18N

# jLabel5.setForeground(new java.awt.Color(27, 38, 79));

# jLabel5.setText("Manage Cars");

# jLabel6.setFont(new java.awt.Font("Dubai", 0, 24)); // NOI18N

# jLabel6.setForeground(new java.awt.Color(27, 38, 79));

# jLabel6.setText("Reg No.");

# jLabel7.setFont(new java.awt.Font("Dubai", 0, 24)); // NOI18N

# jLabel7.setForeground(new java.awt.Color(27, 38, 79));

# jLabel7.setText("Brand");

# jLabel9.setFont(new java.awt.Font("Dubai", 0, 24)); // NOI18N

# jLabel9.setForeground(new java.awt.Color(27, 38, 79));

# jLabel9.setText("Model");

# jLabel10.setFont(new java.awt.Font("Dubai", 0, 24)); // NOI18N

# jLabel10.setForeground(new java.awt.Color(27, 38, 79));

# jLabel10.setText("Status");

# jLabel11.setFont(new java.awt.Font("Dubai", 0, 24)); // NOI18N

# jLabel11.setForeground(new java.awt.Color(27, 38, 79));

# jLabel11.setText("Price/day");

# RegNumTb.setEditable(false);

# RegNumTb.addActionListener(new java.awt.event.ActionListener() {

# public void actionPerformed(java.awt.event.ActionEvent evt) {

# RegNumTbActionPerformed(evt);

# }

# });

# BrandCb.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "Maruti Suzuki", "Hyundai", "Tata", "Kia", "MG", "Honda", "Toyota", "Mercedes", "BMW", "Audi", "Rolls Royce" }));

# BrandCb.addActionListener(new java.awt.event.ActionListener() {

# public void actionPerformed(java.awt.event.ActionEvent evt) {

# BrandCbActionPerformed(evt);

# }

# });

# ResetBtn.setBackground(new java.awt.Color(39, 70, 144));

# ResetBtn.setFont(new java.awt.Font("Dubai", 0, 18)); // NOI18N

# ResetBtn.setForeground(new java.awt.Color(255, 255, 255));

# ResetBtn.setText("Reset");

# ResetBtn.addActionListener(new java.awt.event.ActionListener() {

# public void actionPerformed(java.awt.event.ActionEvent evt) {

# ResetBtnActionPerformed(evt);

# }

# });

# DeleteBtn.setBackground(new java.awt.Color(39, 70, 144));

# DeleteBtn.setFont(new java.awt.Font("Dubai", 0, 18)); // NOI18N

# DeleteBtn.setForeground(new java.awt.Color(255, 255, 255));

# DeleteBtn.setText("Delete");

# DeleteBtn.addActionListener(new java.awt.event.ActionListener() {

# public void actionPerformed(java.awt.event.ActionEvent evt) {

# DeleteBtnActionPerformed(evt);

# }

# });

# EditBtn.setBackground(new java.awt.Color(39, 70, 144));

# EditBtn.setFont(new java.awt.Font("Dubai", 0, 18)); // NOI18N

# EditBtn.setForeground(new java.awt.Color(255, 255, 255));

# EditBtn.setText("Edit");

# EditBtn.addActionListener(new java.awt.event.ActionListener() {

# public void actionPerformed(java.awt.event.ActionEvent evt) {

# EditBtnActionPerformed(evt);

# }

# });

# SaveBtn.setBackground(new java.awt.Color(39, 70, 144));

# SaveBtn.setFont(new java.awt.Font("Dubai", 0, 18)); // NOI18N

# SaveBtn.setForeground(new java.awt.Color(255, 255, 255));

# SaveBtn.setText("Save");

# SaveBtn.addActionListener(new java.awt.event.ActionListener() {

# public void actionPerformed(java.awt.event.ActionEvent evt) {

# SaveBtnActionPerformed(evt);

# }

# });

# jLabel12.setFont(new java.awt.Font("Dubai", 0, 24)); // NOI18N

# jLabel12.setForeground(new java.awt.Color(27, 38, 79));

# jLabel12.setText("Car List");

# CarsTable.setModel(new javax.swing.table.DefaultTableModel(

# new Object [][] {

# {null, null, null, null, null}

# },

# new String [] {

# "Reg No.", "Brand", "Model", "Status", "Price"

# }

# ));

# CarsTable.setRowHeight(23);

# CarsTable.addMouseListener(new java.awt.event.MouseAdapter() {

# public void mouseClicked(java.awt.event.MouseEvent evt) {

# CarsTableMouseClicked(evt);

# }

# });

# jScrollPane1.setViewportView(CarsTable);

# ModelCb.addActionListener(new java.awt.event.ActionListener() {

# public void actionPerformed(java.awt.event.ActionEvent evt) {

# ModelCbActionPerformed(evt);

# }

# });

# StatusCb.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "Available", "Booked" }));

# StatusCb.addActionListener(new java.awt.event.ActionListener() {

# public void actionPerformed(java.awt.event.ActionEvent evt) {

# StatusCbActionPerformed(evt);

# }

# });

# javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

# getContentPane().setLayout(layout);

# layout.setHorizontalGroup(

# layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

# .addGroup(layout.createSequentialGroup()

# .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

# .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

# .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

# .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

# .addComponent(jLabel5)

# .addGap(322, 322, 322))

# .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

# .addGap(18, 18, 18)

# .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

# .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

# .addGroup(layout.createSequentialGroup()

# .addGap(19, 19, 19)

# .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

# .addGroup(layout.createSequentialGroup()

# .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

# .addComponent(jLabel6)

# .addComponent(RegNumTb, javax.swing.GroupLayout.PREFERRED\_SIZE, 99, javax.swing.GroupLayout.PREFERRED\_SIZE))

# .addGap(66, 66, 66)

# .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

# .addComponent(jLabel7)

# .addComponent(BrandCb, javax.swing.GroupLayout.PREFERRED\_SIZE, 131, javax.swing.GroupLayout.PREFERRED\_SIZE))

# .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

# .addGroup(layout.createSequentialGroup()

# .addGap(99, 99, 99)

# .addComponent(SaveBtn)

# .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

# .addComponent(EditBtn)

# .addGap(3, 3, 3)))

# .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

# .addComponent(jLabel9)

# .addGroup(layout.createSequentialGroup()

# .addGap(98, 98, 98)

# .addComponent(DeleteBtn))

# .addGroup(layout.createSequentialGroup()

# .addGap(17, 17, 17)

# .addComponent(jLabel12))

# .addComponent(ModelCb, javax.swing.GroupLayout.PREFERRED\_SIZE, 122, javax.swing.GroupLayout.PREFERRED\_SIZE))

# .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

# .addGroup(layout.createSequentialGroup()

# .addGap(16, 16, 16)

# .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

# .addComponent(jLabel10)

# .addComponent(StatusCb, javax.swing.GroupLayout.PREFERRED\_SIZE, 92, javax.swing.GroupLayout.PREFERRED\_SIZE))

# .addGap(82, 82, 82)

# .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

# .addComponent(jLabel11)

# .addComponent(PriceTb, javax.swing.GroupLayout.PREFERRED\_SIZE, 99, javax.swing.GroupLayout.PREFERRED\_SIZE))

# .addGap(0, 0, Short.MAX\_VALUE))

# .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

# .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

# .addComponent(ResetBtn)

# .addGap(111, 111, 111))))

# .addGroup(layout.createSequentialGroup()

# .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 844, javax.swing.GroupLayout.PREFERRED\_SIZE)

# .addGap(0, 0, Short.MAX\_VALUE)))

# .addContainerGap())))

# );

# layout.setVerticalGroup(

# layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

# .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

# .addGroup(layout.createSequentialGroup()

# .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

# .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

# .addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED\_SIZE, 30, javax.swing.GroupLayout.PREFERRED\_SIZE)

# .addGap(25, 25, 25)

# .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

# .addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED\_SIZE, 28, javax.swing.GroupLayout.PREFERRED\_SIZE)

# .addComponent(jLabel7, javax.swing.GroupLayout.PREFERRED\_SIZE, 28, javax.swing.GroupLayout.PREFERRED\_SIZE)

# .addComponent(jLabel9, javax.swing.GroupLayout.PREFERRED\_SIZE, 28, javax.swing.GroupLayout.PREFERRED\_SIZE)

# .addComponent(jLabel10, javax.swing.GroupLayout.PREFERRED\_SIZE, 28, javax.swing.GroupLayout.PREFERRED\_SIZE)

# .addComponent(jLabel11, javax.swing.GroupLayout.PREFERRED\_SIZE, 28, javax.swing.GroupLayout.PREFERRED\_SIZE))

# .addGap(18, 18, 18)

# .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

# .addComponent(RegNumTb, javax.swing.GroupLayout.PREFERRED\_SIZE, 31, javax.swing.GroupLayout.PREFERRED\_SIZE)

# .addComponent(PriceTb, javax.swing.GroupLayout.PREFERRED\_SIZE, 31, javax.swing.GroupLayout.PREFERRED\_SIZE)

# .addComponent(BrandCb, javax.swing.GroupLayout.PREFERRED\_SIZE, 32, javax.swing.GroupLayout.PREFERRED\_SIZE)

# .addComponent(ModelCb, javax.swing.GroupLayout.PREFERRED\_SIZE, 32, javax.swing.GroupLayout.PREFERRED\_SIZE)

# .addComponent(StatusCb, javax.swing.GroupLayout.PREFERRED\_SIZE, 32, javax.swing.GroupLayout.PREFERRED\_SIZE))

# .addGap(18, 18, 18)

# .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

# .addComponent(SaveBtn, javax.swing.GroupLayout.PREFERRED\_SIZE, 32, javax.swing.GroupLayout.PREFERRED\_SIZE)

# .addComponent(EditBtn, javax.swing.GroupLayout.PREFERRED\_SIZE, 32, javax.swing.GroupLayout.PREFERRED\_SIZE)

# .addComponent(DeleteBtn, javax.swing.GroupLayout.PREFERRED\_SIZE, 32, javax.swing.GroupLayout.PREFERRED\_SIZE)

# .addComponent(ResetBtn, javax.swing.GroupLayout.PREFERRED\_SIZE, 32, javax.swing.GroupLayout.PREFERRED\_SIZE))

# .addGap(31, 31, 31)

# .addComponent(jLabel12, javax.swing.GroupLayout.PREFERRED\_SIZE, 28, javax.swing.GroupLayout.PREFERRED\_SIZE)

# .addGap(18, 18, 18)

# .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

# .addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

# );

# pack();

# setLocationRelativeTo(null);

# }// </editor-fold>//GEN-END:initComponents

# Connection Con = null;

# Statement St = null;

# ResultSet Rs = null;

# private void DisplayCars()

# {

# try{

# Con = DriverManager.getConnection("jdbc:derby://localhost:1527/CarDb","root","root");

# St = Con.createStatement();

# Rs = St.executeQuery("select \* from CARTBL");

# CarsTable.setModel(DbUtils.resultSetToTableModel(Rs));

# }catch(SQLException e)

# {

# e.printStackTrace();

# }

# 

# }

# public void autoCarRegNum()

# {

# try{

# Con = DriverManager.getConnection("jdbc:derby://localhost:1527/CarDb","root","root");

# St = Con.createStatement();

# Rs = St.executeQuery("select max(carreg) from CARTBL");

# 

# int lastid;

# if(Rs.next())

# {

# lastid = Rs.getInt(1);

# lastid++;

# RegNumTb.setText(Integer.toString(lastid));

# }

# 

# 

# }catch(SQLException e){

# e.printStackTrace();

# }

# }

# 

# private void BrandCbActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_BrandCbActionPerformed

# if(BrandCb.getSelectedItem().equals("Maruti Suzuki"))

# {

# ModelCb.removeAllItems();

# 

# ModelCb.addItem("Alto");

# ModelCb.addItem("WagonR");

# ModelCb.addItem("Swift");

# ModelCb.addItem("Baleno");

# ModelCb.addItem("Dzire");

# ModelCb.addItem("Ciaz");

# ModelCb.addItem("Vitara Brezza");

# ModelCb.addItem("S-Cross");

# ModelCb.addItem("Ertiga");

# ModelCb.setSelectedItem(null);

# }

# else

# if(BrandCb.getSelectedItem().equals("Hyundai"))

# {

# ModelCb.removeAllItems();

# 

# ModelCb.addItem("Santro");

# ModelCb.addItem("i10");

# ModelCb.addItem("i20");

# ModelCb.addItem("Verna");

# ModelCb.addItem("Venue");

# ModelCb.addItem("Creta");

# ModelCb.setSelectedItem(null);

# }

# else

# if(BrandCb.getSelectedItem().equals("Tata"))

# {

# ModelCb.removeAllItems();

# 

# ModelCb.addItem("Tiago");

# ModelCb.addItem("Tigor");

# ModelCb.addItem("Punch");

# ModelCb.addItem("Nexon");

# ModelCb.addItem("Harrier");

# ModelCb.addItem("Safari");

# ModelCb.setSelectedItem(null);

# }

# else

# if(BrandCb.getSelectedItem().equals("Kia"))

# {

# ModelCb.removeAllItems();

# 

# ModelCb.addItem("Sonet");

# ModelCb.addItem("Seltos");

# ModelCb.addItem("Carnival");

# ModelCb.setSelectedItem(null);

# 

# }

# else

# if(BrandCb.getSelectedItem().equals("MG"))

# {

# ModelCb.removeAllItems();

# 

# ModelCb.addItem("Hector");

# ModelCb.addItem("Gloster");

# ModelCb.setSelectedItem(null);

# 

# }

# 

# else

# if(BrandCb.getSelectedItem().equals("Honda"))

# {

# ModelCb.removeAllItems();

# 

# ModelCb.addItem("Jazz");

# ModelCb.addItem("Amaze");

# ModelCb.addItem("City");

# ModelCb.addItem("Civic");

# ModelCb.addItem("WR-V");

# //ModelCb.addItem("Safari");

# ModelCb.setSelectedItem(null);

# }

# else

# if(BrandCb.getSelectedItem().equals("Toyota"))

# {

# ModelCb.removeAllItems();

# 

# ModelCb.addItem("Glanza");

# ModelCb.addItem("Yaris");

# ModelCb.addItem("Innova");

# ModelCb.addItem("Innova Crysta");

# ModelCb.addItem("Fortuner");

# ModelCb.setSelectedItem(null);

# //ModelCb.addItem("Safari");

# }

# else

# if(BrandCb.getSelectedItem().equals("Mercedes"))

# {

# ModelCb.removeAllItems();

# 

# ModelCb.addItem("C-class");

# ModelCb.addItem("E-class");

# ModelCb.addItem("S-class");

# ModelCb.addItem("GLA");

# ModelCb.addItem("GLC");

# ModelCb.addItem("GLE");

# ModelCb.addItem("GLS");

# ModelCb.setSelectedItem(null);

# }

# else

# if(BrandCb.getSelectedItem().equals("BMW"))

# {

# ModelCb.removeAllItems();

# 

# ModelCb.addItem("3 series");

# ModelCb.addItem("5 series");

# ModelCb.addItem("7 series");

# ModelCb.addItem("X5");

# ModelCb.addItem("X7");

# ModelCb.addItem("iX");

# ModelCb.setSelectedItem(null);

# }

# else

# if(BrandCb.getSelectedItem().equals("Audi"))

# {

# ModelCb.removeAllItems();

# 

# ModelCb.addItem("A4");

# ModelCb.addItem("A6");

# ModelCb.addItem("Q5");

# ModelCb.addItem("Q7");

# ModelCb.addItem("Q8");

# ModelCb.addItem("E-Tron");

# ModelCb.setSelectedItem(null);

# }

# else

# if(BrandCb.getSelectedItem().equals("Rolls Royce"))

# {

# ModelCb.removeAllItems();

# 

# ModelCb.addItem("Phantom");

# ModelCb.addItem("Dawn");

# ModelCb.addItem("Wraith");

# ModelCb.addItem("Ghost");

# ModelCb.addItem("Cullinan");

# ModelCb.setSelectedItem(null);

# 

# }

# }//GEN-LAST:event\_BrandCbActionPerformed

# private void RegNumTbActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_RegNumTbActionPerformed

# // TODO add your handling code here:

# }//GEN-LAST:event\_RegNumTbActionPerformed

# private void SaveBtnActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_SaveBtnActionPerformed

# if(RegNumTb.getText().isEmpty() || PriceTb.getText().isEmpty() || BrandCb.getSelectedIndex() == -1 || ModelCb.getSelectedIndex() == -1 || StatusCb.getSelectedIndex() == -1)

# {

# JOptionPane.showMessageDialog(this, "Missing Information");

# }else{

# try {

# Con = DriverManager.getConnection("jdbc:derby://localhost:1527/CarDb","root","root");

# PreparedStatement add = Con.prepareStatement("insert into CARTBL values (?,?,?,?,?)");

# add.setString(1, RegNumTb.getText());

# add.setString(2, BrandCb.getSelectedItem().toString());

# add.setString(3, ModelCb.getSelectedItem().toString());

# add.setString(4, StatusCb.getSelectedItem().toString());

# add.setInt(5, Integer.valueOf(PriceTb.getText()));

# 

# int row = add.executeUpdate();

# JOptionPane.showMessageDialog(this, "Car added Successfully");

# DisplayCars();

# autoCarRegNum();

# 

# 

# PriceTb.setText("");

# } catch (Exception e) {

# e.printStackTrace();

# }}

# }//GEN-LAST:event\_SaveBtnActionPerformed

# private void DeleteBtnActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_DeleteBtnActionPerformed

# if(RegNumTb.getText().isEmpty())

# {

# JOptionPane.showMessageDialog(this, "Select the Car to be Deleted");

# }else{

# try {

# Con = DriverManager.getConnection("jdbc:derby://localhost:1527/CarDb","root","root");

# String Reg = RegNumTb.getText();

# String Query = "Delete from root.CARTBL where CarReg='"+Reg+"'";

# Statement Add = Con.createStatement();

# Add.executeUpdate(Query);

# JOptionPane.showMessageDialog(this, "Car deleted Successfully");

# DisplayCars();

# Reset();

# autoCarRegNum();

# 

# } catch (Exception e) {

# e.printStackTrace();

# }}

# }//GEN-LAST:event\_DeleteBtnActionPerformed

# private void CarsTableMouseClicked(java.awt.event.MouseEvent evt) {//GEN-FIRST:event\_CarsTableMouseClicked

# DefaultTableModel model = (DefaultTableModel)CarsTable.getModel();

# int MyIndex = CarsTable.getSelectedRow();

# RegNumTb.setText(model.getValueAt(MyIndex, 0).toString());

# BrandCb.setSelectedItem(model.getValueAt(MyIndex, 1).toString());

# ModelCb.setSelectedItem(model.getValueAt(MyIndex, 2).toString());

# StatusCb.setSelectedItem(model.getValueAt(MyIndex, 3).toString());

# PriceTb.setText(model.getValueAt(MyIndex, 4).toString());

# 

# }//GEN-LAST:event\_CarsTableMouseClicked

# private void Reset()

# {

# RegNumTb.setText("");

# BrandCb.setSelectedIndex(0);

# ModelCb.setSelectedIndex(0);

# StatusCb.setSelectedIndex(0);

# PriceTb.setText("");

# }

# private void ResetBtnActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_ResetBtnActionPerformed

# Reset();

# }//GEN-LAST:event\_ResetBtnActionPerformed

# private void EditBtnActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_EditBtnActionPerformed

# if(RegNumTb.getText().isEmpty() || PriceTb.getText().isEmpty() || BrandCb.getSelectedIndex() == -1 || ModelCb.getSelectedIndex() == -1 || StatusCb.getSelectedIndex() == -1)

# {

# JOptionPane.showMessageDialog(this, "Select the Car to be Updated");

# }else{

# try {

# Con = DriverManager.getConnection("jdbc:derby://localhost:1527/CarDb","root","root");

# String Reg = RegNumTb.getText();

# String Query = "Update root.CARTBL set Brand='"+BrandCb.getSelectedItem().toString()+"',Model='"+ModelCb.getSelectedItem().toString()+"',Status='"+StatusCb.getSelectedItem().toString()+"',Price="+PriceTb.getText()+" where CarReg='"+Reg+"'";

# Statement Add = Con.createStatement();

# Add.executeUpdate(Query);

# JOptionPane.showMessageDialog(this, "Car Updated Successfully");

# DisplayCars();

# Reset();

# autoCarRegNum();

# 

# } catch (Exception e) {

# e.printStackTrace();

# }}

# }//GEN-LAST:event\_EditBtnActionPerformed

# private void jLabel1MouseClicked(java.awt.event.MouseEvent evt) {//GEN-FIRST:event\_jLabel1MouseClicked

# new Customer().setVisible(true);

# this.dispose();

# }//GEN-LAST:event\_jLabel1MouseClicked

# private void jLabel3MouseClicked(java.awt.event.MouseEvent evt) {//GEN-FIRST:event\_jLabel3MouseClicked

# new Rents().setVisible(true);

# this.dispose();

# }//GEN-LAST:event\_jLabel3MouseClicked

# private void jLabel2MouseClicked(java.awt.event.MouseEvent evt) {//GEN-FIRST:event\_jLabel2MouseClicked

# new Return().setVisible(true);

# this.dispose();

# }//GEN-LAST:event\_jLabel2MouseClicked

# private void jLabel4MouseClicked(java.awt.event.MouseEvent evt) {//GEN-FIRST:event\_jLabel4MouseClicked

# new Login().setVisible(true);

# this.dispose();

# }//GEN-LAST:event\_jLabel4MouseClicked

# private void ModelCbActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_ModelCbActionPerformed

# // TODO add your handling code here:

# }//GEN-LAST:event\_ModelCbActionPerformed

# private void StatusCbActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_StatusCbActionPerformed

# // TODO add your handling code here:

# }//GEN-LAST:event\_StatusCbActionPerformed

# /\*\*

# \* @param args the command line arguments

# \*/

# public static void main(String args[]) {

# /\* Set the Nimbus look and feel \*/

# //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

# /\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

# \* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

# \*/

# try {

# for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

# if ("Nimbus".equals(info.getName())) {

# javax.swing.UIManager.setLookAndFeel(info.getClassName());

# break;

# }

# }

# } catch (ClassNotFoundException ex) {

# java.util.logging.Logger.getLogger(Car.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

# } catch (InstantiationException ex) {

# java.util.logging.Logger.getLogger(Car.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

# } catch (IllegalAccessException ex) {

# java.util.logging.Logger.getLogger(Car.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

# } catch (javax.swing.UnsupportedLookAndFeelException ex) {

# java.util.logging.Logger.getLogger(Car.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

# }

# //</editor-fold>

# /\* Create and display the form \*/

# java.awt.EventQueue.invokeLater(new Runnable() {

# public void run() {

# new Car().setVisible(true);

# }

# });

# }

# // Variables declaration - do not modify//GEN-BEGIN:variables

# private javax.swing.JComboBox<String> BrandCb;

# private javax.swing.JTable CarsTable;

# private javax.swing.JButton DeleteBtn;

# private javax.swing.JButton EditBtn;

# private javax.swing.JComboBox<String> ModelCb;

# private javax.swing.JTextField PriceTb;

# private javax.swing.JTextField RegNumTb;

# private javax.swing.JButton ResetBtn;

# private javax.swing.JButton SaveBtn;

# private javax.swing.JComboBox<String> StatusCb;

# private javax.swing.JLabel jLabel1;

# private javax.swing.JLabel jLabel10;

# private javax.swing.JLabel jLabel11;

# private javax.swing.JLabel jLabel12;

# private javax.swing.JLabel jLabel14;

# private javax.swing.JLabel jLabel15;

# private javax.swing.JLabel jLabel16;

# private javax.swing.JLabel jLabel17;

# private javax.swing.JLabel jLabel2;

# private javax.swing.JLabel jLabel20;

# private javax.swing.JLabel jLabel3;

# private javax.swing.JLabel jLabel4;

# private javax.swing.JLabel jLabel5;

# private javax.swing.JLabel jLabel6;

# private javax.swing.JLabel jLabel7;

# private javax.swing.JLabel jLabel8;

# private javax.swing.JLabel jLabel9;

# private javax.swing.JPanel jPanel1;

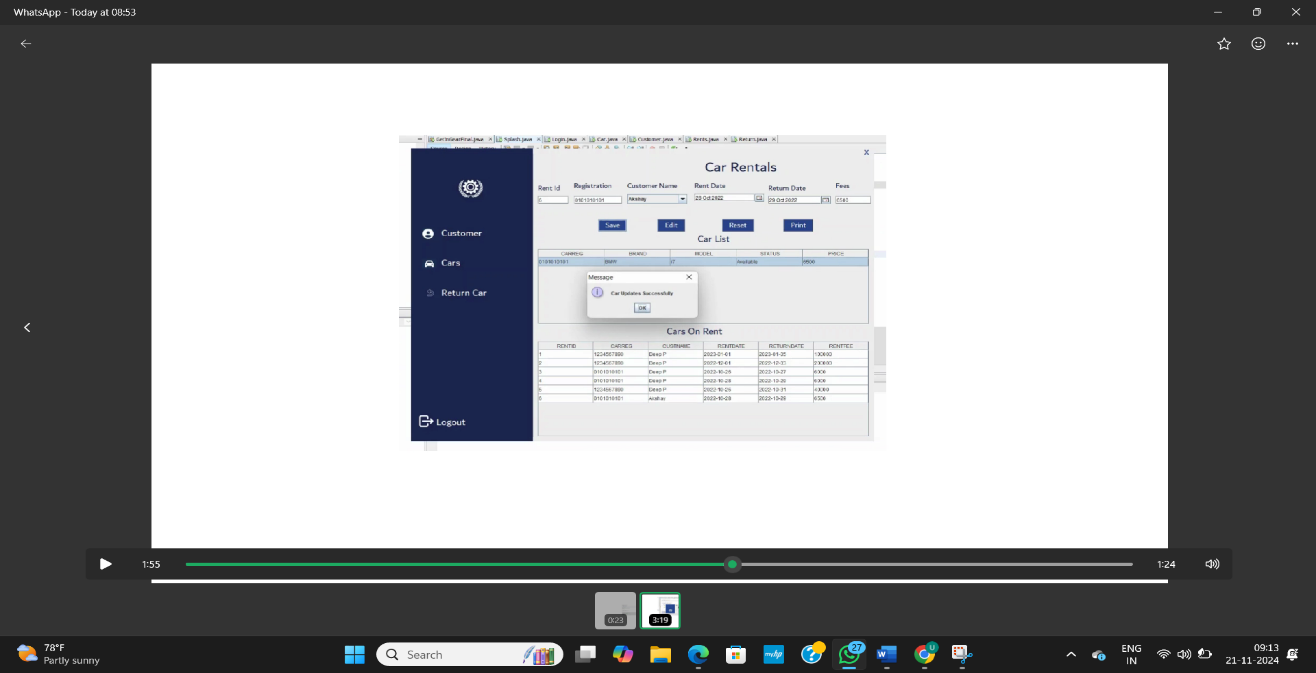
# private javax.swing.JPanel jPanel2;

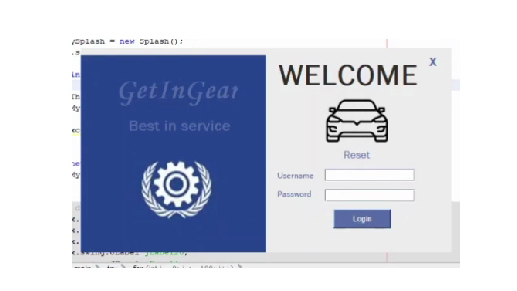
# private javax.swing.JScrollPane jScrollPane1;

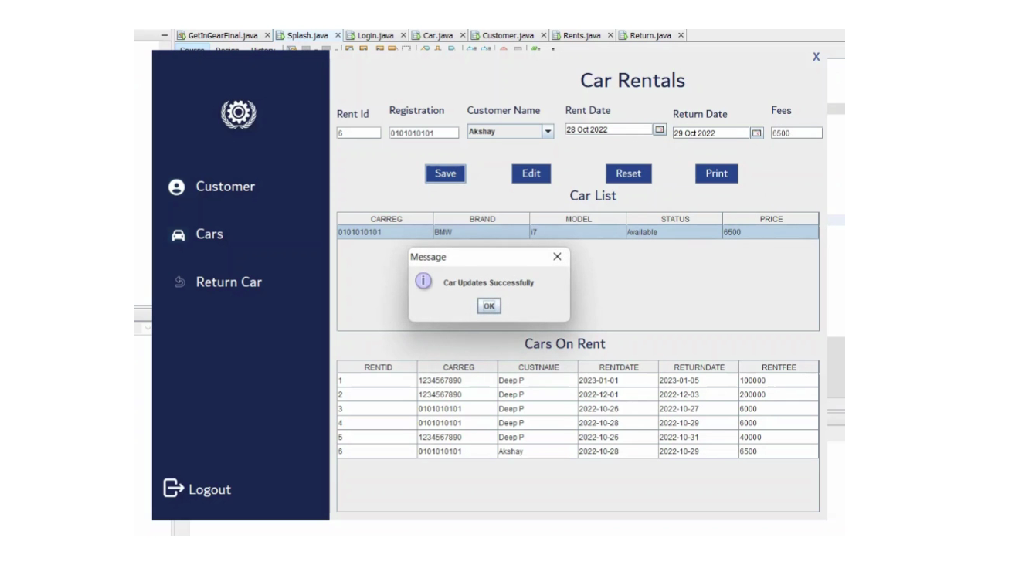
# // End of variables declaration//GEN-END:variables

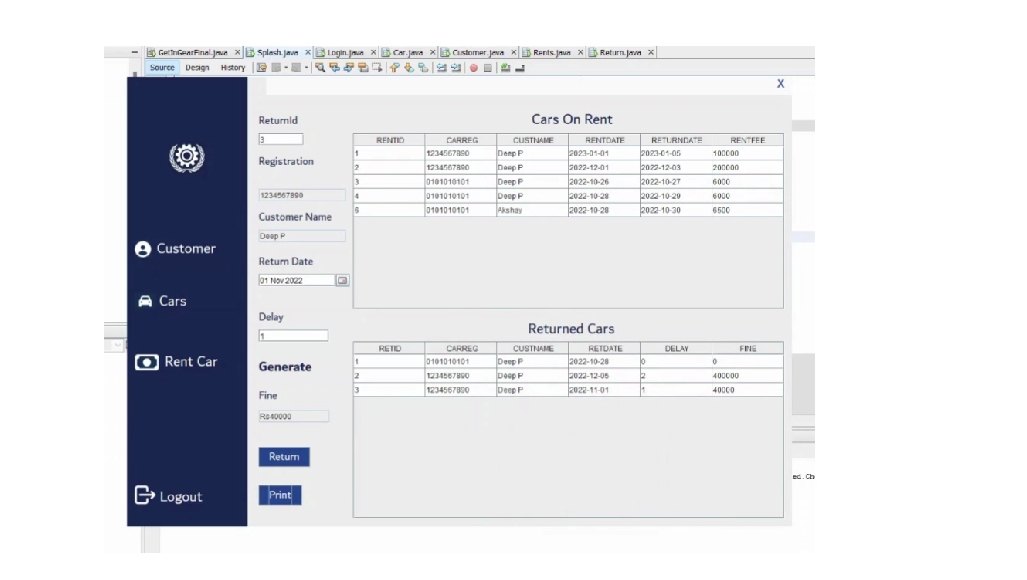
# }

**Result Screenshot**









# CONCLUSION:

In conclusion, the **Car Rental Management System** is an essential tool for automating and streamlining the operations of a car rental business. It allows for efficient management of vehicle inventory, customer data, reservations, payments, and reporting. By integrating modern technologies and following best practices in database management, security, and user interface design, the system ensures a seamless experience for both customers and administrators.

The system’s core functionalities, such as booking management, payment processing, vehicle maintenance tracking, and customer support, offer comprehensive solutions to the challenges faced by car rental companies. Additionally, its non-functional requirements, including performance, security, and scalability, ensure that the system can handle growing demands while maintaining reliability and user satisfaction.

the Car Rental Management System helps businesses optimize their operations, improve customer service, and increase revenue through efficient management of resources. By adopting this system, car rental companies can achieve better control over their operations and deliver a higher-quality service to their customers.

# REFERENCES :

 **SQL Database Concepts and Best Practices**

* *Reference*: "SQL for Web Developers"
* https://www.w3schools.com/

 **Flask Framework Documentation**

* *Reference*: Flask Documentation (Official Site)
* https://flask.palletsprojects.com/en/stable/

 **Car rental Management System Design and Best Practices**

* *Reference*: "Best Practices for car rental Management," TradeGecko Blog.
* https://en.wikipedia.org/wiki/TradeGecko

 **ORMs and Flask Database Integration**

* *Reference*: "Flask SQLAlchemy – Using SQLAlchemy with Flask," Real Python.
* https://www.python.org/