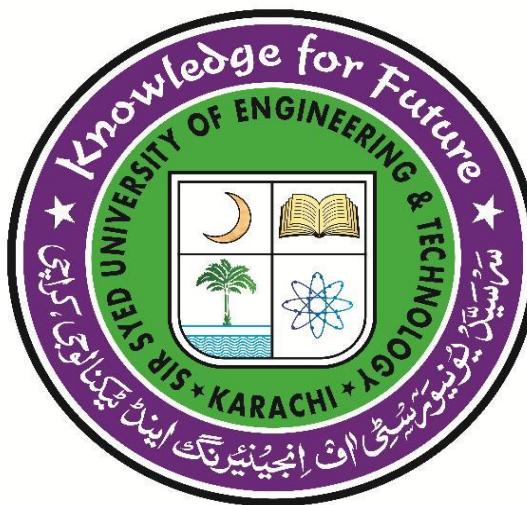


PROJECT REPORT

Database Management System

(CE-220)

Fall Semester 2022



Submitted By:

| | |
|--------------------------------------|-----------------------|
| Student Name 1: Adil Aijaz | 2 0 2 0 - C E - 0 9 5 |
| Student Name 2: Ramish Masood | 2 0 2 0 - C E - 0 8 6 |
| Student Name 3: Zahir Khan | 2 0 2 0 - C E - 1 2 1 |
| Student Name 4: Qasim | 2 0 2 0 - C E - 1 1 3 |

Submitted To:

Lab & Course Teacher: Tehmina Nishat

Computer Engineering Department
Sir Syed University of Engineering & Technology
University Road, Karachi 75300
<http://www.ssuet.edu.pk>

SYSTEM CAR SALES

| SUV |



| Hatchback |



| Sedan |



Table Of Contents

| <u>S NO</u> | <u>CONTENTS</u> | <u>PAGE NO</u> |
|-------------|----------------------------|----------------|
| 1 | Acknowledgement | 4 |
| 2 | Project Overview | 5 |
| 3 | Major Features | 6-7 |
| 4 | Project Description | 8 |
| 5 | ER/EER Diagram | 9 |
| 6 | Mapping of Diagram | 10-13 |
| 7 | Source Code | 14-41 |
| 8 | Output | 42-61 |
| 8 | Conclusion | 62 |
| 9 | Future Scope | 63 |

ACKNOWLEDGEMENT

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend my sincere thanks to all of them. We are highly indebted to teachers for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

We would like to express my gratitude towards members of project for their kind co-operation and encouragement which help me in completion of this project

We would like to express my special gratitude and thanks to industry persons for giving me such attention and time.

Our thanks and appreciations also go to our group members in developing the project, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project. We would like to show our greatest appreciation to lab teacher Miss Ayesha Majeed and Course teacher Miss Kiran. we can't say thank you enough for their tremendous support and help. we feel motivated and encouraged every time. Without their encouragement and guidance this project would not have materialized. The guidance and support received from all the members who contributed and who are contributing to this project, was vital for the success of the project. We are grateful for their constant support and help

PROJECT OVERVIEW

TITLE OF THE PROJECT:

TOYOTA CARS SALES SYSTEM

This is a simple project on Car Sales programmed in JAVA. The name we suggested “Toyota Car Sales System”. The purpose of the project is to build an application program to reduce the manual work for managing Car Models, Car Sales, Car Details, Car Search etc. This project provides searching facilities based on various factors such as Car Models, Car information etc using my SQL database. This intern increases the efficiency of managing various Car Models and improved Sales



In the existing System it is difficult to maintain the car information individually and to supply for the customers who are eager to buy them. Customer must face difficulty to know the information of car like manufacturing year, car model and other valuable information in a single domain. Our main idea is to develop a database system where we can have all the required information for the user to effectively interest him in the process of buying a car

Major Feature

REGISTRATION OF A CUSTOMER:

The customer will Create his Account from which he can purchase the car.

HELP:

If Customer need any type of help regarding car purchasing or company information, then he will be entertained through auto generated answers

SUBSCRIPTION:

The customer can get the latest information by subscribing by using his email.

FILE A COMPLAINT:

The customer can file a complaint letter to the company if he faces some problem.

VIEWING ALL CAR DETAILS:

Each car is shown in JTable whose data is directly coming from sql database. It consists of full car details consist of Car name, Car Code, model, Engine Displacement, Engine Power, Fuel-Type , Gear-Box, Transmission-Type, Seating Capacity, top speed, price, variant.

NAVIGATION OF CARS:

Admin will navigate through records of car with the help of "First", "Next", "Previous", and "Last" button.

ADDING/INSERT MORE CARS:

Using this feature, we can add new cars from admin window which are for sale which are displayed in customer window. This feature will also help customer to add the car in JTable which he wants to purchase.

DELETION OF CAR:

Delete/Remove feature will help admin to delete the Selected car by car's code. Delete/Remove feature will help Customer to delete the selected car which he mistakenly selected.

CLEAR TABLE:

Admin can clear the text Field and Customer can Clear the JTable of his selected cars.

UPDATING OF CAR:

Admin Can update the selecting car, he can renew the price , its Engine and top speed etc.

SEARCHING FOR REQUIRED CAR:

Search feature will help admin to find car which he can add/delete or can update

CUSTOMIZED COLOR SELECTION:

Customized feature will help Customer to select the color of his purchased car by his own.

CUSTOMIZED ACCESSORIES SELECTION:

Customized feature will help Customer to select the HID lights and Alloy Rims of his newly purchased car.

PRINT BILLING:

When customer clicks the "Purchase" button, the billing window will appear in which customer can print his bill in pdf or directly print it from printer.

SAVING RECORD OF BILLS:

When Customer purchases some car, then the record of that car will appear in admin window and admin will view all the purchased cars of the customer.

CLEAR RECORD OF BILLS:

Admin can clear all the previous record of bill of the customer purchased car. Logout ID: Customer and Admin can Logout their ID and will Jump to sign-In window

Project Description

In this project, we created a program that titled Car Sales System. In this program, we started by making a package called Cars Sales System, first we created sign-In form window from which Customer and Admin, both can Login to their respective windows. If customer does not have his account, then he will have to create his account through registration window, that window will require all the basic information about the customer and all his basic data will be saved in the SQL database. In the same window, the can customer accesses the help button in case If Customer need any type of help regarding car purchasing or company information then he will be entertained through auto generated answers. The customer can also subscribe the Toyota car sales system to get latest updates and if he faces some kind of problem he can also file a complaint.

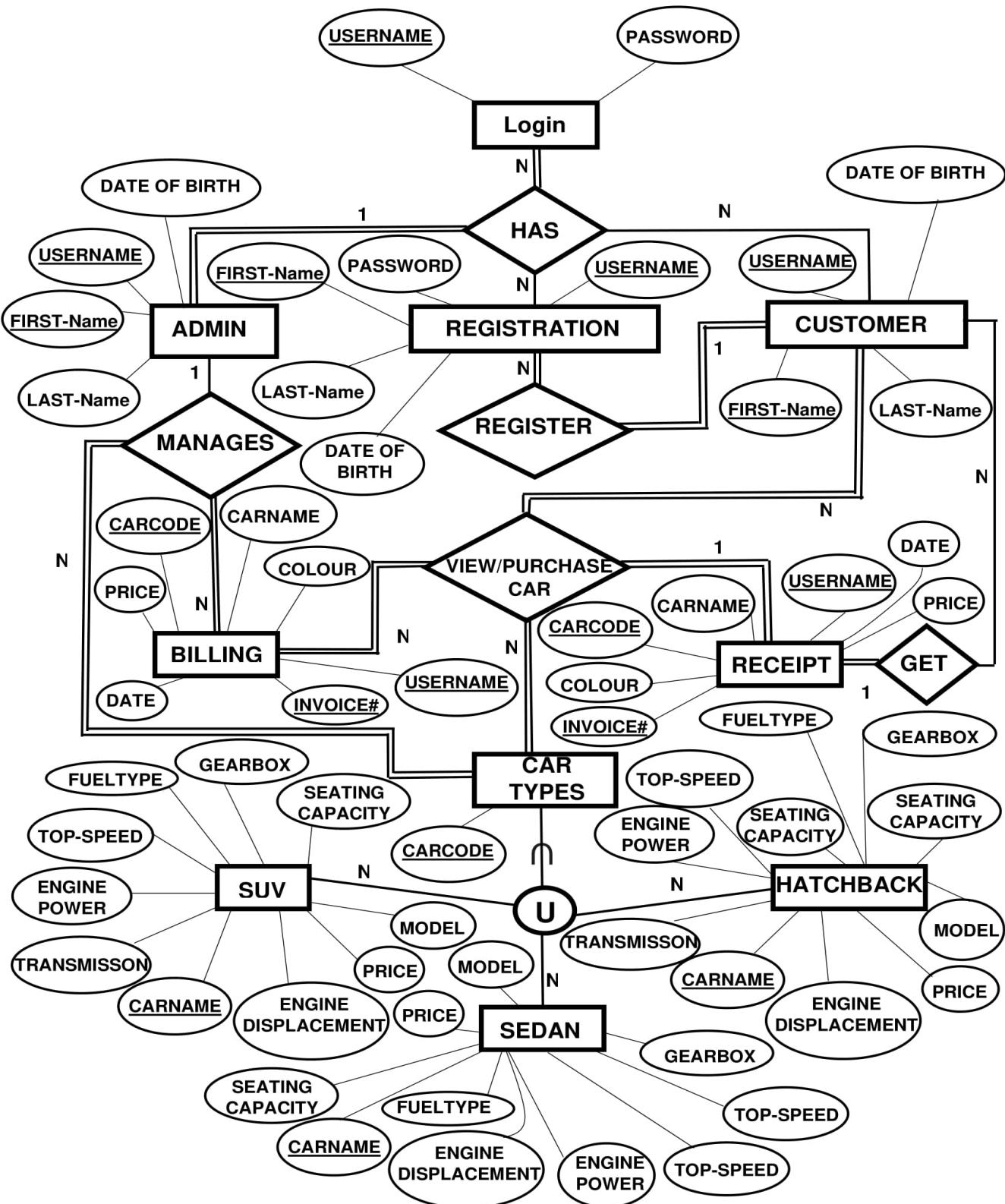
Now let's talk about customer window. In this window the customer can view all types of cars from Toyota in JTable directly coming from SQL database i.e. "HATCHBACK", "SEDAN", "SUV". Then customer needs to choose color of his car and accessories of his car, and add his favorite car to another JTable through add button , the customer can also delete the car which he accidentally added, so , after clicking the add button, the Credit card information window will appear, the customer had to enter his credit card information for the further process, after Customer click checkout button, the Customer's bill will appears, the costumer can print the bill via pdf or printer. Now after successful purchase the costumer can logout his account through logout button which will take the customer to the Sign-In window

Now let's look at the at the administrator window, in this JFrame the admin must select the car types through combo box and then he can add new cars to the database, which customer can view in his window as well. The admin can delete the cars from SQL database which he wants. He can update the cars; the admin can also navigate through records of cars and search cars through car code. He can also view the bills of the costumer. And can clear the records of bills in another JFrame. So, this was our total project which revolves around customer and administrator

ER

DIAGRAM

EER



MAPING OF diagram

STEP 1: Mapping of Regular Entity Types.

1. LOGIN

| | |
|-----------------|-----------------|
| <u>USERNAME</u> | <u>PASSWORD</u> |
|-----------------|-----------------|

2. CUSTOMER

| | | | |
|-----------------|------------------|-----------------|----------------------|
| <u>USERNAME</u> | <u>FIRSTNAME</u> | <u>LASTNAME</u> | <u>DATE-OF-BIRTH</u> |
|-----------------|------------------|-----------------|----------------------|

3. ADMIN

| | | | |
|-----------------|------------------|-----------------|----------------------|
| <u>USERNAME</u> | <u>FIRSTNAME</u> | <u>LASTNAME</u> | <u>DATE-OF-BIRTH</u> |
|-----------------|------------------|-----------------|----------------------|

4. REGISTRATION

| | | | | |
|-----------------|------------------|-----------------|----------------------|-----------------|
| <u>USERNAME</u> | <u>FIRSTNAME</u> | <u>LASTNAME</u> | <u>DATE-OF-BIRTH</u> | <u>PASSWORD</u> |
|-----------------|------------------|-----------------|----------------------|-----------------|

5. CARTYPE

| |
|----------------|
| <u>CARCODE</u> |
|----------------|

6. BILLING

| | | | | | | |
|----------------|-----------------|-----------------|----------------|---------------|--------------|-------------|
| <u>CARCODE</u> | <u>USERNAME</u> | <u>INVOICE#</u> | <u>CARNAME</u> | <u>COLOUR</u> | <u>PRICE</u> | <u>DATE</u> |
|----------------|-----------------|-----------------|----------------|---------------|--------------|-------------|

7. RECEPT

| | | | | | | |
|----------------|-----------------|-----------------|----------------|---------------|--------------|-------------|
| <u>CARCODE</u> | <u>USERNAME</u> | <u>INVOICE#</u> | <u>CARNAME</u> | <u>COLOUR</u> | <u>PRICE</u> | <u>DATE</u> |
|----------------|-----------------|-----------------|----------------|---------------|--------------|-------------|

8. Hatchback

| | | | | |
|-----------------|-----------------|----------------------------|---------------------|------------------|
| <u>CARNAME</u> | <u>MODEL</u> | <u>ENGINE-DISPLACEMENT</u> | <u>ENGINE-POWER</u> | <u>FUEL-TYPE</u> |
| <u>GEAR-BOX</u> | <u>TOPSPEED</u> | <u>SEATING-CAPACITY</u> | <u>TRANSMISSION</u> | <u>PRICE_RS</u> |

9. Sedan

| | | | | |
|-----------------|-----------------|----------------------------|---------------------|------------------|
| <u>CARNAME</u> | <u>MODEL</u> | <u>ENGINE-DISPLACEMENT</u> | <u>ENGINE-POWER</u> | <u>FUEL-TYPE</u> |
| <u>GEAR-BOX</u> | <u>TOPSPEED</u> | <u>SEATING-CAPACITY</u> | <u>TRANSMISSION</u> | <u>PRICE_RS</u> |

10. SUV

| | | | | |
|-----------------|-----------------|----------------------------|---------------------|------------------|
| <u>CARNAME</u> | <u>MODEL</u> | <u>ENGINE-DISPLACEMENT</u> | <u>ENGINE-POWER</u> | <u>FUEL-TYPE</u> |
| <u>GEAR-BOX</u> | <u>TOPSPEED</u> | <u>SEATING-CAPACITY</u> | <u>TRANSMISSION</u> | <u>PRICE_RS</u> |

STEP 2: Mapping of Weak Entity Types
(NOT POSSIBLE)

STEP 3: Mapping of Binary 1:1 Relation Types
(NOT POSSIBLE)

STEP 4: Mapping of Binary 1:N Relationship Types

1. Get

| | | | | |
|-----------------|------------------|------------------|-----------------|----------------------|
| <u>USERNAME</u> | <u>Rinvoice#</u> | <u>FIRSTNAME</u> | <u>LASTNAME</u> | <u>DATE-OF-BIRTH</u> |
|-----------------|------------------|------------------|-----------------|----------------------|

2. Register

| | | | | | |
|------------------|------------------|------------------|-----------------|----------------------|-----------------|
| <u>Rusername</u> | <u>Cusername</u> | <u>FIRSTNAME</u> | <u>LASTNAME</u> | <u>DATE-OF-BIRTH</u> | <u>PASSWORD</u> |
|------------------|------------------|------------------|-----------------|----------------------|-----------------|

STEP 5: Mapping of Binary M:N Relationship Types.
(NOT POSSIBLE)

STEP 6: Mapping of Multivalued attributes
(NOT POSSIBLE)

STEP 7: Mapping of N-ary Relationship Types

- HAS RELATION

1. Has

| | | | |
|------------------|------------------|------------------|------------------|
| <u>Lusername</u> | <u>Ausername</u> | <u>Rusername</u> | <u>Cusername</u> |
|------------------|------------------|------------------|------------------|

2. Login

| | |
|-----------------|-----------------|
| <u>USERNAME</u> | <u>PASSWORD</u> |
|-----------------|-----------------|

3. Admin

| | | | |
|-----------------|------------------|-----------------|----------------------|
| <u>USERNAME</u> | <u>FIRSTNAME</u> | <u>LASTNAME</u> | <u>DATE-OF-BIRTH</u> |
|-----------------|------------------|-----------------|----------------------|

4. Registration

| | | | | |
|-----------------|------------------|-----------------|----------------------|-----------------|
| <u>USERNAME</u> | <u>FIRSTNAME</u> | <u>LASTNAME</u> | <u>DATE-OF-BIRTH</u> | <u>PASSWORD</u> |
|-----------------|------------------|-----------------|----------------------|-----------------|

5. Customer

| | | | |
|-----------------|------------------|-----------------|----------------------|
| <u>USERNAME</u> | <u>FIRSTNAME</u> | <u>LASTNAME</u> | <u>DATE-OF-BIRTH</u> |
|-----------------|------------------|-----------------|----------------------|

- MANAGES RELATION

1. Manages

| | | |
|------------------|------------------|----------------|
| <u>Ausername</u> | <u>Binvoice#</u> | <u>CARCODE</u> |
|------------------|------------------|----------------|

2. ADMIN

| | | | |
|-----------------|------------------|-----------------|----------------------|
| <u>USERNAME</u> | <u>FIRSTNAME</u> | <u>LASTNAME</u> | <u>DATE-OF-BIRTH</u> |
|-----------------|------------------|-----------------|----------------------|

3. BILLING

| | | | | | | |
|----------------|-----------------|-----------------|----------------|---------------|--------------|-------------|
| <u>CARCODE</u> | <u>USERNAME</u> | <u>INVOICE#</u> | <u>CARNAME</u> | <u>COLOUR</u> | <u>PRICE</u> | <u>DATE</u> |
|----------------|-----------------|-----------------|----------------|---------------|--------------|-------------|

4. CARTYPE

| |
|----------------|
| <u>CARCODE</u> |
|----------------|

- **VIEW/PURCHASE CARS RELATION**

1. VIEW/PURCHASE CARS

| | | | |
|------------------|------------------|------------------|----------------|
| <u>Cusername</u> | <u>Rinvoice#</u> | <u>Binvoice#</u> | <u>CARCODE</u> |
|------------------|------------------|------------------|----------------|

2. Customer

| | | | |
|-----------------|------------------|-----------------|----------------------|
| <u>USERNAME</u> | <u>FIRSTNAME</u> | <u>LASTNAME</u> | <u>DATE-OF-BIRTH</u> |
|-----------------|------------------|-----------------|----------------------|

3. RECEPT

| | | | | | | |
|----------------|-----------------|-----------------|----------------|---------------|--------------|-------------|
| <u>CARCODE</u> | <u>USERNAME</u> | <u>INVOICE#</u> | <u>CARNAME</u> | <u>COLOUR</u> | <u>PRICE</u> | <u>DATE</u> |
|----------------|-----------------|-----------------|----------------|---------------|--------------|-------------|

4. BILLING

| | | | | | | |
|----------------|-----------------|-----------------|----------------|---------------|--------------|-------------|
| <u>CARCODE</u> | <u>USERNAME</u> | <u>INVOICE#</u> | <u>CARNAME</u> | <u>COLOUR</u> | <u>PRICE</u> | <u>DATE</u> |
|----------------|-----------------|-----------------|----------------|---------------|--------------|-------------|

5. CARTYPE

| |
|----------------|
| <u>CARCODE</u> |
|----------------|

STEP 8: Options for Mapping Specialization or Generalization. **(NOT POSSIBLE)**

STEP 9: Mapping of Union Types

1. CARTYPE

| | |
|----------------|----------------|
| <u>CARNAME</u> | <u>CARCODE</u> |
|----------------|----------------|

2. Hatchback

| | | | | |
|-----------------|-----------------|----------------------------|---------------------|------------------|
| <u>CARNAME</u> | <u>MODEL</u> | <u>ENGINE-DISPLACEMENT</u> | <u>ENGINE-POWER</u> | <u>FUEL-TYPE</u> |
| <u>GEAR-BOX</u> | <u>TOPSPEED</u> | <u>SEATING-CAPACITY</u> | <u>TRANSMISSION</u> | <u>PRICE_RS</u> |

3. Sedan

| | | | | |
|-----------------|-----------------|----------------------------|---------------------|------------------|
| <u>CARNAME</u> | <u>MODEL</u> | <u>ENGINE-DISPLACEMENT</u> | <u>ENGINE-POWER</u> | <u>FUEL-TYPE</u> |
| <u>GEAR-BOX</u> | <u>TOPSPEED</u> | <u>SEATING-CAPACITY</u> | <u>TRANSMISSION</u> | <u>PRICE_RS</u> |

4. SUV

| | | | | |
|-----------------|-----------------|----------------------------|---------------------|------------------|
| <u>CARNAME</u> | <u>MODEL</u> | <u>ENGINE-DISPLACEMENT</u> | <u>ENGINE-POWER</u> | <u>FUEL-TYPE</u> |
| <u>GEAR-BOX</u> | <u>TOPSPEED</u> | <u>SEATING-CAPACITY</u> | <u>TRANSMISSION</u> | <u>PRICE_RS</u> |

Source code

JFRAME 1: Administrator.java

IMPORTED PACKAGES & CLASSES:

```
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 javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
```

CLASS VARIABLES:

```
Connection con;
Statement stat1=null,stat2=null,stat3=null;
ResultSet rs1,rs2,rs3;
String
CARNAME,MODEL,ENGINEDISPLACEMENT,ENGINEPOWER,FUELTYPE,GEARBOX,TRANSMISSIONTYPE,SEATI
NGCAPACITY,TOPSPEED,PRICE_RS,CARCODE;
PreparedStatement pst;
```

CODE FOR ADMINISTRATOR CONSTRUCTOR:

```
public Administrator() {
    initComponents();
    this.setLocationRelativeTo(null);
    connect();
}
```

CODE FOR CONNECT() METHOD:

```
public void connect() {
    try {
        Class.forName("com.mysql.cj.jdbc.Driver");
        con = DriverManager.getConnection("jdbc:mysql://localhost:3306/carsalessystem","root","adil095");
        stat1 = con.createStatement(resultSet.TYPE_SCROLL_SENSITIVE,resultSet.CONCUR_UPDATABLE);
        stat2 = con.createStatement(resultSet.TYPE_SCROLL_SENSITIVE,resultSet.CONCUR_UPDATABLE);
        stat3 = con.createStatement(resultSet.TYPE_SCROLL_SENSITIVE,resultSet.CONCUR_UPDATABLE);
        rs1=stat1.executeQuery("SELECT * FROM HATCHBACK");
        rs2=stat2.executeQuery("SELECT * FROM SEDAN");
        rs3=stat3.executeQuery("SELECT * FROM SUV");
    } catch (ClassNotFoundException ex) { ex.printStackTrace(); }
    catch (SQLException ex) {}
}
```

CODE FOR TABLE() METHOD:

```
public void table(){
    int value =jComboBox1.getSelectedIndex();
    try{
        Class.forName("com.mysql.cj.jdbc.Driver");
        con = DriverManager.getConnection("jdbc:mysql://localhost:3306/carsalessystem","root","adil095");
        stat1 = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_UPDATABLE);
        stat2 = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_UPDATABLE);
        stat3 = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_UPDATABLE);
        rs1=stat1.executeQuery("SELECT * FROM HATCHBACK");
        rs2=stat2.executeQuery("SELECT * FROM SEDAN");
        rs3=stat3.executeQuery("SELECT * FROM SUV");
        if(value == 1){
            while(rs1.next()){
                String code =String.valueOf(rs1.getInt("CARCODE"));
                String car =String.valueOf(rs1.getString("CARNAME"));
                String model =String.valueOf(rs1.getString("Model"));
                String engine =String.valueOf(rs1.getString("ENGINEDISPLACEMENT"));
                String fuel =String.valueOf(rs1.getString("FUELTYPE"));
                String price=String.valueOf(rs1.getInt("PRICE_RS"));
                String tbdata[]={ "HB-00"+code,car,model,engine,fuel,price};
                DefaultTableModel tblModel =(DefaultTableModel)CarTable.getModel();
                tblModel.addRow(tbdata);
            }
        }
        if(value == 2){
            while(rs2.next()){
                String code =String.valueOf(rs2.getInt("CARCODE"));
                String car =String.valueOf(rs2.getString("CARNAME"));
                String model =String.valueOf(rs2.getString("Model"));
                String engine =String.valueOf(rs2.getString("ENGINEDISPLACEMENT"));
                String fuel =String.valueOf(rs2.getString("FUELTYPE"));
                String price=String.valueOf(rs2.getInt("PRICE_RS"));
                String tbdata[]={ "SED-00"+code,car,model,engine,fuel,price};
                DefaultTableModel tblModel =(DefaultTableModel)CarTable.getModel();
                tblModel.addRow(tbdata);
            }
        }
        if(value == 3){
            while(rs3.next()){
                String code =String.valueOf(rs3.getInt("CARCODE"));
                String car =String.valueOf(rs3.getString("CARNAME"));
                String model =String.valueOf(rs3.getString("Model"));
                String engine =String.valueOf(rs3.getString("ENGINEDISPLACEMENT"));
                String fuel =String.valueOf(rs3.getString("FUELTYPE"));
                String price=String.valueOf(rs3.getInt("PRICE_RS"));
                String tbdata[]={ "SUV-00"+code,car,model,engine,fuel,price};
                DefaultTableModel tblModel =(DefaultTableModel)CarTable.getModel();
                tblModel.addRow(tbdata);
            }
        }
    }catch(Exception ex){JOptionPane.showMessageDialog(this, ex);}
}
```

CODE FOR CLEARTABLE() METHOD:

```
public void clearTable(){
    DefaultTableModel dm =(DefaultTableModel)CarTable.getModel();
    int rowCount = dm.getRowCount();
    for (int i = rowCount - 1; i >= 0; i--) {
        dm.removeRow(i);
    }
}
```

}

CODE FOR CLEARALL() METHOD:

```
public void ClearAll(){  
    TXTCarname.setText("");  
    TXTModel.setText("");  
    TXTEngineDisplay.setText("");  
    TXTEnginePower.setText("");  
    TXTfueltype.setText("");  
    TXTGearBox.setText("");  
    TXTTransmisionType.setText("");  
    TXTSeatingCapacity.setText("");  
    TXTTopSpeed.setText("");  
    TXTPrice.setText("");  
    TXTCarCode.setText("");  
}
```

CODE FOR NAVIGATION FIRST BUTTON:

```
private void BTNfirstActionPerformed(java.awt.event.ActionEvent evt) {  
    int value =jComboBox1.getSelectedIndex();  
    try{  
        if(value==1){  
            rs1.first();  
            TXTCarname.setText(rs1.getString(1));  
            TXTModel.setText(rs1.getString(2));  
            TXTEngineDisplay.setText(rs1.getString(3));  
            TXTEnginePower.setText(rs1.getString(4));  
            TXTfueltype.setText(rs1.getString(5));  
            TXTGearBox.setText(rs1.getString(6));  
            TXTTransmisionType.setText(rs1.getString(7));  
            TXTSeatingCapacity.setText(rs1.getString(8));  
            TXTTopSpeed.setText(rs1.getString(9));  
            TXTPrice.setText(rs1.getString(10));  
            TXTCarCode.setText("HB-00"+rs1.getString(11));  
        }  
        else if(value == 2){  
            rs2.first();  
            TXTCarname.setText(rs2.getString(1));  
            TXTModel.setText(rs2.getString(2));  
            TXTEngineDisplay.setText(rs2.getString(3));  
            TXTEnginePower.setText(rs2.getString(4));  
            TXTfueltype.setText(rs2.getString(5));  
            TXTGearBox.setText(rs2.getString(6));  
            TXTTransmisionType.setText(rs2.getString(7));  
            TXTSeatingCapacity.setText(rs2.getString(8));  
            TXTTopSpeed.setText(rs2.getString(9));  
            TXTPrice.setText(rs2.getString(10));  
            TXTCarCode.setText("SED-00"+rs2.getString(11));  
        }  
        else if(value == 3){  
            rs3.first();  
            TXTCarname.setText(rs3.getString(1));  
            TXTModel.setText(rs3.getString(2));  
            TXTEngineDisplay.setText(rs3.getString(3));  
            TXTEnginePower.setText(rs3.getString(4));  
            TXTfueltype.setText(rs3.getString(5));  
            TXTGearBox.setText(rs3.getString(6));  
            TXTTransmisionType.setText(rs3.getString(7));  
            TXTSeatingCapacity.setText(rs3.getString(8));  
            TXTTopSpeed.setText(rs3.getString(9));  
            TXTPrice.setText(rs3.getString(10));  
            TXTCarCode.setText("SUV-00"+rs3.getString(11));  
    }  
}
```

```
        }
        else{
            JOptionPane.showMessageDialog(this, "Please select Car type");
        }
    } catch (SQLException ex) { ex.printStackTrace();}
}
```

CODE FOR NAVIGATION NEXT BUTTON:

```
private void BTNnextActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        int value =jComboBox1.getSelectedIndex();
        if(value==1){
            if(!rs1.isLast())
            {
                rs1.next();
                TXTCarname.setText(rs1.getString(1));
                TXTModel.setText(rs1.getString(2));
                TXTEngineDisplay.setText(rs1.getString(3));
                TXTEnginePower.setText(rs1.getString(4));
                TXTfueltype.setText(rs1.getString(5));
                TXTGearBox.setText(rs1.getString(6));
                TXTTransmisionType.setText(rs1.getString(7));
                TXTSeatingCapacity.setText(rs1.getString(8));
                TXTTopSpeed.setText(rs1.getString(9));
                TXTPrice.setText(rs1.getString(10));
                TXTCarCode.setText("HB-00"+rs1.getString(11));
            }
        }
        else if(value==2){
            if(!rs2.isLast())
            {
                rs2.next();
                TXTCarname.setText(rs2.getString(1));
                TXTModel.setText(rs2.getString(2));
                TXTEngineDisplay.setText(rs2.getString(3));
                TXTEnginePower.setText(rs2.getString(4));
                TXTfueltype.setText(rs2.getString(5));
                TXTGearBox.setText(rs2.getString(6));
                TXTTransmisionType.setText(rs2.getString(7));
                TXTSeatingCapacity.setText(rs2.getString(8));
                TXTTopSpeed.setText(rs2.getString(9));
                TXTPrice.setText(rs2.getString(10));
                TXTCarCode.setText("SED-00"+rs2.getString(11));
            }
        }
        else if(value==3){
            if(!rs3.isLast())
            {
                rs3.next();
                TXTCarname.setText(rs3.getString(1));
                TXTModel.setText(rs3.getString(2));
                TXTEngineDisplay.setText(rs3.getString(3));
                TXTEnginePower.setText(rs3.getString(4));
                TXTfueltype.setText(rs3.getString(5));
                TXTGearBox.setText(rs3.getString(6));
                TXTTransmisionType.setText(rs3.getString(7));
                TXTSeatingCapacity.setText(rs3.getString(8));
                TXTTopSpeed.setText(rs3.getString(9));
                TXTPrice.setText(rs3.getString(10));
                TXTCarCode.setText("SUV-00"+rs3.getString(11));
            }
        }
    }
}
```

```
        else{
            JOptionPane.showMessageDialog(this, "Please select Car type ");
        } } catch (SQLException ex) { } }
```

CODE FOR NAVIGATION PREVIOUS BUTTON:

```
private void BTNpreviousActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try {
        int value =jComboBox1.getSelectedIndex();
        if(value==1){
            if(!rs1.isFirst()){
                rs1.previous();
                TXTCarname.setText(rs1.getString(1));
                TXTModel.setText(rs1.getString(2));
                TXTEngineDisplay.setText(rs1.getString(3));
                TXTEnginePower.setText(rs1.getString(4));
                TXTfueltype.setText(rs1.getString(5));
                TXTGearBox.setText(rs1.getString(6));
                TXTTransmissionType.setText(rs1.getString(7));
                TXTSeatingCapacity.setText(rs1.getString(8));
                TXTTopSpeed.setText(rs1.getString(9));
                TXTPrice.setText(rs1.getString(10));
                TXTCarCode.setText("HB-00"+rs1.getString(11));
            }
        else if(value==2){
            if(!rs2.isFirst()){
                rs2.previous();
                TXTCarname.setText(rs2.getString(1));
                TXTModel.setText(rs2.getString(2));
                TXTEngineDisplay.setText(rs2.getString(3));
                TXTEnginePower.setText(rs2.getString(4));
                TXTfueltype.setText(rs2.getString(5));
                TXTGearBox.setText(rs2.getString(6));
                TXTTransmissionType.setText(rs2.getString(7));
                TXTSeatingCapacity.setText(rs2.getString(8));
                TXTTopSpeed.setText(rs2.getString(9));
                TXTPrice.setText(rs2.getString(10));
                TXTCarCode.setText("SED-00"+rs2.getString(11));
            }
        }
        else if(value==3){
            if(!rs3.isFirst()){
                rs3.previous();
                TXTCarname.setText(rs3.getString(1));
                TXTModel.setText(rs3.getString(2));
                TXTEngineDisplay.setText(rs3.getString(3));
                TXTEnginePower.setText(rs3.getString(4));
                TXTfueltype.setText(rs3.getString(5));
                TXTGearBox.setText(rs3.getString(6));
                TXTTransmissionType.setText(rs3.getString(7));
                TXTSeatingCapacity.setText(rs3.getString(8));
                TXTTopSpeed.setText(rs3.getString(9));
                TXTPrice.setText(rs3.getString(10));
                TXTCarCode.setText("SUV-00"+rs3.getString(11)); } }
        else{
            JOptionPane.showMessageDialog(this, "Please select Car type ");
        }
    } } catch (SQLException ex) { } }
```

CODE FOR NAVIGATION LAST BUTTON:

```
private void BTNlastActionPerformed(java.awt.event.ActionEvent evt) {
```

```
// TODO add your handling code here:
try {
    int value =jComboBox1.getSelectedIndex();
    if(value==1){
        rs1.last();
        TXTCarname.setText(rs1.getString(1));
        TXTModel.setText(rs1.getString(2));
        TXTEngineDisplay.setText(rs1.getString(3));
        TXTEnginePower.setText(rs1.getString(4));
        TXTfueltype.setText(rs1.getString(5));
        TXTGearBox.setText(rs1.getString(6));
        TXTTransmissionType.setText(rs1.getString(7));
        TXTSeatingCapacity.setText(rs1.getString(8));
        TXTTopSpeed.setText(rs1.getString(9));
        TXTPrice.setText(rs1.getString(10));
        TXTCarCode.setText("HB-00"+rs1.getString(11));
    }
    else if(value==2){
        rs2.last();
        TXTCarname.setText(rs2.getString(1));
        TXTModel.setText(rs2.getString(2));
        TXTEngineDisplay.setText(rs2.getString(3));
        TXTEnginePower.setText(rs2.getString(4));
        TXTfueltype.setText(rs2.getString(5));
        TXTGearBox.setText(rs2.getString(6));
        TXTTransmissionType.setText(rs2.getString(7));
        TXTSeatingCapacity.setText(rs2.getString(8));
        TXTTopSpeed.setText(rs2.getString(9));
        TXTPrice.setText(rs2.getString(10));
        TXTCarCode.setText("SED-00"+rs2.getString(11));
    }
    else if(value==3){
        rs3.last();
        TXTCarname.setText(rs3.getString(1));
        TXTModel.setText(rs3.getString(2));
        TXTEngineDisplay.setText(rs3.getString(3));
        TXTEnginePower.setText(rs3.getString(4));
        TXTfueltype.setText(rs3.getString(5));
        TXTGearBox.setText(rs3.getString(6));
        TXTTransmissionType.setText(rs3.getString(7));
        TXTSeatingCapacity.setText(rs3.getString(8));
        TXTTopSpeed.setText(rs3.getString(9));
        TXTPrice.setText(rs3.getString(10));
        TXTCarCode.setText("SUV-00"+rs3.getString(11));
    }
    else{
        JOptionPane.showMessageDialog(this, "Please select Car type ");
    }
} catch (SQLException ex) { }
```

CODE FOR COMBOBOX BUTTON:

```
private void jComboBox1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String item = jComboBox1.getSelectedItem().toString();
    TXTcombo.setText(item);
    clearTable();
    table();
}
```

CODE FOR INSERT BUTTON:

```
private void BTNinsertActionPerformed(java.awt.event.ActionEvent evt) {
```

```
// TODO add your handling code here:
int value =jComboBox1.getSelectedIndex();
CARNAME = TXTCarname.getText();
MODEL = TXTModel.getText();
ENGINEDISPLACEMENT = TXTEngineDisplay.getText();
ENGINEPOWER = TXTEnginePower.getText();
FUELTYPE = TXTfueltype.getText();
GEARBOX = TXTGearBox.getText();
TRANSMISSIONTYPE = TXTTransmisionType.getText();
SEATINGCAPACITY = TXTSeatingCapacity.getText();
TOPSPEED = TXTTopSpeed.getText();
PRICE_RS = TXTPrice.getText();
CARCODE = TXTCarCode.getText();
int Price = Integer.parseInt(PRICE_RS);
int code = Integer.parseInt(CARCODE);
try{
    if(value==1){
        // SELECT
        String sql ="INSERT INTO HATCHBACK VALUES
        ("+CARNAME+",""+MODEL+",""+ENGINEDISPLACEMENT+
        ",""+ENGINEPOWER+",""+FUELTYPE+",""+GEARBOX+",""+TRANSMISSIONTYPE+",
        + """+SEATINGCAPACITY+",""+TOPSPEED+","+Price+","+code+")";
        stat1.executeUpdate(sql);
        JOptionPane.showMessageDialog(this, "Car sucessfully inserted in HATCHBACK ");
    }
    else if(value==2){
        String sql ="INSERT INTO SEDAN VALUES ("+CARNAME+",""+MODEL+",""+ENGINEDISPLACEMENT+
        ",""+ENGINEPOWER+",""+FUELTYPE+",""+GEARBOX+",""+TRANSMISSIONTYPE+",
        + """+SEATINGCAPACITY+",""+TOPSPEED+","+Price+","+code+")";
        stat1.executeUpdate(sql);
        JOptionPane.showMessageDialog(this, "Car sucessfully inserted in SEDAN ");
    }
    else if(value==3){
        String sql ="INSERT INTO SUV VALUES ("+CARNAME+",""+MODEL+",""+ENGINEDISPLACEMENT+
        ",""+ENGINEPOWER+",""+FUELTYPE+",""+GEARBOX+",""+TRANSMISSIONTYPE+",
        + """+SEATINGCAPACITY+",""+TOPSPEED+","+Price+","+code+")";
        stat1.executeUpdate(sql);
        JOptionPane.showMessageDialog(this, "Car sucessfully inserted in SUV ");
    }
    else{
        JOptionPane.showMessageDialog(this, "Please Select your Car ");
    }
}catch (Exception ex) {JOptionPane.showMessageDialog(this, ""+ex); }
clearTable();
table();
}
}
```

CODE FOR UPDATE BUTTON:

```
private void BTNupdateActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    int value =jComboBox1.getSelectedIndex();
    CARNAME = TXTCarname.getText();
    MODEL = TXTModel.getText();
    ENGINEDISPLACEMENT = TXTEngineDisplay.getText();
    ENGINEPOWER = TXTEnginePower.getText();
    FUELTYPE = TXTfueltype.getText();
    GEARBOX = TXTGearBox.getText();
    TRANSMISSIONTYPE = TXTTransmisionType.getText();
    SEATINGCAPACITY = TXTSeatingCapacity.getText();
    TOPSPEED = TXTTopSpeed.getText();
    PRICE_RS = TXTPrice.getText();
    CARCODE=TXTCarCode.getText();
}
```

```
String code3 =CARCODE.substring(CARCODE.length() - 3);
int Price = Integer.parseInt(PRICE_RS);
int code = Integer.parseInt(code3);
try{
    if(value==1){
        String sql ="UPDATE HATCHBACK SET CARNOME = "+CARNOME+",MODEL
        =" +MODEL+",ENGINEDISPLACEMENT = "+ENGINEDISPLACEMENT+
        ",ENGINEPOWER = "+ENGINEPOWER+",FUELTYPE = "+FUELTYPE+",GEARBOX =
        "+GEARBOX+",TRANSMISSIONTYPE =
        "+TRANSMISSIONTYPE+",SEATINGCAPACITY = "+SEATINGCAPACITY+",TOPSPEED =
        "+TOPSPEED+",PRICE_RS =
        +Price+" Where CARCODE = "+code ;
        stat1.executeUpdate(sql);

        JOptionPane.showMessageDialog(this, "Selected HATCHBACK Sucessfully Updated");
    }
    if(value==2){
        String sql ="UPDATE SEDAN SET CARNOME = "+CARNOME+",MODEL =" +MODEL+",ENGINEDISPLACEMENT
        = "+ENGINEDISPLACEMENT+
        ",ENGINEPOWER = "+ENGINEPOWER+",FUELTYPE = "+FUELTYPE+",GEARBOX =
        "+GEARBOX+",TRANSMISSIONTYPE =
        "+TRANSMISSIONTYPE+",SEATINGCAPACITY = "+SEATINGCAPACITY+",TOPSPEED =
        "+TOPSPEED+",PRICE_RS =
        +Price+" Where CARCODE = "+code ;
        stat1.executeUpdate(sql);
        JOptionPane.showMessageDialog(this, "Selected SEDAN Sucessfully Updated");
    }
    if(value==3){
        String sql ="UPDATE SUV SET CARNOME = "+CARNOME+",MODEL =" +MODEL+",ENGINEDISPLACEMENT =
        "+ENGINEDISPLACEMENT+
        ",ENGINEPOWER = "+ENGINEPOWER+",FUELTYPE = "+FUELTYPE+",GEARBOX =
        "+GEARBOX+",TRANSMISSIONTYPE =
        "+TRANSMISSIONTYPE+",SEATINGCAPACITY = "+SEATINGCAPACITY+",TOPSPEED =
        "+TOPSPEED+",PRICE_RS =
        +Price+" Where CARCODE = "+code ;
        stat1.executeUpdate(sql);
        JOptionPane.showMessageDialog(this, "Selected SUV Sucessfully Updated");
    }
}catch (Exception ex) {
    JOptionPane.showMessageDialog(this, "+ex);
    clearTable();
    table();
}
}
```

CODE FOR DELETE BUTTON:

```
private void BTNdeleteActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    CARCODE=TXTCarCode.getText();
    String code3 =CARCODE.substring(CARCODE.length() - 3);
    int code = Integer.parseInt(code3);
    int value =jComboBox1.getSelectedIndex();
    try{
        if(value==1){
            String sql ="DELETE FROM HATCHBACK WHERE CARCODE = "+code;
            stat1.executeUpdate(sql);
            JOptionPane.showMessageDialog(this, "Selected Car Sucessfully Deleted");
        }
        else if(value==2){
            String sql ="DELETE FROM SEDAN WHERE CARCODE = "+code;
            stat2.executeUpdate(sql);
            JOptionPane.showMessageDialog(this, "Selected Car Sucessfully Deleted");
        }
        else if(value==3){

```

```
String sql ="DELETE FROM SUV WHERE CARCODE = "+code;
stat3.executeUpdate(sql);
JOptionPane.showMessageDialog(this, "Selected Car Sucessfully Deleted");
}
}catch (Exception ex) {
    JOptionPane.showMessageDialog(this, ""+ex);
}
clearTable();
table();
ClearAll();
}
```

CODE FOR SEARCH BUTTON:

```
private void BTNsearchActionPerformed(java.awt.event.ActionEvent evt) {
    int value =jComboBox1.getSelectedIndex();
    CARCODE = TXTCarCode.getText();
    int code = Integer.parseInt(CARCODE);
    if(value==1){
        try{
            pst=con.prepareStatement("Select * from HATCHBACK where CARCODE = ?");
            pst.setInt(1, code);
            rs1=pst.executeQuery();
            if(rs1.next()){
                TXTCarname.setText(rs1.getString("CARNAME"));
                TXTModel.setText(rs1.getString("MODEL"));
                TXTEngineDisplay.setText(rs1.getString("ENGINEDISPLACEMENT"));
                TXTEnginePower.setText(rs1.getString("ENGINEPOWER"));
                TXTfueltype.setText(rs1.getString("FUELTYPE"));
                TXTGearBox.setText(rs1.getString("GEARBOX"));
                TXTTransmissionType.setText(rs1.getString("TRANSMISSIONTYPE"));
                TXTSeatingCapacity.setText(rs1.getString("SEATINGCAPACITY"));
                TXTTopSpeed.setText(rs1.getString("TOPSPEED"));
                TXTPrice.setText(rs1.getString("PRICE_RS"));
                TXTCarCode.setText("HB-00"+rs1.getString("CARCODE"));
            }
            else{
                JOptionPane.showMessageDialog(this, "Car does not exist.");
            }
        }catch (Exception ex) {
            JOptionPane.showMessageDialog(this, ""+ex);
        }
    }
    if(value==2){
        try{
            pst=con.prepareStatement("Select * from SEDAN where CARCODE= ?");
            pst.setInt(1, code);
            rs2=pst.executeQuery();
            if(rs2.next()){
                TXTCarname.setText(rs2.getString("CARNAME"));
                TXTModel.setText(rs2.getString("MODEL"));
                TXTEngineDisplay.setText(rs2.getString("ENGINEDISPLACEMENT"));
                TXTEnginePower.setText(rs2.getString("ENGINEPOWER"));
                TXTfueltype.setText(rs2.getString("FUELTYPE"));
                TXTGearBox.setText(rs2.getString("GEARBOX"));
                TXTTransmissionType.setText(rs2.getString("TRANSMISSIONTYPE"));
                TXTSeatingCapacity.setText(rs2.getString("SEATINGCAPACITY"));
                TXTTopSpeed.setText(rs2.getString("TOPSPEED"));
                TXTPrice.setText(rs2.getString("PRICE_RS"));
                TXTCarCode.setText("SED-00"+rs2.getString("CARCODE"));
            }
            else{
                JOptionPane.showMessageDialog(this, "Car does not exist");
            }
        }catch (Exception ex) {
            JOptionPane.showMessageDialog(this, ""+ex);
        }
    }
    if(value==3){

```

```
try{
    pst=con.prepareStatement("Select * from SUV where CARCODE = ?");
    pst.setInt(1, code);
    rs3=pst.executeQuery();
    if(rs3.next()){
        TXTCarname.setText(rs3.getString("CARNAME"));
        TXTModel.setText(rs3.getString("MODEL"));
        TXTEngineDisplay.setText(rs3.getString("ENGINEDISPLACEMENT"));
        TXTEnginePower.setText(rs3.getString("ENGINEPOWER"));
        TXTfueltype.setText(rs3.getString("FUELTYPE"));
        TXTGearBox.setText(rs3.getString("GEARBOX"));
        TXTTransmissionType.setText(rs3.getString("TRANSMISSIONTYPE"));
        TXTSeatingCapacity.setText(rs3.getString("SEATINGCAPACITY"));
        TXTTopSpeed.setText(rs3.getString("TOPSPEED"));
        TXTPrice.setText(rs3.getString("PRICE_RS"));
        TXTCarCode.setText("SUV-00"+rs3.getString("CARCODE"));
    }
    else{
        JOptionPane.showMessageDialog(this, "Car does not exist");
    }
}catch (Exception ex) {
    JOptionPane.showMessageDialog(this, ""+ex);
}
}
```

CODE FOR MINIMIZE WINDOW LABEL:

```
private void LBLminimizeMouseClicked(java.awt.event.MouseEvent evt) {
    this.setState(JFrame.ICONIFIED);
}
```

CODE FOR CLOSE WINDOW LABEL:

```
private void LBLcloseMouseClicked(java.awt.event.MouseEvent evt) {
    System.exit(0);
}
```

CODE FOR LOGOUT BUTTON:

```
private void BTNlogoutActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    SignInForm lf = new SignInForm();
    lf.setVisible(true);
    lf.pack();
    lf.setLocationRelativeTo(null);
    lf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    this.dispose();
}
```

CODE FOR CLEAR BUTTON:

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    ClearAll();
}
```

CODE FOR RECORD OF BILLS BUTTON:

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    AdminViewBilling bill =new AdminViewBilling();
    bill.setVisible(true);
    bill.pack();
    bill.setLocationRelativeTo(null);
    bill.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    this.dispose();
}
```

JFRAME 2: AdminViewBilling.java

IMPORTED PACKAGES & CLASSES:

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
```

CLASS VARIABLES:

```
Connection con;
Statement stat=null;
ResultSet rs;
```

CODE FOR ADMINVIEWBILLING CONSTRUCTOR:

```
public AdminViewBilling() {
    initComponents();
    this.setLocationRelativeTo(null);
    Cosbiling();
}
```

CODE FOR COSBILING () METHOD (BILLING METHOD):

```
public void Cosbiling () {
    try{
        Class.forName("com.mysql.cj.jdbc.Driver");
        con = DriverManager.getConnection("jdbc:mysql://localhost:3306/carsalessystem","root","adil095");
        stat = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_UPDATABLE);
        rs=stat.executeQuery("SELECT * FROM ADMINVIEWBILLING");
        while(rs.next()){
            String date =String.valueOf(rs.getString("DATE"));
            String user =String.valueOf(rs.getString("USERNAME"));
            String code =String.valueOf(rs.getString("CARCODE"));
            String carname =String.valueOf(rs.getString("CARNNAME"));
            String engine =String.valueOf(rs.getString("ENGINE"));
            String colour =String.valueOf(rs.getString("COLOUR"));
            String price=String.valueOf(rs.getInt("PRICE_RS"));
            String tbdata[]={date,user,code,carname,engine,colour,price};
            DefaultTableModel tblModel =(DefaultTableModel)BillTable.getModel();
            tblModel.addRow(tbdata);
        }
    }catch(Exception e){}
}
```

CODE FOR CLEAR RECORD BUTTON:

```
private void BTNclearActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try{
        DefaultTableModel dm =(DefaultTableModel)BillTable.getModel();
        int rowCount = dm.getRowCount();
        for (int i = rowCount - 1; i >= 0; i--) {
            dm.removeRow(i);
        }
        stat = con.createStatement();
        String query = "DELETE FROM ADMINVIEWBILLING";
        int deletedRows=stat.executeUpdate(query);
        if(deletedRows>0){
            System.out.println("Deleted All Rows In The Table Successfully...");}
    }
```

```
else{
    System.out.println("Table already empty.");
}
}catch(Exception e){
    JOptionPane.showMessageDialog(this,e);
}
```

CODE FOR RETURN TO ADMIN WINDOW LABEL:

```
private void LBLloginMouseClicked(java.awt.event.MouseEvent evt) {
    Administrator lf = new Administrator();
    lf.setVisible(true);
    lf.pack();
    lf.setLocationRelativeTo(null);
    lf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    this.dispose();
}
```

JFRAME 3: Complaint.java

IMPORTED PACKAGES & CLASSES:

```
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.net.Socket;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
```

CODE FOR COMPLAIN CONSTRUCTOR:

```
public Complaint() {
    initComponents();
    this.setLocationRelativeTo(null);
}
```

CODE FOR CLIENT () METHOD (BILLING METHOD):

```
public void client(String comp){
    String complaint = comp;
    try{
        System.out.println("Complaint Client Side Program");
        System.out.println("Connecting to server...");
        Socket client = new Socket("127.0.0.1",4512);
        System.out.println("Connected to server.");

        DataInputStream din = new DataInputStream(client.getInputStream());
        DataOutputStream dout = new DataOutputStream(client.getOutputStream());
        dout.writeUTF(complaint);
    }
    catch(Exception e){
        e.printStackTrace();
    }
}
```

CODE FOR SUBMIT BUTTON:

```
private void btn5ActionPerformed(java.awt.event.ActionEvent evt) {
    String compalint = jTextArea1.getText();
    client(compalint);
    JOptionPane.showMessageDialog(Complaint.this,"We regret for your inconvenience.We will respond on your email
soon.");
    jTextArea1.setText("");
    this.dispose(); }
```

CODE FOR CLEAR BUTTON:

```
private void btn6ActionPerformed(java.awt.event.ActionEvent evt) {  
    jTextArea1.setText("");}
```

JFRAME 4: CreditCardinfo.java**IMPORTED PACKAGES & CLASSES:**

```
import java.text.DateFormat;  
import java.text.SimpleDateFormat;  
import java.util.Date;  
import javax.swing.JFrame;  
import javax.swing.JOptionPane;
```

STARTING CODE FOR CLASS CREDITCARDINFO:

```
Date date = new Date();  
DateFormat dateFormat =new SimpleDateFormat("MM/dd/YY");
```

CODE FOR CREDITCARDINFO CONSTRUCTOR:

```
public CreditCardinfo() {  
    initComponents();  
    this.setLocationRelativeTo(null);  
    TXTDate.setText(""+dateFormat.format(date));  
}
```

**CODE FOR CREDITCARDINFO CONSTRUCTOR WITH STRING
PARAMETER:**

```
public CreditCardinfo(String a) {  
    initComponents();  
    this.setLocationRelativeTo(null);  
    TXTusername.setText(a);  
    TXTDate.setText(""+dateFormat.format(date));  
}
```

CODE FOR COMBOBOX BUTTON:

```
private void CMBcardActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    String item = CMBcard.getSelectedItem().toString();  
    TXTcombo.setText(item);  
}
```

CODE FOR CHECKOUT BUTTON:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    String Cardno = TXTCardNumber.getText();  
    String MMYY = TXTMMYY.getText();  
    String CVV = TXTcvv.getText();  
    if(Cardno.length()<16){  
        JOptionPane.showMessageDialog(this, "Invalid Card Number !");  
    }  
    else if(CVV.length()<3){  
        JOptionPane.showMessageDialog(this, "Invalid Card verification value !");  
    }  
}
```

```
else if(MMYY.length()<5){  
    JOptionPane.showMessageDialog(this, "Invalid moth and year !");  
}  
else{  
    JOptionPane.showMessageDialog(this, "Car successfully Purchased");  
    String a = TXTusername.getText();  
    new CustomerBilling(a).setVisible(true);  
    setVisible(false);  
    this.dispose();  
}  
}  
}
```

JFRAME 5: Customer.java

IMPORTED PACKAGES & CLASSES:

```
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.DateFormat;  
import java.text.SimpleDateFormat;  
import java.util.Calendar;  
import java.util.Date;  
import javax.swing.JFrame;  
import javax.swing.JOptionPane;  
import javax.swing.table.DefaultTableModel;
```

CLASS VARIABLES:

```
Connection con;  
Statement stat1=null,stat2=null,stat3=null,stat4=null,stat5=null;  
ResultSet rs1,rs2,rs3;  
String  
CARNAME,MODEL,ENGINEDISPLACEMENT,ENGINEPOWER,FUELTYPE,GEARBOX,TRANSMISSIONTYPE,SEATI  
NGCAPACITY,TOPSPEED,PRICE_RS,CARCODE;  
PreparedStatement pst;  
DateFormat dateFormat =new SimpleDateFormat("MM/dd/YY");  
Date date = new Date();  
Calendar cal= Calendar.getInstance();  
DefaultTableModel Model;
```

CODE FOR CUSTOMER CONSTRUCTOR:

```
public Customer() {  
    initComponents();  
    this.setLocationRelativeTo(null);  
    connect();  
    Model = (DefaultTableModel) table2.getModel();  
    TXTDate.setText(""+dateFormat.format(date));  
}
```

CODE FOR CUSTOMER CONSTRUCTOR WITH STRING

PARAMETER:

```
public Customer(String a) {  
    initComponents();  
    this.setLocationRelativeTo(null);  
    name.setText(a);  
    connect();  
    Model = (DefaultTableModel) table2.getModel();
```

```
    TXTDate.setText(""+dateFormat.format(date));  
}
```

STARTING CODE FOR CLASS CREDITCARDINFO:

```
Date date = new Date();  
DateFormat dateFormat =new SimpleDateFormat("MM/dd/YY");
```

CODE FOR CREDITCARDINFO CONSTRUCTOR:

```
public CreditCardinfo() {  
    initComponents();  
    this.setLocationRelativeTo(null);  
    TXTDate.setText(""+dateFormat.format(date)); }
```

CODE FOR CREDITCARDINFO CONSTRUCTOR WITH

PARAMETER A:

```
public CreditCardinfo(String a) {  
    initComponents();  
    this.setLocationRelativeTo(null);  
    TXTusername.setText(a);  
    TXTDate.setText(""+dateFormat.format(date));  
}
```

CODE FOR CONNECT() METHOD:

```
public void connect() {  
    try {  
        Class.forName("com.mysql.cj.jdbc.Driver");  
        con = DriverManager.getConnection("jdbc:mysql://localhost:3306/carsalessystem", "root", "adil095");  
        stat1 = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_UPDATABLE);  
        stat2 = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_UPDATABLE);  
        stat3 = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_UPDATABLE);  
        rs1=stat1.executeQuery("SELECT * FROM HATCHBACK");  
        rs2=stat2.executeQuery("SELECT * FROM SEDAN");  
        rs3=stat3.executeQuery("SELECT * FROM SUV");  
        } catch (ClassNotFoundException ex) {  
    }  
    catch (SQLException ex) {}  
}
```

CODE FOR TABLE1() METHOD:

```
public void table1(){  
int value =jComboBox1.getSelectedIndex();  
try{  
    Class.forName("com.mysql.cj.jdbc.Driver");  
    con = DriverManager.getConnection("jdbc:mysql://localhost:3306/carsalessystem", "root", "adil095");  
    stat1 = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_UPDATABLE);  
    stat2 = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_UPDATABLE);  
    stat3 = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_UPDATABLE);  
    rs1=stat1.executeQuery("SELECT * FROM HATCHBACK");  
    rs2=stat2.executeQuery("SELECT * FROM SEDAN");  
    rs3=stat3.executeQuery("SELECT * FROM SUV");  
    if(value == 1){  
        while(rs1.next()){  
            String code =String.valueOf(rs1.getInt("CARCODE"));  
            String car =String.valueOf(rs1.getString("CARNAME"));  
            String model =String.valueOf(rs1.getString("Model"));  
            String engineDisplacement = String.valueOf(rs1.getString("ENGINEDISPLACEMENT"));  
            String enginePower = String.valueOf(rs1.getString("ENGINEPOWER"));  
            String fuel =String.valueOf(rs1.getString("FUELTYPE"));  
            String Gear =String.valueOf(rs1.getString("GEARBOX"));  
            String Transmission =String.valueOf(rs1.getString("TRANSMISSIONTYPE"));  
        }  
    }  
}
```

```
String Seat =String.valueOf(rs1.getString("SEATINGCAPACITY"));
String Spead =String.valueOf(rs1.getString("TOPSPEED"));
String price=String.valueOf(rs1.getInt("PRICE_RS"));
String tbdata[]={"HB-
00"+code,car,model,engineDisplacement,enginePowert,fuel,Gear,Transmission,Seat,Spead,price};
DefaultTableModel tblModel =(DefaultTableModel)table1.getModel();
tblModel.addRow(tbdata); }
if(value == 2){
while(rs2.next()){
String code =String.valueOf(rs2.getInt("CARCODE"));
String car =String.valueOf(rs2.getString("CARNAME"));
String model =String.valueOf(rs2.getString("Model"));
String engineDisplacement = String.valueOf(rs2.getString("ENGINEDISPLACEMENT"));
String enginePowert = String.valueOf(rs2.getString("ENGINEPOWER"));
String fuel =String.valueOf(rs2.getString("FUELTYPE"));
String Gear =String.valueOf(rs2.getString("GEARBOX"));
String Transmission =String.valueOf(rs2.getString("TRANSMISSIONTYPE"));
String Seat =String.valueOf(rs2.getString("SEATINGCAPACITY"));
String Spead =String.valueOf(rs2.getString("TOPSPEED"));
String price=String.valueOf(rs2.getInt("PRICE_RS"));
String tbdata[]={"SED-
00"+code,car,model,engineDisplacement,enginePowert,fuel,Gear,Transmission,Seat,Spead,price};
DefaultTableModel tblModel =(DefaultTableModel)table1.getModel();
tblModel.addRow(tbdata); }
if(value == 3){
while(rs3.next()){
String code =String.valueOf(rs3.getInt("CARCODE"));
String car =String.valueOf(rs3.getString("CARNAME"));
String model =String.valueOf(rs3.getString("Model"));
String engineDisplacement = String.valueOf(rs3.getString("ENGINEDISPLACEMENT"));
String enginePowert = String.valueOf(rs3.getString("ENGINEPOWER"));
String fuel =String.valueOf(rs3.getString("FUELTYPE"));
String Gear =String.valueOf(rs3.getString("GEARBOX"));
String Transmission =String.valueOf(rs3.getString("TRANSMISSIONTYPE"));
String Seat =String.valueOf(rs3.getString("SEATINGCAPACITY"));
String Spead =String.valueOf(rs3.getString("TOPSPEED"));
String price=String.valueOf(rs3.getInt("PRICE_RS"));
String tbdata[]={"SUV-
00"+code,car,model,engineDisplacement,enginePowert,fuel,Gear,Transmission,Seat,Spead,price};
DefaultTableModel tblModel =(DefaultTableModel)table1.getModel();
tblModel.addRow(tbdata); }
}catch(Exception ex){JOptionPane.showMessageDialog(this, ex);}
}
```

CODE FOR CLEARTABLE() METHOD:

```
public void clear(){
DefaultTableModel dm =(DefaultTableModel)table1.getModel();
int rowCount = dm.getRowCount();
for (int i = rowCount - 1; i >= 0; i--) {
dm.removeRow(i); }}
```

CODE FOR JTABLETODB() METHOD:

```
public void JtableToDB(){
DefaultTableModel tblModel =(DefaultTableModel)table2.getModel();
if(tblModel.getRowCount()==0){ }
else{
try{
Class.forName("com.mysql.cj.jdbc.Driver");
con = DriverManager.getConnection("jdbc:mysql://localhost:3306/carsalessystem","root","adil095");
stat4 = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_UPDATABLE);
stat5 = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_UPDATABLE);
rs1=stat4.executeQuery("SELECT * FROM CUSTOMERBILLING");
```

```
rs2=stat5.executeQuery("SELECT * FROM ADMINVIEWBILLING");
for(int i=0; i<tblModel.getRowCount(); i++){
    String Date =tblModel.getValueAt(i,0).toString();
    String Username=name.getText();
    String CarCode =tblModel.getValueAt(i,1).toString();
    String CarName =tblModel.getValueAt(i,2).toString();
    String Engine =tblModel.getValueAt(i,3).toString();
    String Colour =tblModel.getValueAt(i,5).toString();
    String Price =tblModel.getValueAt(i,7).toString();
    int price = Integer.parseInt(Price);
    String sql1 ="INSERT INTO CUSTOMERBILLING VALUES
    (""+Date+"",""+Username+"",""+CarCode+"",""+CarName+"",""+Engine+"",""+Colour+"",""+price+"");
    stat4.executeUpdate(sql1);
    String sql2 ="INSERT INTO ADMINVIEWBILLING VALUES
    (""+Date+"",""+Username+"",""+CarCode+"",""+CarName+"",""+Engine+"",""+Colour+"",""+price+"");
    stat5.executeUpdate(sql2); }
}catch(Exception e){
    e.printStackTrace();} }}
```

CODE FOR COMBOBOX BUTTON:

```
private void jComboBox1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String item = jComboBox1.getSelectedItem().toString();
    TXTcombo.setText(item);
    clear();
    table1();
    TXTCarCode.setText(""); }}
```

CODE FOR ADD TO CART BUTTON:

```
private void BTNaddActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    CARCODE=TXTCarCode.getText();
    int code = Integer.parseInt(CARCODE);
    int value =jComboBox1.getSelectedIndex();
    String t=null;
    String s=null;
    if(RDBTNred.isSelected()){
        t=RDBTNred.getText(); }
    else if(RDBTNblue.isSelected()){
        t=RDBTNblue.getText(); }
    else if(RDBTNblack.isSelected()){
        t=RDBTNblack.getText(); }
    else if(RDBTNsilver.isSelected()){
        t=RDBTNsilver.getText(); }
    else if(RDBTNwhite.isSelected()){
        t=RDBTNwhite.getText(); }
    else if(RDBTNyellow.isSelected()){
        t=RDBTNyellow.getText(); }
    else if(RDBTNgray.isSelected()){
        t=RDBTNgray.getText(); }
    if(CBhidLight.isSelected()&&CBallyRim.isSelected()){
        s=CBhidLight.getText()+" & "+CBallyRim.getText(); }
    else if(CBhidLight.isSelected()){
        s=CBhidLight.getText(); }
    else if(CBallyRim.isSelected()){
        s=CBallyRim.getText(); }
    else{
        s="No Accessories"; }
    try{
        if(value==1){
            pst=con.prepareStatement("Select * from HATCHBACK where CARCODE= ?");
```

```
    pst.setInt(1, code);
    rs1=pst.executeQuery();
    if(rs1.next()){
        if(t==null){
            JOptionPane.showMessageDialog(this, "Please Select Colour of your Car");}
        else{
            Model.insertRow(Model.getRowCount(),new Object[]{TXTDate.getText(),"HB-
00"+rs1.getString("CARCODE"),rs1.getString("CARNAME"),rs1.getString("ENGINEDIPLACEMENT"),rs1.getString("FUE
LTYPE"),t,s,rs1.getInt("PRICE_RS"))};    }
    else{
        JOptionPane.showMessageDialog(this, "This Car does not exist");    }
    else if(value==2){
        pst=con.prepareStatement("Select * from SEDAN where CARCODE= ?");
        pst.setInt(1, code);
        rs2=pst.executeQuery();
        if(rs2.next()){
            if(t==null){
                JOptionPane.showMessageDialog(this, "Please Select Colour of your Car");    }
            else{
                Model.insertRow(Model.getRowCount(),new Object[]{TXTDate.getText(),"SED-
00"+rs2.getString("CARCODE"),rs2.getString("CARNAME"),rs2.getString("ENGINEDIPLACEMENT"),rs2.getString("FUE
LTYPE"),t,s,rs2.getInt("PRICE_RS"))};    }
            else{
                JOptionPane.showMessageDialog(this, "This Car does not exist");    }
        else if(value==3){
            pst=con.prepareStatement("Select * from SUV where CARCODE= ?");
            pst.setInt(1, code);
            rs3=pst.executeQuery();
            if(rs3.next()){
                if(t==null){
                    JOptionPane.showMessageDialog(this, "Please Select Colour of your Car");    }
                else{
                    Model.insertRow(Model.getRowCount(),new Object[]{TXTDate.getText(),"SUV-
00"+rs3.getString("CARCODE"),rs3.getString("CARNAME"),rs3.getString("ENGINEDIPLACEMENT"),rs3.getString("FUE
LTYPE"),t,s,rs3.getInt("PRICE_RS"))};    }
            else{
                JOptionPane.showMessageDialog(this, "This Car does not exist");    }
        }catch (Exception ex){JOptionPane.showMessageDialog(this, ""+ex);}
        //Model.insertRow(Model.getRowCount(),new
        Object[]{TXTDate.getText(),CarCode.getText(),TXTNum.getText(),TXTAge.getText()});    }
    }
```

CODE FOR RADIO BUTTON BUTTONS:

```
private void RDBTNredActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if(RDBTNred.isSelected()) {
        RDBTNblue.setSelected(false);
        RDBTNblack.setSelected(false);
        RDBTNsilver.setSelected(false);
        RDBTNwhite.setSelected(false);
        RDBTNyellow.setSelected(false);
        RDBTNgray.setSelected(false);
    }    }
```

Same Coding for remaining Radio Buttons (all colors)

CODE FOR DELETE BUTTON:

```
private void BTNdeleteActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try{
        DefaultTableModel xm =(DefaultTableModel)table2.getModel();
        int rm =table2.getSelectedRow();
        xm.removeRow(rm);
```

```
    }catch(Exception ex){  
        JOptionPane.showMessageDialog(this, "Please Select Which car you want to remove");  
    } }  
This button will delete the car from the cart
```

CODE FOR PURCHASE BUTTON:

```
private void BTNpurchaseActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    JtableToDB();  
    JOptionPane.showMessageDialog(this, "Enter Credit Card information");  
    String a =name.getText();  
    new CreditCardinfo(a).setVisible(true);  
    setVisible(false);  
    this.dispose();  
}  
This button will purchase all the cars available in the cart
```

CODE FOR CLEAR BUTTON:

```
private void BTNclearActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    DefaultTableModel dm =(DefaultTableModel)table2.getModel();  
    int rowCount = dm.getRowCount();  
    for (int i = rowCount - 1; i >= 0; i--) {  
        dm.removeRow(i);  
    }  
This button will clear all the cars from the jTable
```

CODE FOR LOGOUT BUTTON:

```
private void BTNlogoutActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    SignInForm lf =new SignInForm();  
    lf.setVisible(true);  
    lf.pack();  
    lf.setLocationRelativeTo(null);  
    lf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    this.dispose();  
}  
This button will logout the Customer and redirect to login window
```

JFRAME 6: CustomerBilling.java

IMPORTED PACKAGES & CLASSES:

```
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.ResultSet;  
import java.sql.Statement;  
import java.text.DateFormat;  
import java.text.SimpleDateFormat;  
import java.util.Calendar;  
import java.util.Date;  
import javax.swing.JFrame;  
import javax.swing.JOptionPane;
```

CLASS VARIABLES:

```
DateFormat dateFormat =new SimpleDateFormat("MM/dd/YY");
Date date = new Date();
Calendar cal= Calendar.getInstance();
Connection con;
Statement stat1=null;
ResultSet rs1;
```

CODE FOR CUSTOMER BILLING CONSTRUCTOR:

```
public CustomerBilling() {
    initComponents();
    this.setLocationRelativeTo(null);
    TXTDate.setText(""+dateFormat.format(date));
    bill();
}
```

CODE FOR CUSTOMER BILLING CONSTRUCTOR WITH PARAMETER:

```
public CustomerBilling(String a) {
    initComponents();
    this.setLocationRelativeTo(null);
    TXTusername.setText(a);
    TXTDate.setText(""+dateFormat.format(date));
    bill();
}
```

CODE FOR BILL() METHOD:

```
public void bill(){
    try{
        String date = TXTDate.getText();
        String username = TXTusername.getText();
        Class.forName("org.apache.derby.jdbc.ClientDriver");
        AreaTXTbill.setText("=====
                        TOYOTA CAR SALES SYSTEM" +
                        "=====
                        +"\nLUCKY-ONE MALL \nTOYOTAL MOTORS\nKARACHI"
                        +"\nNTN No : 0698064-3"
                        +"\n=====
                        +"\n User Name :" +username +"
                        +"\n Date :" +date +"
                        +"\n Invoice No : LOM-LM-T3-24805"
                        +"\n=====
                        =====");
        con = DriverManager.getConnection("jdbc:mysql://localhost:3306/carsalessystem","root","adil095");
        stat1 = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_UPDATABLE);
        rs1=stat1.executeQuery("SELECT * FROM CUSTOMERBILLING");
        int i=0,Tprice=0;
        while(rs1.next()){
            i++;
            AreaTXTbill.setText(AreaTXTbill.getText()+"\n Car Name "+i+": "+rs1.getString("CARNAME")+"\n Car Code :
            "+rs1.getString("CARCODE")+"\n Engine : "+rs1.getString("ENGINE")+"\n Colour : "+rs1.getString("COLOUR")+"\n Price
            : "+rs1.getInt("PRICE_RS")+
            "\n=====");
            Tprice=Tprice+rs1.getInt("PRICE_RS");
        }
        long Plus_Tax=(long)(Tprice*0.05)+Tprice;
        AreaTXTbill.setText(AreaTXTbill.getText()+"\nTotal Price : "+Tprice+
            "\nTex per Car: 5%"+
            "\nTotal Amount plus Tex : "+Plus_Tax
            +"\n=====");
    }
}
```

```
+"\n\tTERMS & CONDITIONS"
+"n1. No Cash Refund." +
"\n2. Total price inclusive of all applicable taxes." +
"\n3. Cars on \"Sales\" can not be exchanged." +
"\n4. Kindly check the car at the time of purchase." +
"\n5. Sale Tax can be verified by using FBR invoice number at www.e.fbr.gov.pk after 24 Hrs"
+ " \n from time of invoice generated."
+"n=====
+"n                      THANKS FOR PURCHASING"
+"n=====
+"n                      Info: 0312-2552726"
+"n                      www.ToyotaCarSales.com"
+"n=====
+"nFBR Invoice Number : 11166821122017582706\n"
+"nVerify this invoice through FBR TaxAsaan MobileApp\nSMS at 9966 and win exciting prizes in Draw"
+"n=====
stat1 = con.createStatement();
String query = "DELETE FROM CUSTOMERBILLING";
int deletedRows=stat1.executeUpdate(query);
if(deletedRows>0){
    System.out.println("Deleted All Rows In The Table Successfully...");}
else{
    System.out.println("Table already empty.");}
}catch(Exception s){ }}
```

CODE FOR PRINT BUTTON:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try{
        AreaTXTbill.print();
        AreaTXTbill.setText("");
        String a =TXTusername.getText();
        new Customer(a).setVisible(true);
        setVisible(false);
        this.dispose();
    }catch(Exception e){ JOptionPane.showMessageDialog(this,e);}
}
```

This button will print the bill of the user in real time and it'll also give option to save bill as pdf

JFRAME 7: FAQ.java

IMPORTED PACKAGES & CLASSES:

```
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.net.Socket;
import javax.swing.JFrame;
```

CODE FOR FAQ CONSTRUCTOR:

```
public FAQ() {
    initComponents();
    this.setLocationRelativeTo(null);
}
```

CODE FOR CLIENT() METHOD:

```
public String client(String ques){
    String question = ques;
    String ans="";
    try{
        System.out.println("Client Side Program");
```

```
System.out.println("Connecting to server...");  
Socket client = new Socket("127.0.0.1",4512);  
System.out.println("Connected to server.");  
DataInputStream din = new DataInputStream(client.getInputStream());  
DataOutputStream dout = new DataOutputStream(client.getOutputStream());  
dout.writeUTF(question);  
ans = din.readUTF();  
catch(Exception e){  
    e.printStackTrace();  
}  
return ans; }
```

CODE FOR PRINT BUTTON:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    try{  
        AreaTXTbill.print();  
        AreaTXTbill.setText("");  
        String a =TXTusername.getText();  
        new Customer(a).setVisible(true);  
        setVisible(false);  
        this.dispose();  
    }catch(Exception e){ JOptionPane.showMessageDialog(this,e);}  
}
```

This method will take string value from user and pass it to the server and the server will send back answer to this method and then this method will give the answer to customer

CODE FOR QUESTION BUTTONS:

```
private void btn4ActionPerformed(java.awt.event.ActionEvent evt) {  
    jLabel1.setText("");jLabel2.setText("");jLabel3.setText("");  
    jLabel4.setText("");jLabel5.setText("");  
    String question="How to contact our company?";  
    String answer = client(question);  
    jLabel6.setText(answer);  
}  
private void btn5ActionPerformed(java.awt.event.ActionEvent evt) {  
    jLabel1.setText("");jLabel2.setText("");jLabel3.setText("");  
    jLabel5.setText("");jLabel6.setText("");  
    String question = "How to buy car through this software?";  
    String answer = client(question);  
    jLabel4.setText(answer);  
}  
private void btn6ActionPerformed(java.awt.event.ActionEvent evt) {  
    jLabel1.setText("");jLabel2.setText("");jLabel3.setText("");  
    jLabel4.setText("");jLabel6.setText("");  
    String question = "Is Insurance available for Toyota cars?";  
    String answer = client(question);  
    jLabel5.setText(answer); }  
private void btn1ActionPerformed(java.awt.event.ActionEvent evt) {  
    jLabel2.setText("");jLabel3.setText("");jLabel4.setText("");  
    jLabel5.setText("");jLabel6.setText("");  
    String question="Most sold car of 2021?";  
    String answer = client(question);  
    jLabel1.setText(answer);  
}  
private void btn2ActionPerformed(java.awt.event.ActionEvent evt) {  
    jLabel1.setText("");jLabel3.setText("");jLabel4.setText("");  
    jLabel5.setText("");jLabel6.setText("");  
    String question="Where is our company located?";  
    String answer = client(question);  
    jLabel2.setText(answer); }  
private void btn3ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
jLabel1.setText("");jLabel2.setText("");jLabel4.setText("");
jLabel5.setText("");jLabel6.setText("");
String question="Work hours of our company?";
String answer = client(question);
jLabel3.setText(answer); }
```

CODE FOR OPENING NEWSLETTER WINDOW BUTTON:

```
private void btn7ActionPerformed(java.awt.event.ActionEvent evt) {
    Newsletter nl = new Newsletter();
    nl.setVisible(true);
    nl.pack();
    nl.setLocationRelativeTo(null);
    nl.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
}
```

This button will redirect customer to Newsletter window where he/she can enter email to get email subscription of latest updates from admin/company

CODE FOR OPENING COMPLAINT WINDOW BUTTON:

```
private void btn8ActionPerformed(java.awt.event.ActionEvent evt) {
    Complaint cm = new Complaint();
    cm.setVisible(true);
    cm.pack();
    cm.setLocationRelativeTo(null);
    cm.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);}
```

This button will redirect the customer to Complaint window to submit complain

CODE FOR RETURN TO SIGNIN WINDOW LABEL:

```
private void LBLlogin1MouseClicked(java.awt.event.MouseEvent evt) {
    SignInForm lf =new SignInForm();
    lf.setVisible(true);
    lf.pack();
    lf.setLocationRelativeTo(null);
    lf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    this.dispose();}
```

This label will redirect customer to Sign-In window

JFRAME 8: Newsletter.java

IMPORTED PACKAGES & CLASSES:

```
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.net.Socket;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
```

CODE FOR NEWSLETTER CONSTRUCTOR:

```
public Newsletter() {
    initComponents();
    this.setLocationRelativeTo(null);
}
```

CODE FOR CLIENT() METHOD:

```
public void client(String em){
    String email = em;
    try{
        System.out.println("Client Side Program");
        System.out.println("Connecting to server... ");
        Socket client = new Socket("127.0.0.1",4512);
        System.out.println("Connected to server.");
    }
```

```
DataStream din = new DataInputStream(client.getInputStream());
DataStream dout = new DataOutputStream(client.getOutputStream());
dout.writeUTF(email);
catch(Exception e){
    e.printStackTrace();
}}
```

CODE FOR SUBSCRIBE BUTTON:

```
private void btn1ActionPerformed(java.awt.event.ActionEvent evt) {
    String email = jTextField1.getText();
    client(email);
    JOptionPane.showMessageDialog(Newsletter.this,"You have successfully subscribed to our newsletter.");
    jTextField1.setText("");
    this.dispose();
}
```

JFRAME 9: RegisterForm.java

IMPORTED PACKAGES & CLASSES:

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
```

CLASS VARIABLES:

```
Connection con;
Statement stat=null,stat1=null;
ResultSet rs;
String FIRSTNAME, LASTNAME, PASSWORD, BIRTHDATE, USERNAME, Cpassword;
```

CODE FOR REGISTERFORM CONSTRUCTOR:

```
public RegisterForm() {
    initComponents();
    this.setLocationRelativeTo(null);
    connect();
}
```

CODE FOR CONNECT() METHOD:

```
public void connect() {
    try {
        Class.forName("com.mysql.cj.jdbc.Driver");
        con = DriverManager.getConnection("jdbc:mysql://localhost:3306/carsalesystem", "root", "adil095");
        stat = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_READ_ONLY);
    } catch (ClassNotFoundException ex) {
        ex.printStackTrace();
    } catch (SQLException ex) {
    }
}
```

CODE FOR REGISTER AS CUSTOMER BUTTON:

```
private void BTNregCostumerActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    FIRSTNAME = TXTFirstName.getText();
    LASTNAME = TXTFirstLast.getText();
    USERNAME = TXTusername.getText();
    PASSWORD = jPasswordPASSWORD.getText();
```

```
Cpassword = jPasswordCONFIRM.getText();
BIRTHDATE = TXTDateOfBirth.getText();

if(FIRSTNAME.equals("")||LASTNAME.equals("")||USERNAME.equals("")||PASSWORD.equals("")||Cpassword.equals("")||BIRTHDATE.equals("")){
    JOptionPane.showMessageDialog(this, "Please enter all the fields.");
} else{
    if(PASSWORD.equals(Cpassword)==false){
        JOptionPane.showMessageDialog(this, "Password does not match");
    } else if(!PASSWORD.equals(Cpassword)==false){
        try{
            String sql ="INSERT INTO CUSTOMER VALUES
(""+FIRSTNAME+"','"+"+LASTNAME+"','"+"+USERNAME+"','"+"+PASSWORD+"','"+"+BIRTHDATE+"')";
            stat.executeUpdate(sql);
            JOptionPane.showMessageDialog(this, "You are Successfully registered. Now Sign In to your Account");
        } catch (Exception ex) {}
    }
}
```

CODE FOR CLEAR BUTTON:

```
private void BTNcancelActionPerformed(java.awt.event.ActionEvent evt) {
    TXTFirstName.setText("");
    TXTLastName.setText("");
    TXTusername.setText("");
    jPasswordPASSWORD.setText("");
    jPasswordCONFIRM.setText("");
    TXTDateOfBirth.setText("");}
}
```

JFRAME 10: SignInForm.java

IMPORTED PACKAGES & CLASSES:

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
```

CLASS VARIABLES:

```
String AdminUsername ="Leader123";
String Adminpassword ="ssuet";
Connection con;
Statement stat=null,stat1=null;
ResultSet rs,rs1;
```

CODE FOR SIGNINFORM CONSTRUCTOR:

```
public SignInForm() {
    initComponents();
    this.setLocationRelativeTo(null);
    connect();
    MyServer ms= new MyServer();
}
```

CODE FOR CONNECT() METHOD:

```
public void connect() {  
    try {  
        Class.forName("com.mysql.cj.jdbc.Driver");  
        con = DriverManager.getConnection("jdbc:mysql://localhost:3306/carsalessystem","root","adil095");  
        stat = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_READ_ONLY);  
        stat1 = con.createStatement();  
        rs=stat.executeQuery("SELECT USERNAME, PASSWORD FROM CUSTOMER");  
    } catch (ClassNotFoundException ex) {}  
    catch (SQLException ex) {}  
}
```

CODE FOR CLEAR BUTTON:

```
private void BTNclearActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    TXTusername.setText("");  
    jPasswordField1.setText("");  
}
```

CODE FOR OPENING REGISTER FORM WINDOW LABEL:

```
private void LBLRegisterMouseClicked(java.awt.event.MouseEvent evt) {  
    RegisterForm rgf =new RegisterForm();  
    rgf.setVisible(true);  
    rgf.pack();  
    rgf.setLocationRelativeTo(null);  
    rgf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    this.dispose();  
}
```

This label will redirect the customer to the Register/Sign up Window

CODE FOR SIGN IN AS CUSTOMER BUTTON:

```
private void BTNsingCustomerActionPerformed(java.awt.event.ActionEvent evt) {  
    connect();  
    String x = TXTusername.getText();  
    String y = jPasswordField1.getText();  
    try{  
        int count1 =0;  
        int count2 =0;  
        int count3 =0;  
        while(rs.next()){  
            if(rs.getString("USERNAME").equals(x)==true && rs.getString("PASSWORD").equals(y)==true ){  
                count1++;  
            }  
            else if(rs.getString("USERNAME").equals(x)==true && rs.getString("PASSWORD").equals(y)==false ){  
                count2++;  
            }  
            else{  
                count3++;  
            }  
        }  
        if(count1>0){  
            String a =TXTusername.getText();  
            new Customer(a).setVisible(true);  
            setVisible(false);  
            this.dispose();  
        }  
        else if(count2>0){  
            JOptionPane.showMessageDialog(this, "You entered Wrong password"); }  
        else{  
            JOptionPane.showMessageDialog(this, "Your User name and Paswword are incorrect");  
        }  
    }
```

```
        }
    catch(Exception ex){ JOptionPane.showMessageDialog(this,ex);}
    }
```

CODE FOR SIGN IN AS CUSTOMER BUTTON:

```
private void BTNsiginActionPerformed(java.awt.event.ActionEvent evt) {
    String x = TXTusername.getText();
    String y = jPasswordField1.getText();
    if(AdminUsername.equals(x)==true && Adminpassword.equals(y)==true){
        Administrator admin =new Administrator();
        admin.setVisible(true);
        admin.pack();
        admin.setLocationRelativeTo(null);
        admin.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.dispose();
    }
    else if (AdminUsername.equals(x)==true && Adminpassword.equals(y)==false){
        JOptionPane.showMessageDialog(this, "You entered Wrong password");
    }   else{
        JOptionPane.showMessageDialog(this, "Your User name and Pasword are incorrect");
    }
}
```

CODE FOR HELP BUTTON:

```
private void BTNhelpActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    FAQ f=new FAQ();
    f.setVisible(true);
    f.pack();
    f.setLocationRelativeTo(null);
    f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    this.dispose();
}
```

This button will redirect the customer to the FAQ Window where customer can seek help from Admin

Class: MyServer.java

```
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.net.ServerSocket;
public class MyServer {
    public static void main(String[] args) {
        boolean run = true;
        while(run){ //runs forever (zahir's logic)
            try{
                String question="", answer="";
                System.out.println("Server Side Program");
                System.out.println("Establishing the connection with client...");
                ServerSocket server = new ServerSocket(4512);
                java.net.Socket client = server.accept();
                System.out.println("Connection Established.");
                DataInputStream din = new DataInputStream(client.getInputStream());
                DataOutputStream dout = new DataOutputStream(client.getOutputStream());
                question = din.readUTF();
                System.out.println(question);
                switch(question){
                    case "Most sold car of 2021?":
                        answer = "Toyota Yaris";
                        System.out.println(answer);
                        dout.writeUTF(answer);
                        dout.flush();
                        break;
                }
            }
        }
    }
}
```

```
case "Where is our company located?":  
    answer = "Defence Block 6, Karachi";  
    System.out.println(answer);  
    dout.writeUTF(answer);  
    dout.flush();  
    break;  
case "Work hours of our company?":  
    answer = "Monday-Friday: 9 AM - 6 PM";  
    System.out.println(answer);  
    dout.writeUTF(answer);  
    dout.flush();  
    break;  
case "How to contact our company?":  
    answer = "Our Contact Numbers are: 0336-4855036,0331-3114993 ";  
    System.out.println(answer);  
    dout.writeUTF(answer);  
    dout.flush();  
    break;  
case "How to buy car through this software?":  
    answer = "Register your account and click on login button,then buy car. ";  
    System.out.println(answer);  
    dout.writeUTF(answer);  
    dout.flush();  
    break;  
case "Is Insurance available for Toyota cars?":  
    answer = "Yes,for some cars we offer complete insurance. ";  
    System.out.println(answer);  
    dout.writeUTF(answer);  
    dout.flush();  
    break; }  
server.close();  
client.close();}  
catch(Exception e){  
    e.printStackTrace();}}}  
}
```

output

Customer Sign In:



Sign In window that will login user

Sign Up Window:

Toyota Car Sales System

SignUp

First Name : Zahir

Last Name : Khan

Username : zahir9048

Password : ****

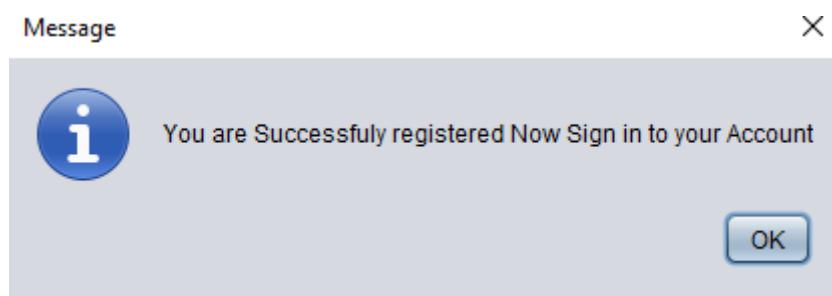
Confirm : ****

Date of Birth: 26-Dec-2000

Register As Customer Clear

Already registered? Click here to Login

After clicking on click here to create new account, Sign Up window will show that will register user account



After clicking on Register as Customer, this message shows

Welcome To Toyota Car Sales System

- X



SignIn

Need Help?

Username : zahir9048

Password : ****

Clear

Sign In As Administrator

Sign In As Customer

If you did not signed up before then, Click here to create a new account

Singing In as Customer

Please Select Car type :

HATCHBACK

HATCHBACK

| Car Code | Car Name | Model | Engine Displacem... | Engine Power | Fuel-Type | Gear Box | Transmission-Type | Seating Capacity | Top-Speed | Price_RS |
|----------|-------------|-------|---------------------|------------------|-----------|----------|-------------------|------------------|-----------|----------|
| HB-001 | Aqua | 2021 | 1496 cc | 72 hp @ 4800 RPM | Hybrid | CVT | Automatic (CVT) | 5 persons | 180 KM/H | 3500000 |
| HB-002 | Vitz Jewela | 2017 | 996 cc | 71 hp @ 6000 RPM | Petrol | 5 Speed | Automatic | 5 persons | 180 KM/H | 2500000 |
| HB-003 | Vitz | 2021 | 996 cc | 68 hp @ 6000 RPM | Petrol | CVT | Automatic | 5 persons | 180 KM/H | 2700000 |

The table shows all the available cars with specifications of the selected vehicle category

Please Select Car type :

SEDAN

SEDAN

| Car Code | Car Name | Model | Engine Displacem... | Engine Power | Fuel-Type | Gear Box |
|----------|----------------|-------|---------------------|-------------------|-----------|----------|
| SED-001 | Corolla Gr... | 2021 | 1798 cc | 138 hp @ 6400 RPM | Petrol | 6 speed |
| SED-002 | Corolla Alt... | 2021 | 1598 cc | 120 hp @ 6000 RPM | Petrol | 5 speed |
| SED-003 | Corolla Gli | 2021 | 1299 cc | 84 hp @ 6000 RPM | Petrol | 5 speed |
| SED-004 | Corolla Xli | 2021 | 1299 cc | 84 hp @ 6000 RPM | Petrol | 5 speed |
| SED-005 | Corolla | 2021 | 1299 cc | 107 hp @ 6000 RPM | Petrol | 5 speed |

Enter the code of a car:

Car Code :

2

Select the color of your car :

- Red
- Blue
- Black
- Silver
- White
- Yellow
- Gray

Please Select Accessories if you want :

 HID Lights Alloy Rims

Welcome , zahir9048

DATE : 01/22/22 [Logout](#)

Please Select Car type :

HATCHBACK **HATCHBACK**

| Car Code | Car Name | Model | Engine Displacem... | Engine Power | Fuel-Type | Gear Box | Transmission-Type | Seating Capacity | Top-Speed | Price_RS |
|----------|------------|-------|---------------------|------------------|-----------|----------|-------------------|------------------|-----------|----------|
| HB-001 | Aqua | 2021 | 1496 cc | 72 hp @ 4800 RPM | Hybrid | CVT | Automatic(CVT) | 5 persons | 180 KM/H | 3500000 |
| HB-002 | Vitz Jewel | 2017 | 996 cc | 71 hp @ 6000 RPM | Petrol | 5 Speed | Automatic | 5 persons | 180 KM/H | 2500000 |
| HB-003 | Vitz | 2021 | 996 cc | 68 hp @ 6000 RPM | Petrol | CVT | Automatic | 5 persons | 180 KM/H | 2700000 |

Enter the code of a car:

Car Code :

Select the color of your car :

Red Blue Black Silver White Yellow Gray

Please Select Accessories if you want :

HID Lights Alloy Rims

Add your car to purchase :

| Date | Car Code | Car Name | Engine | Fuel-Type | Colour | Accessories | Price |
|----------|----------|------------|---------|-----------|--------|-------------------------|----------|
| 01/22/22 | SUV-003 | Prado | 2693 cc | Petrol | Black | HID Lights & Alloy Rims | 27930000 |
| 01/22/22 | HB-002 | Vitz Jewel | 996 cc | Petrol | Gray | HID Lights | 2500000 |

Add to Cart **Delete** **Clear** **Purchase**

Enter car code, select car color, and extra accessories of the car you want to purchase

Selecting the car and its specifications, then clicking on Add to Cart button

Welcome , zahir9048

DATE : 01/22/22 [Logout](#)

Please Select Car type :

SUV **SUV**

| Car Code | Car Name | Model | Engine Displacem... | Engine Power | Fuel-Type | Gear Box | Transmission-Type | Seating Capacity | Top-Speed | Price_RS |
|----------|----------|-------|---------------------|-------------------|-----------|----------|-------------------|------------------|-----------|----------|
| SUV-001 | Fortuner | 2021 | 2694 cc | 164 hp @ 5200 RPM | Petrol | 6 speed | Automatic | 7 persons | 200 KM/H | 7699000 |
| SUV-002 | Revo | 2020 | 2755 cc | 175 hp @ 3400 RPM | Diesel | 6 speed | Automatic | 7 persons | 200 KM/H | 6342000 |
| SUV-003 | Prado | 2020 | 2693 cc | 160 hp @ 5200 RPM | Petrol | 5 speed | Automatic | 7 persons | 180 KM/H | 27930000 |
| SUV-004 | Vigo | 2020 | 2494 cc | 100 hp @ 3600 RPM | Diesel | 4 speed | Automatic | 7 persons | 200 KM/H | 4200000 |
| SUV-005 | Prado | 2021 | 1492 cc | 100 hp @ 3600 RPM | Petrol | 6 speed | Automatic | 7 persons | 200 KM/H | 5000000 |

Enter the code of a car:

Car Code :

Select the color of your car :

Red Blue Black Silver White Yellow Gray

Please Select Accessories if you want :

HID Lights Alloy Rims

Add your car to purchase :

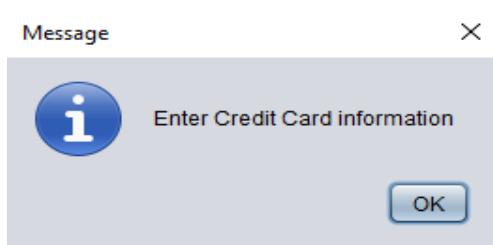
| Date | Car Code | Car Name | Engine | Fuel-Type | Colour | Accessories | Price |
|----------|----------|----------|---------|-----------|--------|-------------------------|----------|
| 01/22/22 | SUV-003 | Prado | 2693 cc | Petrol | Black | HID Lights & Alloy Rims | 27930000 |

Add to Cart **Delete** **Clear** **Purchase**

Adding another car in the Cart

| Add your car to purchase : | | | | | |
|-----------------------------|----------|------------------------|---------|-----------------------|--------|
| Date | Car Code | Car Name | Engine | Fuel-Type | Colour |
| 01/23/22 | SED-002 | Corolla Altis 1.6 | 1598 cc | Petrol | Blue |
| 01/23/22 | SUV-001 | Fortuner | 2694 cc | Petrol | Silver |
| | | | | | |
| Add to Cart | | Delete | | Clear | |
| Purchase | | | | | |

Delete button can delete car from Cart if a car is selected mistakenly



After clicking on Purchase, this message shows

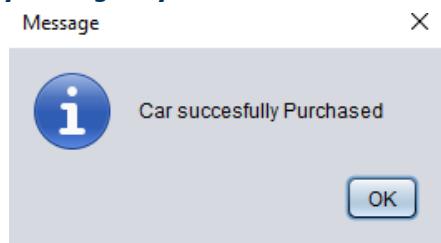
Toyota Car Sales System

Credit Card Information

TOYOTA

| | | | |
|--------------------------|------------------|---|-----------|
| DATE : | 01/22/22 | User Name : | zahir9048 |
| Card Type : | Visa Card | Visa Card | |
| Card Number : | 5823177530016592 | | |
| MM/YY : | 12/2000 | CVV : | 341 |
| CheckOut | |   | |

Clicking on Ok directs you to give your credit card details to complete transaction



Car has been purchased

Toyota Car Sales System - X


TOYOTA

Customer's Bill

DATE : User Name :

=====

TOYOTA CAR SALES SYSTEM

=====

LUCKY-ONE MALL
TOYOTAL MOTORS
KARACHI
NTN No : 0698064-3

=====

User Name :zahir9048
Date :01/22/22
Invoice No : LOM-LM-T3-24805

=====

Car Name 1: Prado
Car Code : SUV-003
Engine : 2693 cc
Colour : Black
Price : 27930000

=====

Car Name 2: Vitz Jewela

Print

After clicking on Ok, the total bill of customer shows along with print button

=====
Car Name 1: Prado
Car Code : SUV-003
Engine : 2693 cc
Colour : Black
Price : 27930000
=====

Car Name 2: Vitz Jewela
Car Code : HB-002
Engine : 996 cc
Colour : Gray
Price : 2500000
=====

Total Price : 30430000
Tex per Car: 5%
Total Amount plus Tex : 31951500
=====

=====
TERMS & CONDITIONS

1. No Cash Refund.
 2. Total price inclusive of all applicable taxes.
 3. Cars on "Sales" can not be exchanged.
 4. Kindly check the car at the time of purchase.
 5. Sale Tax can be verified by using FBR invoice number at www.e.fbr.gov.pk after 24 Hrs from time of invoice generated.
- =====

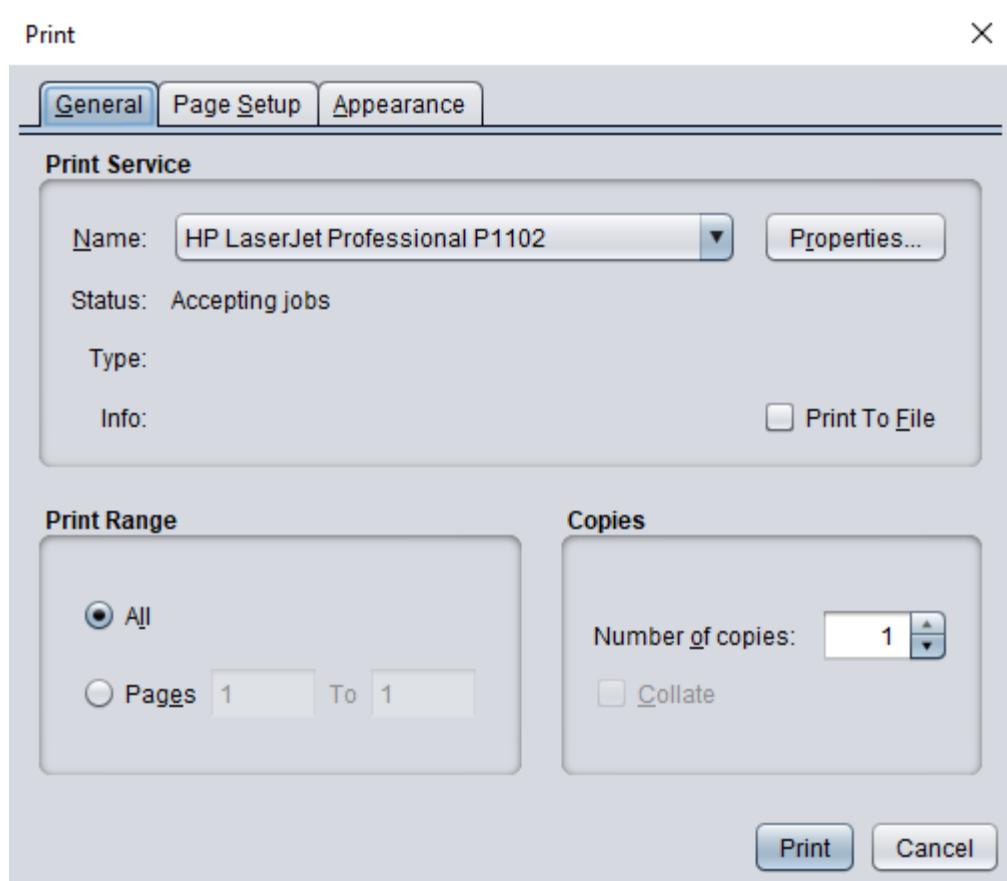
THANKS FOR PURCHASING

=====
Info: 0312-2552726
www.ToyotaCarSales.com
=====

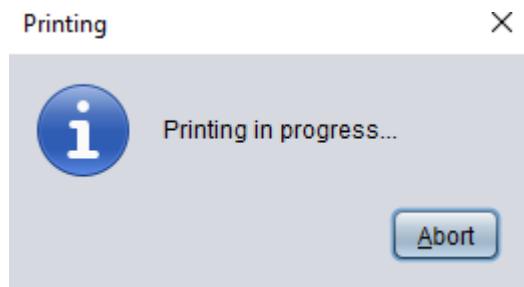
FBR Invoice Number : 11166821122017582706

Verify this invoice through FBR TaxAsaan MobileApp
SMS at 9966 and win exciting prizes in Draw

=====
Customer Total Bill Receipt



You can print your bill receipt instantly using Print button



Printing...

Toyota Car Sales System - Purchase a Car

DATE : 01/22/22

Welcome , zahir9048

Logout

Please Select Car type :

Select

| Car Code | Car Name | Model | Engine Displacem... | Engine Power | Fuel-Type | Gear Box | Transmission-Type | Seating Capacity | Top-Speed | Price_RS |
|----------|----------|-------|---------------------|--------------|-----------|----------|-------------------|------------------|-----------|----------|
| | | | | | | | | | | |

Enter the code of a car:

Car Code :

Select the color of your car :

Red Blue Black Silver White Yellow Gray

Please Select Accessories if you want :

HID Lights Alloy Rims

Add your car to purchase :

| Date | Car Code | Car Name | Engine | Fuel-Type | Colour | Accessories | Price |
|------|----------|----------|--------|-----------|--------|-------------|-------|
| | | | | | | | |

After printing is done, customer is directed to the previous window

Welcome To Toyota Car Sales System

Sign In

Need Help?

TOYOTA

Username :

Password :

If you did not signed up before then, Click here to create a new account

*Customer returns to Sign In page after clicking "log out"

Toyota Car Sales System - Frequently Asked Questions - X

Frequently Asked Questions-FAQs

TOYOTA

Most sold car of 2022 ?

Work hours of our company ?

Where is our company located ?

Is Insurance available for Toyota car...

How to buy car through this software ?

How to Contact our company ?

Subscribe to our Newsletter using your Email

Having bad experience? File a complaint !

Return to SignIn Window

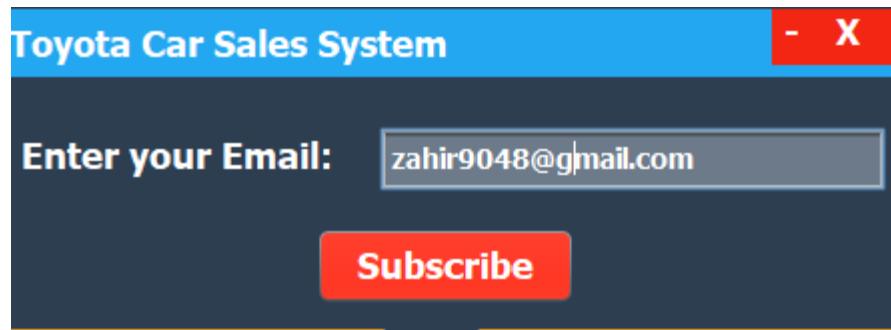
Help button shows some questions, pressing on them will show answers

Where is our company located ?

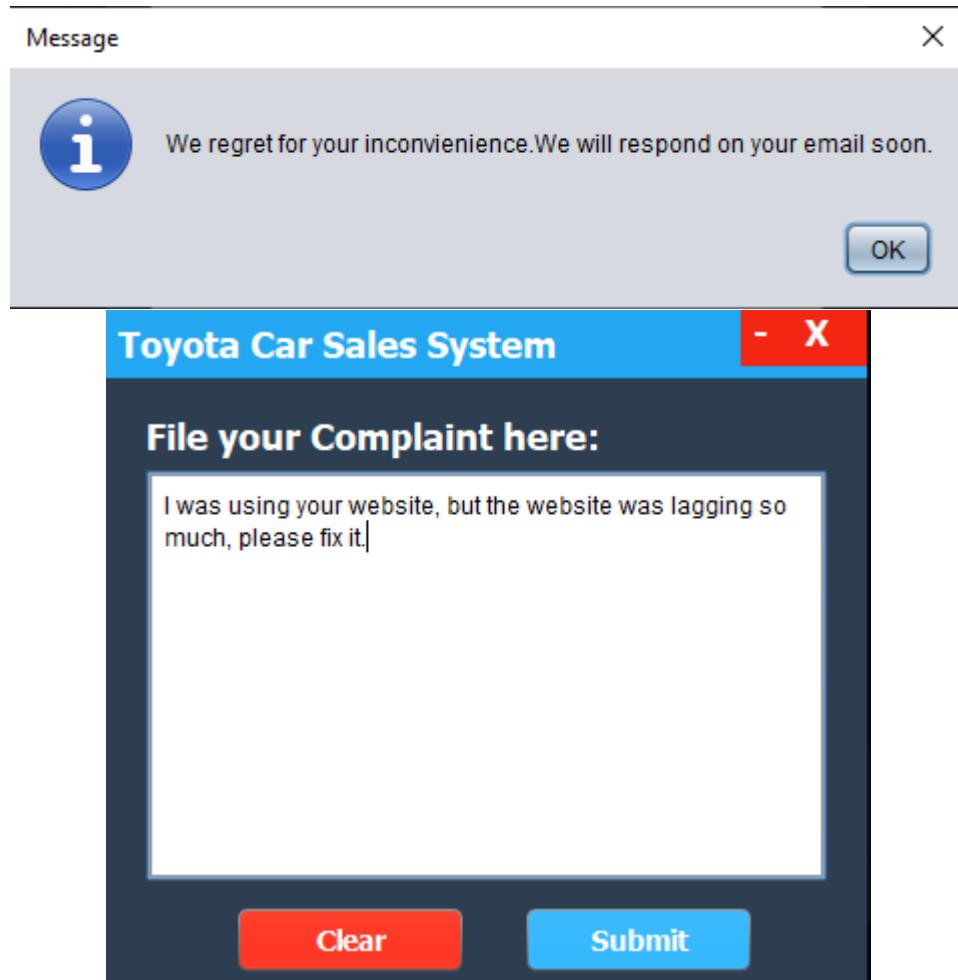
Defence Block 6, Karachi

How to Contact our company ?

Our Contact Numbers are: 0336-4855036,0331-3114993

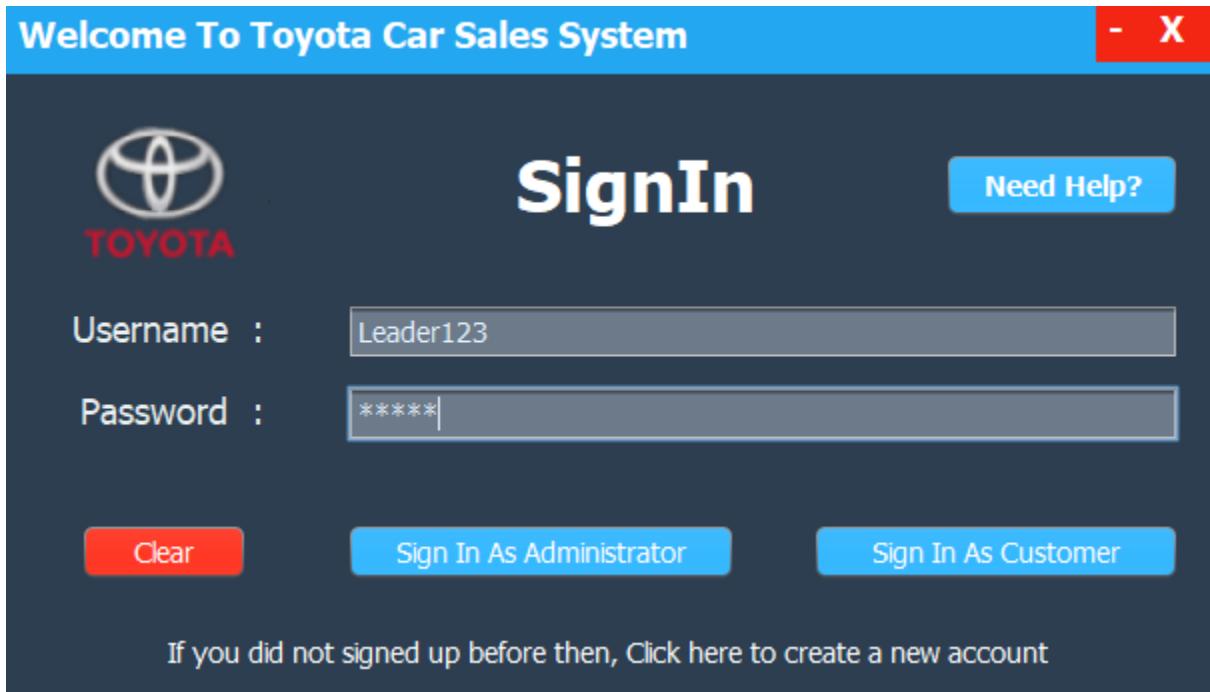


Customer can subscribe to the email subscription service of company by entering email here

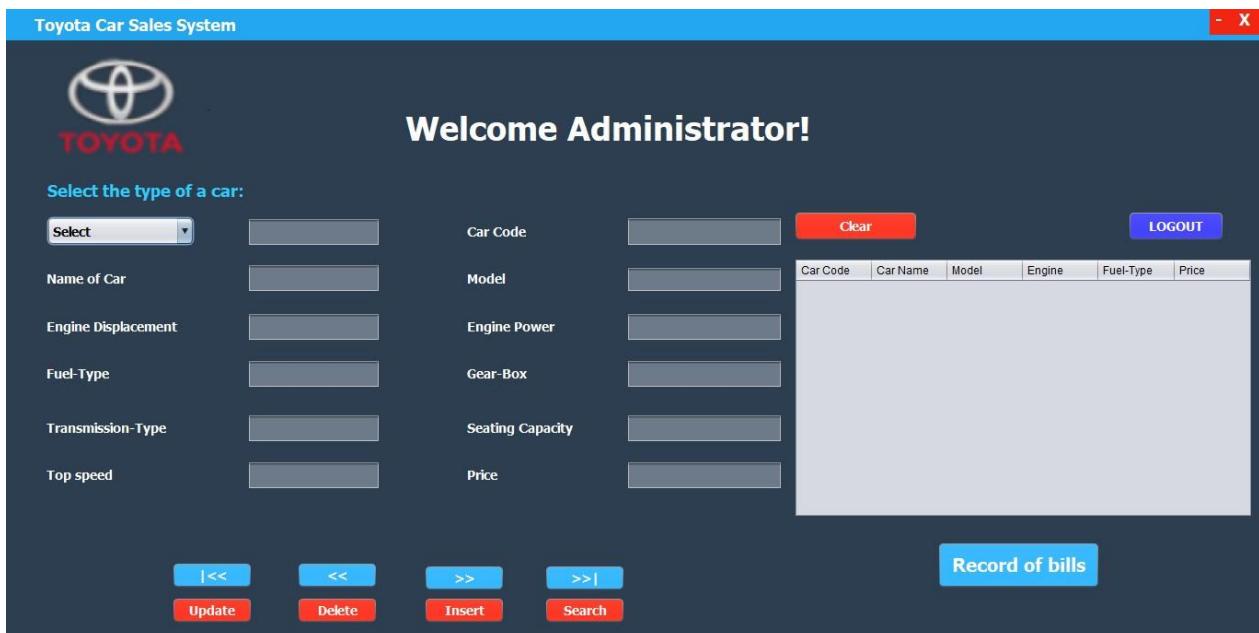


Customer can file complain if he/she doesn't like something on the company's site/app or car

Admin Sign In:



Singing in as admin by entering admin credentials



Admin window after logging in



Admin can select between different categories of car easily

Toyota Car Sales System

Welcome Administrator!

Select the type of a car:

| | | | | |
|---------------------|-------|------------------|-------|--------|
| SEDAN | SEDAN | Car Code | Clear | LOGOUT |
| Name of Car | | Model | | |
| Engine Displacement | | Engine Power | | |
| Fuel-Type | | Gear-Box | | |
| Transmission-Type | | Seating Capacity | | |
| Top speed | | Price | | |

Car Code | Car Name | Model | Engine | Fuel-Type | Price

| | | | | | |
|---------|-----------------|------|---------|--------|---------|
| SED-001 | Corolla Gra... | 2021 | 1798 cc | Petrol | 4000000 |
| SED-002 | Corolla Alti... | 2021 | 1598 cc | Petrol | 3800000 |
| SED-003 | Corolla Gli | 2021 | 1299 cc | Petrol | 3500000 |
| SED-004 | Corolla Xli | 2021 | 1299 cc | Petrol | 3300000 |
| SED-005 | Camry | 2021 | 2362 cc | Petrol | 9000000 |
| SED-006 | Yaris | 2021 | 1329 cc | Petrol | 2800000 |
| SED-007 | Prius | 2021 | 1798 cc | Hybrid | 7500000 |
| SED-008 | Premio | 2021 | 1496 cc | Petrol | 1000000 |
| SED-009 | Axio | 2021 | 1496 cc | Petrol | 5500000 |

| << << >> >> | Record of bills

Update Delete Insert Search

| Car Code | Car Name | Model | Engine | Fuel-Type | Price |
|----------|-----------------|-------|---------|-----------|---------|
| SED-001 | Corolla Gra... | 2021 | 1798 cc | Petrol | 4000000 |
| SED-002 | Corolla Alti... | 2021 | 1598 cc | Petrol | 3800000 |
| SED-003 | Corolla Gli | 2021 | 1299 cc | Petrol | 3500000 |
| SED-004 | Corolla Xli | 2021 | 1299 cc | Petrol | 3300000 |
| SED-005 | Camry | 2021 | 2362 cc | Petrol | 9000000 |
| SED-006 | Yaris | 2021 | 1329 cc | Petrol | 2800000 |
| SED-007 | Prius | 2021 | 1798 cc | Hybrid | 7500000 |
| SED-008 | Premio | 2021 | 1496 cc | Petrol | 1000000 |
| SED-009 | Axio | 2021 | 1496 cc | Petrol | 5500000 |

Table which shows all the cars of the selected categories from the SQL database

Select the type of a car:

| | | | |
|---------------------|-------------------|------------------|-------------------|
| SEDAN | SEDAN | Car Code | SED-002 |
| Name of Car | Corolla Altis 1.6 | Model | 2021 |
| Engine Displacement | 1598 cc | Engine Power | 120 hp @ 6000 RPM |
| Fuel-Type | Petrol | Gear-Box | 5 speed |
| Transmission-Type | Automatic | Seating Capacity | 5 persons |
| Top speed | 240 KM/H | Price | 3800000 |

|<< << >> >>|

Update Delete Insert Search

Select the type of a car:

| | | | |
|---------------------|-------------|------------------|------------------|
| SEDAN | SEDAN | Car Code | SED-003 |
| Name of Car | Corolla Gli | Model | 2021 |
| Engine Displacement | 1299 cc | Engine Power | 84 hp @ 6000 RPM |
| Fuel-Type | Petrol | Gear-Box | 5 speed |
| Transmission-Type | Manual | Seating Capacity | 5 persons |
| Top speed | 240 KM/H | Price | 3500000 |

|<< << >> >>|

Update Delete Insert Search

Navigation between different cars using navigate buttons

Search Operation:

Select the type of a car:

| | | | |
|---------------------|-----------|------------------|-------------------|
| SUV | SUV | Car Code | SUV-003 |
| Name of Car | Prado | Model | 2020 |
| Engine Displacement | 2693 cc | Engine Power | 160 hp @ 5200 RPM |
| Fuel-Type | Petrol | Gear-Box | 5 speed |
| Transmission-Type | Automatic | Seating Capacity | 7 persons |
| Top speed | 180 KM/H | Price | 27930000 |

|<< << >> >>|

Update Delete Insert Search

Searching of cars by Car Code using cars using Search Button

Update Operation:

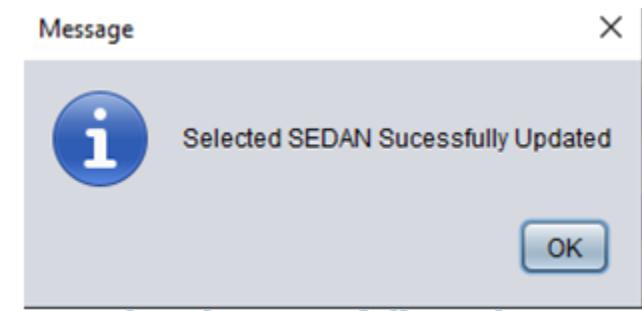
Select the type of a car:

| | | | |
|---------------------|-------------|------------------|------------------|
| SEDAN | SEDAN | Car Code | SED-003 |
| Name of Car | Corolla GLi | Model | 2021 |
| Engine Displacement | 1299 cc | Engine Power | 84 hp @ 6000 RPM |
| Fuel-Type | Petrol | Gear-Box | 5 speed |
| Transmission-Type | Manual | Seating Capacity | 5 persons |
| Top speed | 240 KM/H | Price | 3500000 |

| << << >> >> |

Update Delete Insert Search

Update Button updates the car specifications, here transmission type is updated to manual from auto



Car updated successfully in the system

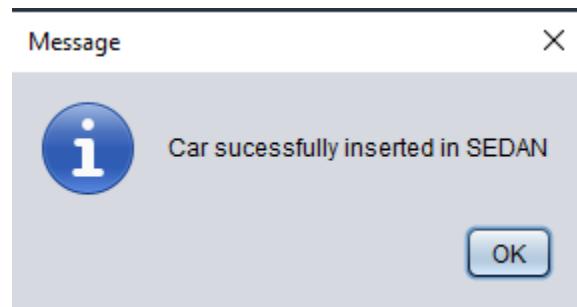
Insert Operation:

Select the type of a car:

| | | | |
|---------------------|-------------|------------------|------------------|
| SEDAN | SEDAN | Car Code | 10 |
| Name of Car | Corolla Xli | Model | 2020 |
| Engine Displacement | 1299 cc | Engine Power | 90 hp @ 6000 RPM |
| Fuel-Type | CNG | Gear-Box | 5 speed |
| Transmission-Type | Manual | Seating Capacity | 5 persons |
| Top speed | 260 KM/H | Price | 3000000 |

| << << >> >> |

Update Delete Insert Search

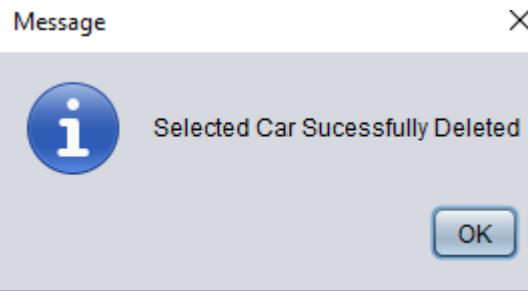


Car Inserted Successfully in the system

Delete Operation:

| Car Code | Car Name | Model | Engine | Fuel-Type | Price |
|----------|-----------------|-------|---------|-----------|---------|
| SED-001 | Corolla Gra... | 2021 | 1798 cc | Petrol | 4000000 |
| SED-002 | Corolla Alti... | 2021 | 1598 cc | Petrol | 3800000 |
| SED-003 | Corolla Gli | 2021 | 1299 cc | Petrol | 3500000 |
| SED-004 | Corolla Xli | 2021 | 1299 cc | Petrol | 3300000 |
| SED-005 | Camry | 2021 | 2362 cc | Petrol | 9000000 |
| SED-006 | Yaris | 2021 | 1329 cc | Petrol | 2800000 |
| SED-007 | Prius | 2021 | 1798 cc | Hybrid | 7500000 |
| SED-008 | Premio | 2021 | 1496 cc | Petrol | 1000000 |
| SED-009 | Axio | 2021 | 1496 cc | Petrol | 5500000 |
| SED-0010 | Corolla Xli | 2020 | 1299 cc | CNG | 3000000 |

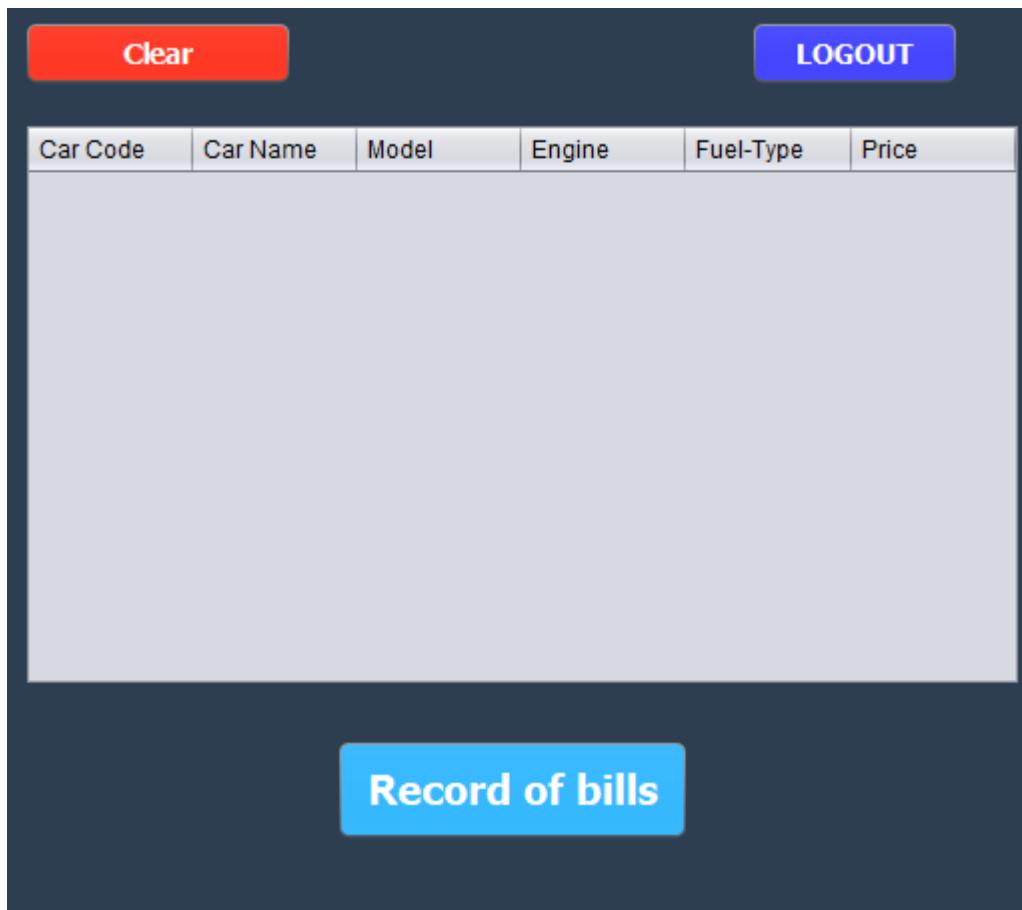
| Car Code | Car Name | Model | Engine | Fuel-Type | Price |
|----------|-----------------|-------|---------|-----------|---------|
| SED-001 | Corolla Gra... | 2021 | 1798 cc | Petrol | 4000000 |
| SED-002 | Corolla Alti... | 2021 | 1598 cc | Petrol | 3800000 |
| SED-003 | Corolla Gli | 2021 | 1299 cc | Petrol | 3500000 |
| SED-004 | Corolla Xli | 2021 | 1299 cc | Petrol | 3300000 |
| SED-005 | Camry | 2021 | 2362 cc | Petrol | 9000000 |
| SED-006 | Yaris | 2021 | 1329 cc | Petrol | 2800000 |
| SED-007 | Prius | 2021 | 1798 cc | Hybrid | 7500000 |
| SED-008 | Premio | 2021 | 1496 cc | Petrol | 1000000 |
| SED-009 | Axio | 2021 | 1496 cc | Petrol | 5500000 |
| SED-0010 | Corolla Xli | 2020 | 1299 cc | CNG | 3000000 |



Car deleted successfully from the system

| Car Code | Car Name | Model | Engine | Fuel-Type | Price |
|----------|------------------|-------|---------|-----------|---------|
| SED-001 | Corolla Gra... | 2021 | 1798 cc | Petrol | 4000000 |
| SED-002 | Corolla Altis... | 2021 | 1598 cc | Petrol | 3800000 |
| SED-003 | Corolla Gli | 2021 | 1299 cc | Petrol | 3500000 |
| SED-004 | Corolla Xli | 2021 | 1299 cc | Petrol | 3300000 |
| SED-005 | Camry | 2021 | 2362 cc | Petrol | 9000000 |
| SED-006 | Yaris | 2021 | 1329 cc | Petrol | 2800000 |
| SED-007 | Prius | 2021 | 1798 cc | Hybrid | 7500000 |
| SED-008 | Premio | 2021 | 1496 cc | Petrol | 1000000 |
| SED-009 | Axio | 2021 | 1496 cc | Petrol | 5500000 |

Deleted car will no longer be shown in the table



Record of Bills Button shows all records of all customers who bought cars

Toyota Car Sales System - X

 **Bills Of Customer**

| Date | User Name | Car Code | Car Name | Engine | Colour | Prize |
|----------|-----------|----------|----------------|---------|--------|----------|
| 01/21/22 | zahir9048 | SED-001 | Corolla Gra... | 1798 cc | Black | 4000000 |
| 01/21/22 | zahir9048 | SED-004 | Corolla Xli | 1299 cc | Silver | 3300000 |
| 01/22/22 | zahir9048 | SUV-003 | Prado | 2693 cc | Black | 27930000 |
| 01/22/22 | zahir9048 | HB-002 | Vitz Jewela | 996 cc | Gray | 2500000 |

Clear Record

[click here to return to admin window](#)

Records of customers who bought cars, clear button deletes all the records of customers from the system

LOGOUT

Logout button logs out the customer/admin and directs to the login window

Clear

Clear button clears all the text fields

CONCLUSION

In this project, Car Sales System we have made 2 separate systems, one for customer and one for Admin. In admin section, we can perform different operations like adding, modifying, deleting, navigating, and searching different cars and their specifications using SQL database. While in customer system, customer can view any car and can buy his/her favorite car easily with our system. Customer can also file a complaint using our System. Our project's aim is to provide a solution that is feasible to both the consumer and producer. A software which producer and customer both can understand easily. In future, we hope that more and more software's/websites like these could launch so that the people would have best of cars buying experience

FUTURE SCOPE

In the existing System it is difficult to maintain the car information individually and to supply for the customers who are eager to buy them. Customer must face difficulty to know the information of car like manufacturing year, car model and other valuable information in a single domain. Our main idea is to develop a database system where we can have all the required information for the user to effectively interest him in the process of buying a car. In the Proposed System, application can maintain car details like manufacturer, year of manufacturing, price and model etc. We can also view all the car details which are kept for sale effectively and we can search for our desired car. With this Customer can get the information quickly like car details which have been entered clearly.

The main goal of this project is designing and developing an interactive Car Sales System. These details are routinely maintained depending on the numerous inputs and the changes made in the systems database. The alterations in the database can happen spontaneously, for example, after the system performs calculations on the direct inputs. This system essentially works on input-output entry mechanism. The Car Sales System follows closely with other developed Information Systems focusing on achieving better transactions and more customer satisfaction