



# PRG455: PROJECT MANUAL

STUDENT NAME: ARPIT SONDHI

PROFESSOR'S NAME: GORAN SVENK

COURSE CODE: PRG455

SECTION: NAB

DATE: APRIL 01, 2023

## Table of Contents

Introduction: Car Rental System .....	3
Form 1: Car Rental Main Screen .....	4
Fig.1 When all 6 cars are in stock. ....	5
Fig.2 When one of the cars is out of stock. ....	5
Form 2: Cars Information .....	6
Fig.3 Showing Car details with 1 <sup>st</sup> Picture. ....	7
Fig.4 Showing Car details with 2 <sup>nd</sup> Picture when the right arrow button is clicked. ....	7
Form 3: Book Your Car .....	8
Fig.5 Showing Create Booking form, with all fields filled accurately. ....	9
Fig.6 Showing Message Box error when Card number doesn't contain 16 digits. ....	10
Fig.7 Showing Message Box error when Phone number doesn't contain 10 digits. ....	10
Fig.8 Showing Message Box error when the date incorrect/past date is selected. ....	11
Fig.9 Showing Message Box error when Phone number contains anything except digits. ....	11
Fig.10 Showing Message Box error when Card number contains anything except digits. ....	12
Fig.11 Showing the 'Submit Order' button is disabled after it's clicked. ....	12
Form 4: View Receipt .....	13
Fig.12 Showing the Receipt. ....	13
Table1: CarsData .....	14
Table2: CustomerInfo .....	14
Table3: Bookings .....	15
Code: Form1 .....	16
Code: FormCarInfo .....	18
Code: FormBooking .....	22
Code: FormReceipt .....	27

## Introduction: Car Rental System

I have designed a car rental C sharp project using Windows Forms which enables the customers to rent their desirable cars. It uses access database to store information related to cars, customers and booking. The program has 4 forms:

1. Car Rental Main Screen
2. Cars Information
3. Book Your Car
4. View Receipt

The Access database used to store information has 3 tables:

1. CarsData
2. CustomerInfo
3. Bookings

In this manual, I will explain the working of the whole C sharp Car Rental application using 4 Windows Forms and a Database containing 3 tables.

## Form 1: Car Rental Main Screen

When you start the program, you can see the first windows form in front of you which is the main screen of the application. This form has 2 labels, 1 picture box and 7 buttons by default.

- One label is used to show the main title of the program, which is the company name. The other label is used to prompt the user to select their desired car.
- The picture box is used for the wallpaper of the application.
- One button is used to close the application on the top right corner of the form marked by 'X'. The other 6 buttons are made by the code which is connected to the database containing the number of cars in the 'CarsData' table. For example, if there are 6 cars in the database, then the program will create and show 6 buttons containing the respective car names. However, if the quantity of any car becomes zero, the button wouldn't be created. This program has been designed to show at most 6 car buttons because of the size and location of the buttons and to meet the project requirements.

When the user clicks on any button with car names, the second form will open showing the respective car's information.

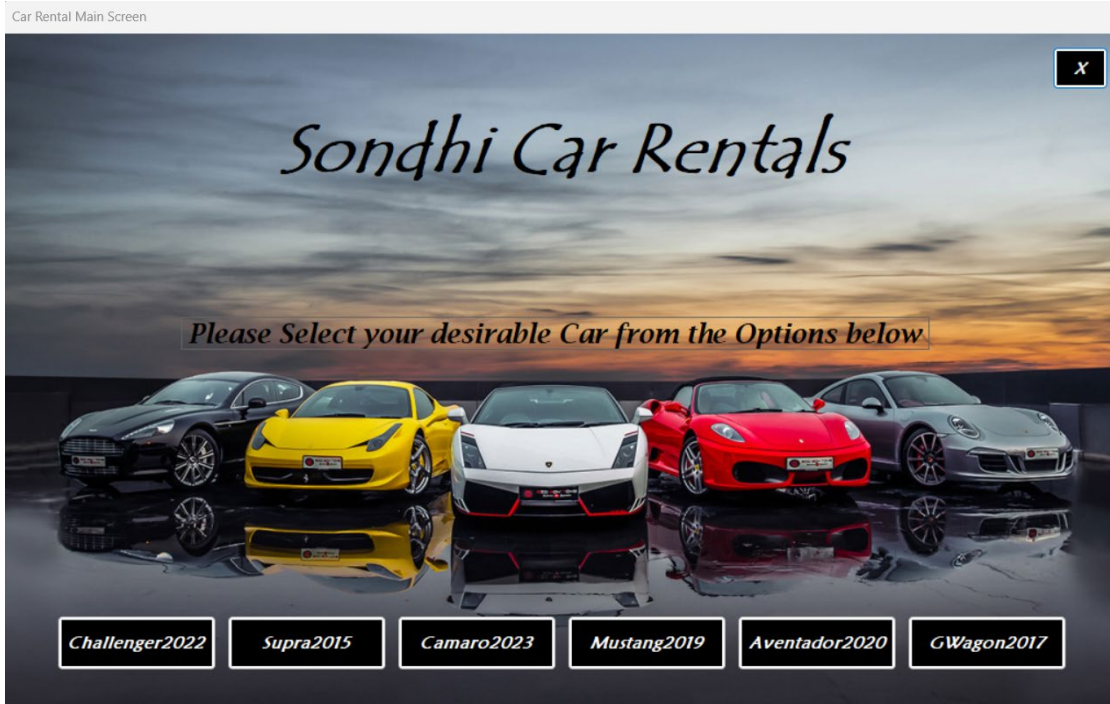


Fig.1 When all 6 cars are in stock.

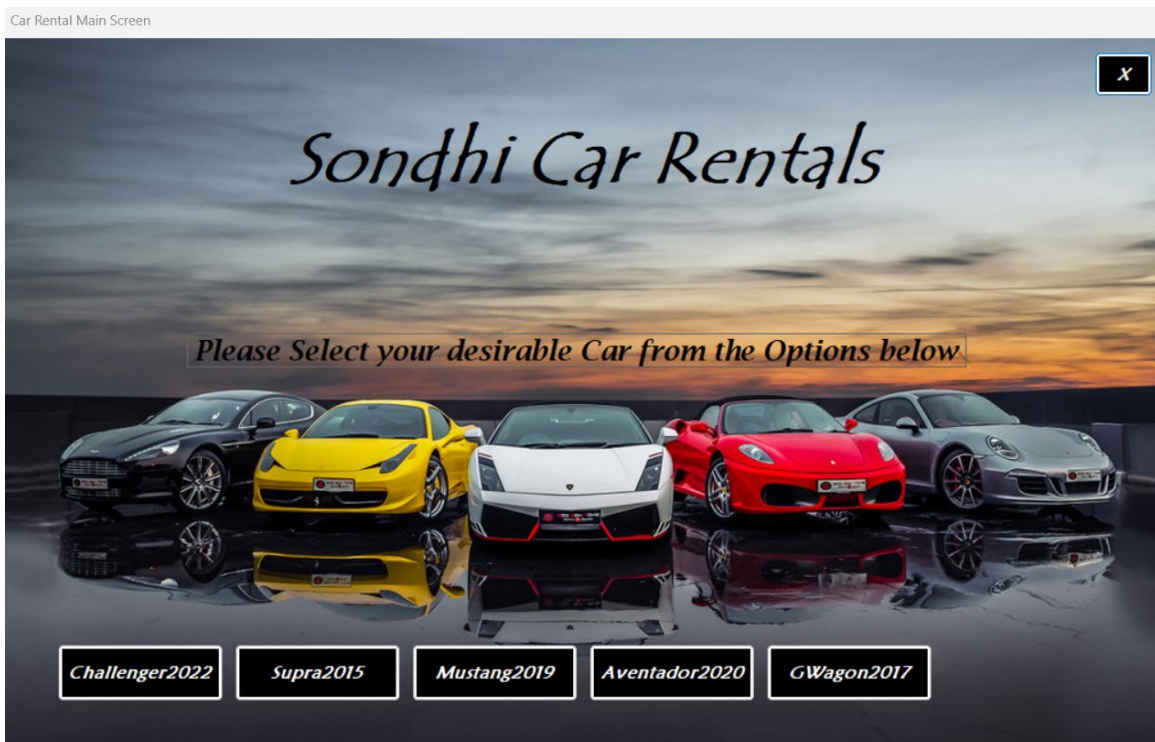


Fig.2 When one of the cars is out of stock.



## Form 2: Cars Information

The second form shows the information about the car which the user/customer has selected to rent. I have used 5 labels, 1 picture box, and 5 buttons in this form.

- First label is used to display the title 'Car Details'. The second label displays the Car Type and Car Model, whose data is taken from the database containing the 'CarsData' table. The third label is used to display the rent price of the car per day, whose data is also taken from the 'CarsData' table, in case the price is to be updated. The fourth label is used to display the features and other information respective to the car. The last label is just used in designing the background.
- The picture box is used to display the car pictures.
- There are 2 arrow buttons (<, >) used to show 2 car pictures in the picture box, back and forth. The third button is the close (X) button, used to close the application. The fourth button (Back to Main Menu) is used to go back to the first form. The last button (Book) is used to book the car and it takes the user to the third form.

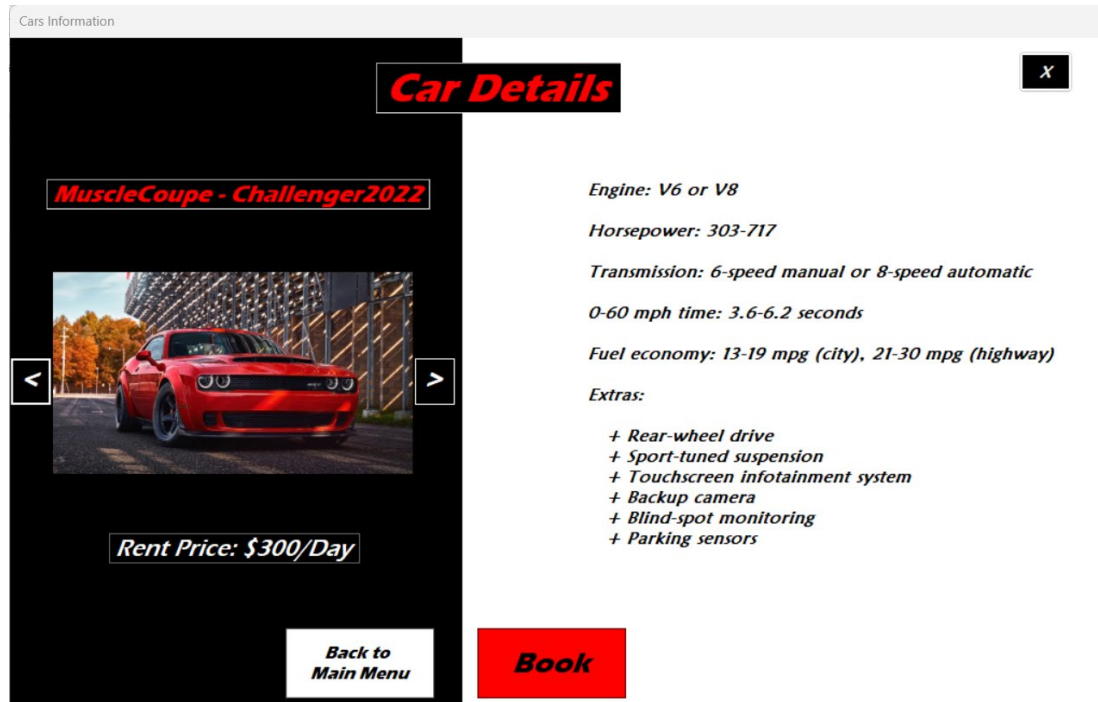


Fig.3 Showing Car details with 1<sup>st</sup> Picture.

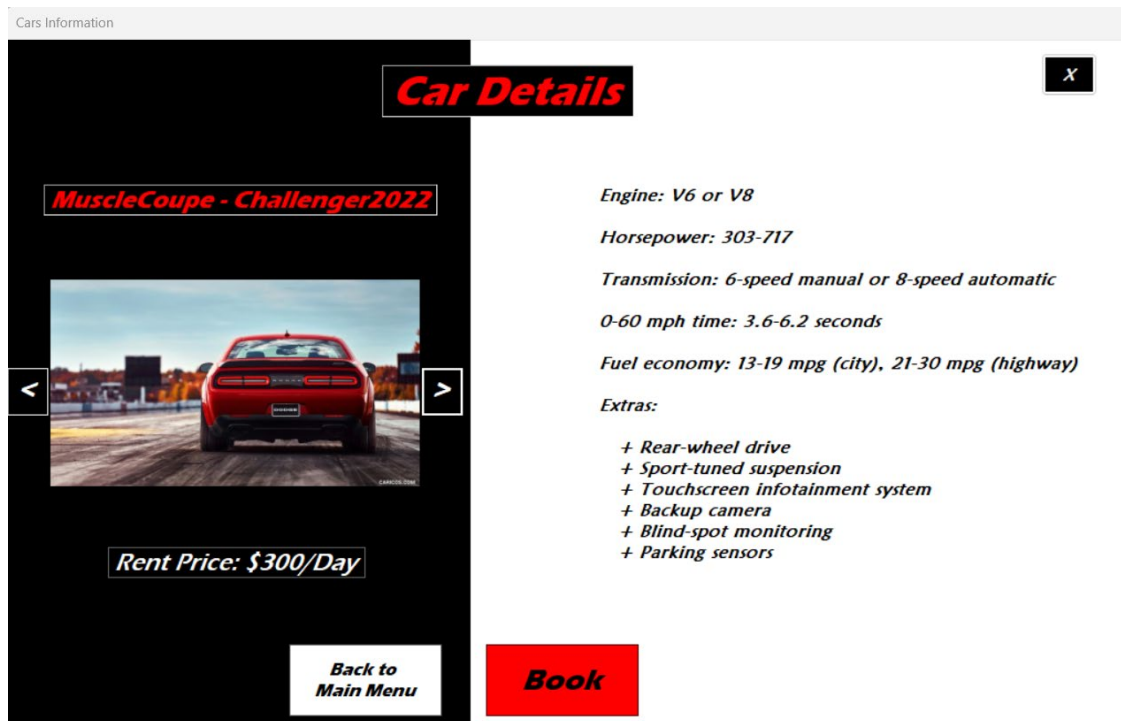


Fig.4 Showing Car details with 2<sup>nd</sup> Picture when the right arrow button is clicked.

## Form 3: Book Your Car

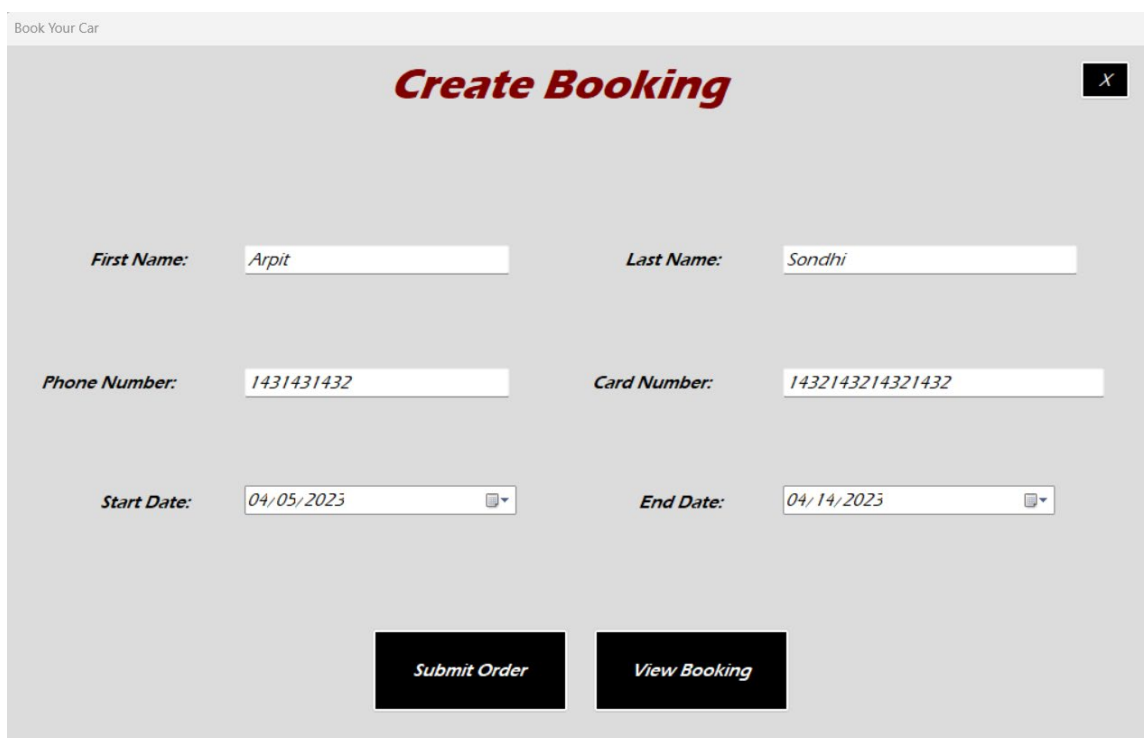
When the user clicks on the 'Book' option on the second form, it opens the 3<sup>rd</sup> form which is used to create a booking. This form uses 7 labels, 4 text boxes, 2 date/time picker tools, and 3 buttons.

- First label is used to display the title 'Create Booking'. There are 6 other labels used to prompt the user to choose/enter their information which include: First name, Last name, Phone number, Card number, Start date and End date.
- There are 4 textboxes in the form for the user to input their details including their First name, Last name, Phone number and Card number.
- There are 2 date/time pickers tools in the form for the user to select their car pick up and drop off dates.
- The form also has 3 buttons. The first button (X) is to close the application. The second button (Submit Order) is used to save the information entered by the customer to the 'CustomerInfo' table in the database. The third button (View Booking) is used to view the third form containing the booking receipt. This form also handles the exceptions and doesn't allow the user to leave any field empty and show the respective error in the form of a message box. Moreover, it checks for 10-digit phone numbers and 16 digits card numbers and shows the respective message boxes for the same. The program also checks for the wrong pickup and drop-off dates and doesn't allow the user to select past dates as well as the cases where the drop-off



date is set to before the pickup date. It also shows the respective errors for the same in the form of message boxes.

When the user enters the correct data and clicks on the 'Submit Order' button, the program shows 2 message boxes which remark that the data has been saved in both the tables of the database: 'CustomerInfo' and 'Bookings'. When both the booking and customer data are saved successfully, the 'Submit Order' button is disabled. Then, the user can only click on the 'View Booking' button to open the fourth form.



Book Your Car

## Create Booking

**First Name:**  **Last Name:**

**Phone Number:**  **Card Number:**

**Start Date:**  **End Date:**

**Submit Order** **View Booking**

Fig.5 Showing Create Booking form, with all fields filled accurately.

Book Your Car

## Create Booking

**First Name:**  **Last Name:**

**Phone Number:**

**Start Date:**  **End Date:**

Please enter a 16-digit card number.

OK

Fig.6 Showing Message Box error when Card number doesn't contain 16 digits.

Book Your Car

## Create Booking

**First Name:**  **Last Name:**

**Phone Number:**

**Start Date:**  **End Date:**

Please enter a 10-digit phone number.

OK

Fig.7 Showing Message Box error when Phone number doesn't contain 10 digits.

Book Your Car

## Create Booking

**First Name:**  **Last Name:**

**Phone Number:**

**Start Date:**  **End Date:**

Invalid pickup date. Please select a date on or after today's date.

OK

**Submit Order** **View Booking**

Fig.8 Showing Message Box error when the date incorrect/past date is selected.

Book Your Car

## Create Booking

**First Name:**  **Last Name:**

**Phone Number:**

**Start Date:**  **End Date:**

Please enter valid phone number.

OK

**Submit Order** **View Booking**

Fig.9 Showing Message Box error when Phone number contains anything except digits.

Book Your Car

## Create Booking

**First Name:**  **Last Name:**

**Phone Number:**  **Card Number:**

**Start Date:**  **End Date:**

Please enter valid card number.

OK

**Submit Order** **View Booking**

Fig.10 Showing Message Box error when Card number contains anything except digits.

Book Your Car

## Create Booking

**First Name:**  **Last Name:**

**Phone Number:**  **Card Number:**

**Start Date:**  **End Date:**

**Submit Order** **View Booking**

Fig.11 Showing the 'Submit Order' button is disabled after it's clicked.

## Form 4: View Receipt

The fourth form is used to display the customer receipt and has 23 labels and 1 button.

- First label is used to display the title of the receipt with company's name 'Sondhi Car Receipt'. There are 11 labels to display the receipt data and their 11 corresponding description texts: First name, Last name, Phone number, Car ID, Booking ID, Car type, Car model, Pickup Date, Drop off date, Card number, and Total Cost. In addition, only last 4 digits of Card number are shown to protect customer's privacy. Also, the total cost includes all the taxes.
- There is only 1 Close (X) button on the top right corner to exit the application.

View Receipt

**Sondhi Car Rental Receipt** X

<b>First Name:</b> <u>Arpit</u>	<b>Last Name:</b> <u>Sondhi</u>
<b>Phone Number:</b> <u>1431431432</u>	
<b>Car ID:</b> <u>1</u>	<b>Booking ID:</b> <u>62</u>
<b>Car Type:</b> <u>MuscleCoupe</u>	<b>Car Model:</b> <u>Challenger2022</u>
<b>Pickup Date:</b> <u>2023-04-19 12:00:00 AM</u>	<b>DropOff Date:</b> <u>2023-04-27 12:00:00 AM</u>
<b>CardNumber:</b> <u>XXXXXXXXXXXX1432</u>	<b>Total Cost:</b> <u>\$2712</u>

Fig.12 Showing the Receipt.

**Table1: CarsData**

	CarID	CarType	CarModel	StockQuantit	CostPerDay	Click to Add
+	1	MuscleCoupe	Challenger2022	11	\$300.00	
+	2	Sports	Supra2015	3	\$350.00	
+	3	Sedan	Camaro2023	7	\$150.00	
+	4	Sports	Mustang2019	8	\$200.00	
+	5	Sports	Aventador2020	5	\$450.00	
+	6	SUV	GWagon2017	3	\$250.00	
*	(New)			0	\$0.00	

The data in the 'CarsData' table is input by the programmer or the Owner of the application. It can be used to add/remove cars as well as change their related information such as Stock or Cost. CarID field is the primary key and it's auto assigned and incremented with every new car.

**Table2: CustomerInfo**

	CustomerID	FirstName	LastName	TelephoneNumbe	CardNumber	Click to Add
+	61	Arpit	Sondhi	6451234567	5464789745641324	
+	62	asd	xcz	5645456456	5456456456456123	
+	63	nmm	lpp	1123357789	1111222233334444	
+	64	Anshu	SIngh	4546456445	7878787878787877	
+	65	Ibran	Sheikh	1111222232	1111000022225555	
+	66	Harsh	Singh	2727383849	2828383848482929	
+	67	Harshit	Bajaj	1231231231	1231231231231233	
+	68	varun	arora	7777888898	8888666655552233	
*	(New)			0	0	



The data in the 'CustomerInfo' table is updated through Form 3 where user enters his details and click on the 'Submit Order' option. This table is used to store all the customer related information where 'CustomerID' is the primary key and it's auto assigned and incremented each time the new record is added.

## Table3: Bookings

BookingID	CarID	CustomerID	PickupDate	DropOffDate	TotalCost	Click to Add
53	2	62	April 6, 2023	April 8, 2023	\$791.00	
54	5	63	April 5, 2023	April 6, 2023	\$508.50	
55	3	64	April 5, 2023	May 4, 2023	\$4,915.50	
56	2	65	April 5, 2023	April 6, 2023	\$395.50	
57	4	66	April 4, 2023	April 7, 2023	\$678.00	
58	5	67	April 4, 2023	May 5, 2023	\$15,763.50	
59	6	68	April 4, 2023	April 19, 2023	\$4,237.50	
⌘ (New)	0	0			\$0.00	

The date in the 'Bookings' table is also updated through Form 3 where user enters his details and the booking information and then click on the 'Submit Order' button. The program is coded to add the corresponding records for 'CarID' and 'CustomerID' which are both primary keys from the other two tables to create a relation between the tables. BookingID field is the primary key for the table and CarID and CustomerID are the foreign keys.

All three tables are also connected to each other by creating relationship among them to access/update/insert data by the program.

## Code: Form1

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.OleDb;

namespace CarRentalProject
{
    public partial class Form1 : Form
    {
        DataTable vt = new DataTable();
        public Form1()
        {
            InitializeComponent();
        }
        public void dbfun(string str)
        {
            string constr = "Provider=Microsoft.ACE.OLEDB.12.0;" + "Data Source=ProjectTables.accdb";
            string sqlstr = str;
            OleDbDataAdapter da = new OleDbDataAdapter(sqlstr, constr);
            da.Fill(vt);
            da.Dispose();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            int x = 50;
            this.dbfun("SELECT CarModel FROM CarsData WHERE StockQuantity > 0");
            foreach (DataRow row in vt.Rows)
            {
                Button btnCar = new Button();
                btnCar.Text = row["CarModel"].ToString();
                btnCar.Name = "btn" + row["CarModel"].ToString();
                btnCar.Location = new Point(x, 530);
                btnCar.Size = new Size(145, 50);
                btnCar.BackColor = Color.Black;
                btnCar.ForeColor = Color.White;
                btnCar.Click += new EventHandler(carModelBtn_Click);
                this.Controls.Add(btnCar);
                x += 155;
            }
        }
        private void carModelBtn_Click(object sender, EventArgs e)
        {
            Button clickedBtn = (Button)sender;
            string selectedCar = clickedBtn.Text;
        }
    }
}

```

```
        this.Hide();  
        FormCarInfo form2 = new FormCarInfo(selectedCar);  
        form2.Show();  
    }  
  
    private void btnExit_Click(object sender, EventArgs e)  
    {  
        Application.Exit();  
    }  
}
```

## Code: FormCarInfo

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.OleDb;

namespace CarRentalProject
{
    public partial class FormCarInfo : Form
    {
        private string selectedCar;

        public FormCarInfo(string selectedCar)
        {
            InitializeComponent();
            this.selectedCar = selectedCar;
        }

        private void FormCarInfo_Load(object sender, EventArgs e)
        {
            pictureBoxCars.SizeMode = PictureBoxSizeMode.Zoom;

            string constr = "Provider=Microsoft.ACE.OLEDB.12.0;" + "Data Source=ProjectTables.accdb";
            string sqlstr = $"SELECT * FROM CarsData WHERE CarModel='{selectedCar}'";
            OleDbDataAdapter da = new OleDbDataAdapter(sqlstr, constr);
            DataTable dt = new DataTable();
            da.Fill(dt);
            da.Dispose();

            if (dt.Rows.Count > 0)
            {
                DataRow dr = dt.Rows[0];
                lblCarName.Text = dr["CarType"].ToString() + " - " + dr["CarModel"].ToString();
                lblPrice.Text = "Rent Price: $" + dr["CostPerDay"].ToString() + "/Day";
            }
            else
            {
                MessageBox.Show("No records found.");
            }

            switch (selectedCar)
            {
                case "Challenger2022":
                    challenger();
                    break;
                case "Supra2015":
                    supra();
                    break;
                case "Mustang2019":

```

```

        mustang();
        break;
    case "Camaro2023":
        camaro();
        break;
    case "Aventador2020":
        aventador();
        break;
    case "GWagon2017":
        gwagon();
        break;
    default:
        break;
    }
}
private void challenger()
{
    pictureBoxCars.Image = Properties.Resources.Challenger;
    lblFeatures.Text = "Engine: V6 or V8\n\nHorsepower: 303-717\n\nTransmission: 6-speed manual or 8-
speed automatic\n\n0-60 mph time: 3.6-6.2 seconds\n\nFuel economy: 13-19 mpg (city), 21-30 mpg
(highway)\n\nExtras:\n\n + Rear-wheel drive\n + Sport-tuned suspension\n + Touchscreen infotainment
system\n + Backup camera\n + Blind-spot monitoring\n + Parking sensors";
}
private void mustang()
{
    pictureBoxCars.Image = Properties.Resources.Mustang;
    lblFeatures.Text = "Engine: 5.2-liter V8\n\nHorsepower: 526 hp\n\nTransmission: 6-speed manual
\n\n0-60 mph time: 3.5 seconds\n\nFuel economy: 14 mpg (city), 21 mpg (highway)\n\nExtras:\n\n + Rear-
wheel drive\n + Sport-tuned suspension\n + 12-inch digital instrument cluster\n + 8-inch touchscreen
Infotainment system\n + Apple CarPlay/Android Auto\n + Adaptive cruise control\n + Blind-spot
monitoring";
}
private void gwagon()
{
    pictureBoxCars.Image = Properties.Resources.GWagon;
    lblFeatures.Text = "Engine: Turbocharged 4.0-liter V8\n\nHorsepower: 416 hp\n\nTransmission: 9-
speed automatic \n\n0-60 mph time: 5.7 seconds\n\nFuel economy: 13 mpg (city), 17 mpg
(highway)\n\nExtras:\n\n + All-wheel drive\n + Adaptive suspension\n + 12.3-inch Digital Instrument
Cluster\n + Touchscreen Infotainment System\n + Apple CarPlay/Android Auto\n + Adaptive Cruise
Control\n + Blind-spot monitoring";
}
private void aventador()
{
    pictureBoxCars.Image = Properties.Resources.Aventador;
    lblFeatures.Text = "Engine: 6.5-liter V12\n\nHorsepower: 770 hp\n\nTransmission: 7-speed automatic
\n\n0-60 mph time: 2.8 seconds\n\nFuel economy: 9 mpg (city), 16 mpg (highway)\n\nExtras:\n\n + All-
wheel drive\n + Adaptive Suspension\n + Digital cluster\n + 8.4-inch Touchscreen Infotainment system\n
+ Apple CarPlay\n + Collision and Lane-Departure Warning\n + Adaptive Cruise Control";
}
private void camaro()
{
    pictureBoxCars.Image = Properties.Resources.Camaro;
    lblFeatures.Text = "Engine: Turbocharged 3.6-liter V6\n\nHorsepower: 550 hp\n\nTransmission: 6-
speed manual\n\n0-60 mph time: 5.3 seconds\n\nFuel economy: 22 mpg (city), 31 mpg
(highway)\n\nExtras:\n\n + Rear-Wheel Drive\n + Magnetic Ride Control Suspension\n + Touchscreen
Infotainment System\n + Backup Camera\n + Forward-Collision Warning\n + Rear Cross-Traffic Alert";
}

```

```

    }
    private void supra()
    {
        pictureBoxCars.Image = Properties.Resources.Supra;
        lblFeatures.Text = "Engine: 3.0-liter turbocharged inline-six\n\nHorsepower: 335 hp\n\nTransmission:
8-speed automatic\n\n0-60 mph time: 4.1 seconds\n\nFuel economy: 22 mpg (city), 30 mpg
(highway)\n\nExtras:\n\n + Rear-wheel drive\n + Adaptive Suspension\n + 8.8-inch Touchscreen
Tnfotainment system\n + Apple CarPlay/Android Auto\n + Collision warning and Lane-departure Warning\n
+ Adaptive Cruise Control.";
    }
    private void btnBack_Click(object sender, EventArgs e)
    {
        this.Close();
        Form1 form = new Form1();
        form.Show();
    }

    private void btnClose_Click(object sender, EventArgs e)
    {
        Application.Exit();
    }

    private void btnBook_Click(object sender, EventArgs e)
    {
        this.Close();
        FormBooking form3 = new FormBooking(selectedCar);
        form3.Show();
    }

    private void btnNextImage_Click(object sender, EventArgs e)
    {
        switch (selectedCar)
        {
            case "Challenger2022":
                pictureBoxCars.Image = Properties.Resources.Challenger2;
                break;
            case "Supra2015":
                pictureBoxCars.Image = Properties.Resources.Supra2;
                break;
            case "Mustang2019":
                pictureBoxCars.Image = Properties.Resources.Mustang2;
                break;
            case "Camaro2023":
                pictureBoxCars.Image = Properties.Resources.Camaro2;
                break;
            case "Aventador2020":
                pictureBoxCars.Image = Properties.Resources.Aventador2;
                break;
            case "GWagon2017":
                pictureBoxCars.Image = Properties.Resources.GWagon2;
                break;
            default:
                break;
        }
    }

    private void btnPreviousImage_Click(object sender, EventArgs e)

```



```
{
    switch (selectedCar)
    {
        case "Challenger2022":
            pictureBoxCars.Image = Properties.Resources.Challenger;
            break;
        case "Supra2015":
            pictureBoxCars.Image = Properties.Resources.Supra;
            break;
        case "Mustang2019":
            pictureBoxCars.Image = Properties.Resources.Mustang;
            break;
        case "Camaro2023":
            pictureBoxCars.Image = Properties.Resources.Camaro;
            break;
        case "Aventador2020":
            pictureBoxCars.Image = Properties.Resources.Aventador;
            break;
        case "GWagon2017":
            pictureBoxCars.Image = Properties.Resources.GWagon;
            break;
        default:
            break;
    }
}
}
```

## Code: FormBooking

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.OleDb;

namespace CarRentalProject
{
    public partial class FormBooking : Form
    {
        bool submitted = false;
        private string selectedCar;
        private const string connectionString = "Provider=Microsoft.ACE.OLEDB.12.0;" + "Data
Source=ProjectTables.accdb";

        DataTable table = new DataTable();
        public void dbfun(string str)
        {
            string constr = "Provider=Microsoft.ACE.OLEDB.12.0;" + "Data Source=ProjectTables.accdb";
            string sqlstr = str;
            OleDbDataAdapter da = new OleDbDataAdapter(sqlstr, constr);
            da.Fill(table);
            da.Dispose();
        }
        public FormBooking(string selectedCar)
        {
            InitializeComponent();
            this.selectedCar = selectedCar;
        }
        private void btnClose_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }
        private void btnSubmit_Click(object sender, EventArgs e)
        {
            string firstName = txtFirstName.Text;
            string lastName = txtLastName.Text;
            string telephoneNumber = txtPhnNmbr.Text;
            string cardNumber = txtCardNmbr.Text;
            DateTime pickupDate = datePickFrom.Value;
            DateTime dropOffDate = datePickTo.Value;
            long phoneNum;
            long cardNum;
            bool allConditionsMet = true;

            if (string.IsNullOrEmpty(firstName) || string.IsNullOrEmpty(lastName) ||
string.IsNullOrEmpty(telephoneNumber) || string.IsNullOrEmpty(cardNumber))
            {

```

```

    MessageBox.Show("Please fill out all fields.");
    allConditionsMet = false;
}

if (!long.TryParse(txtPhnNmbr.Text, out phoneNum) )
{
    MessageBox.Show("Please enter valid phone number.");
    allConditionsMet = false;
}

if (!long.TryParse(txtCardNmbr.Text, out cardNum))
{
    MessageBox.Show("Please enter valid card number.");
    allConditionsMet = false;
}

if (telephoneNumber.Length != 10)
{
    MessageBox.Show("Please enter a 10-digit phone number.");
    allConditionsMet = false;
}

if (cardNumber.Length != 16)
{
    MessageBox.Show("Please enter a 16-digit card number.");
    allConditionsMet = false;
}

if (pickupDate < DateTime.Now.Date)
{
    MessageBox.Show("Invalid pickup date. Please select a date on or after today's date.");
    allConditionsMet = false;
}

if (allConditionsMet)
{
    try
    {
        using (OleDbConnection connection = new OleDbConnection(connectionString))
        {
            connection.Open();
            int carID = GetCarID(connection, selectedCar);

            OleDbCommand command = new OleDbCommand("INSERT INTO Bookings (CarID,
CustomerID, PickupDate, DropOffDate, TotalCost) VALUES (@carID, @customerID, @pickupDate,
@dropOffDate, @totalCost)", connection);
            command.Parameters.AddWithValue("@carID", carID);
            command.Parameters.AddWithValue("@customerID", GetLatestCustomerID(connection));
            command.Parameters.AddWithValue("@pickupDate", pickupDate.Date);
            command.Parameters.AddWithValue("@dropOffDate", dropOffDate.Date);
            command.Parameters.AddWithValue("@totalCost", CalculateTotalCost(carID, pickupDate,
dropOffDate));
            int rowsAffected = command.ExecuteNonQuery();
            if (rowsAffected > 0)
            {
                MessageBox.Show("Booking data has been saved successfully.");
            }
        }
    }
    catch { }
}

```

```

    }

    OleDbCommand command2 = new OleDbCommand("UPDATE CarsData SET StockQuantity =
@stockQuantity WHERE CarID = @carID", connection);
    command2.Parameters.AddWithValue("@stockQuantity", GetStockQuantity(connection, carID)
- 1);

    command2.Parameters.AddWithValue("@carID", carID);
    command2.ExecuteNonQuery();

    OleDbCommand command3 = new OleDbCommand("INSERT INTO CustomerInfo (FirstName,
LastName, TelephoneNumber, CardNumber) VALUES (@firstName, @lastName, @telephoneNumber,
@cardNumber)", connection);
    command3.Parameters.AddWithValue("@firstName", firstName);
    command3.Parameters.AddWithValue("@lastName", lastName);
    command3.Parameters.AddWithValue("@telephoneNumber", telephoneNumber);
    command3.Parameters.AddWithValue("@cardNumber", cardNumber);
    int rowsAffected2 = command3.ExecuteNonQuery();
    if (rowsAffected2 > 0)
    {
        MessageBox.Show("Customer data has been saved successfully.");
        submitted = true;
    }
    btnSubmit.Enabled = false;
}
}
catch (Exception ex)
{
    MessageBox.Show("An error occurred while saving customer data: " + ex.Message);
}
}

private int GetCarID(OleDbConnection connection, string carModel)
{
    OleDbCommand command4 = new OleDbCommand("SELECT CarID FROM CarsData WHERE CarModel
= @selectedCar", connection);
    command4.Parameters.AddWithValue("@selectedCar", carModel);
    object result = command4.ExecuteScalar();
    if (result != DBNull.Value && result != null)
    {
        return Convert.ToInt32(result);
    }
    else
    {
        throw new Exception("Invalid car model.");
    }
}

private int GetLatestCustomerID(OleDbConnection connection)
{
    OleDbCommand command5 = new OleDbCommand("SELECT MAX(CustomerID) FROM CustomerInfo",
connection);
    object result = command5.ExecuteScalar();
    if (result != DBNull.Value && result != null)
    {
        return Convert.ToInt32(result) + 1;
    }
    else

```

```

    {
        return 1;
    }
}
private int GetStockQuantity(OleDbConnection connection, int carID)
{
    OleDbCommand command6 = new OleDbCommand("SELECT StockQuantity FROM CarsData WHERE
CarID = @carID", connection);
    command6.Parameters.AddWithValue("@carID", carID);
    object result = command6.ExecuteScalar();
    if (result != DBNull.Value && result != null)
    {
        return Convert.ToInt32(result);
    }
    else
    {
        return 0;
    }
}
private decimal CalculateTotalCost(int carID, DateTime pickupDate, DateTime dropOffDate)
{
    decimal costPerDay;
    using (OleDbConnection connection = new OleDbConnection(connectionString))
    {
        connection.Open();
        OleDbCommand command7 = new OleDbCommand("SELECT CostPerDay FROM CarsData WHERE
CarID = @carID", connection);
        command7.Parameters.AddWithValue("@carID", carID);
        object result = command7.ExecuteScalar();
        if (result != DBNull.Value && result != null)
        {
            costPerDay = Convert.ToDecimal(result);
        }
        else
        {
            throw new Exception("Invalid car ID.");
        }
        int numDays = (int)(dropOffDate - pickupDate).TotalDays;
        if (numDays <= 0)
        {
            throw new Exception("Invalid pickup and drop off dates.");
        }
        Decimal total = costPerDay * numDays;
        return total * Convert.ToDecimal(1.13);
    }
}
private void btnViewBooking_Click(object sender, EventArgs e)
{
    if(submitted == true)
    {
        this.dbfun("SELECT * FROM Bookings ORDER BY BookingID DESC");
        DataRow dr = table.Rows[0];
        string BookingID = dr["BookingID"].ToString();
        this.Close();
        FormReceipt form4 = new FormReceipt(BookingID);
        form4.Show();
    }
}

```

}

}

}



## Code: FormReceipt

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.OleDb;

namespace CarRentalProject
{
    public partial class FormReceipt : Form
    {
        DataTable BookingTable = new DataTable();
        DataTable CarTable = new DataTable();
        DataTable CustomerTable = new DataTable();
        private string BookingID, customerID, carID, firstName, lastName, telephone, carType, model, pickup,
        dropoff, totalcost, cardnumber;
        public FormReceipt(string BookingID)
        {
            InitializeComponent();
            this.BookingID = BookingID;
        }
        private void FormReceipt_Load(object sender, EventArgs e)
        {
            this.dbfun($"SELECT * FROM Bookings WHERE BookingID = {BookingID}", BookingTable);
            DataRow bookingRow = BookingTable.Rows[0];
            this.customerID = bookingRow["CustomerID"].ToString();
            this.carID = bookingRow["CarID"].ToString();
            this.pickup = bookingRow["PickupDate"].ToString();
            this.dropoff = bookingRow["DropOffDate"].ToString();
            this.totalcost = bookingRow["TotalCost"].ToString();
            this.dbfun($"SELECT * FROM CarsData WHERE CarID = {this.carID}", CarTable);
            DataRow carRow = CarTable.Rows[0];
            this.carType = carRow["CarType"].ToString();
            this.model = carRow["CarModel"].ToString();
            this.dbfun($"SELECT * FROM CustomerInfo WHERE CustomerID = {this.customerID}", CustomerTable);
            DataRow customerRow = CustomerTable.Rows[0];
            this.firstName = customerRow["FirstName"].ToString();
            this.lastName = customerRow["LastName"].ToString();
            this.telephone = customerRow["TelephoneNumber"].ToString();
            this.cardnumber = customerRow["CardNumber"].ToString();

            lblPickupDate.Text = pickup;
            lblDropOffDate.Text = dropoff;
            lblBookingID.Text = BookingID;
            lblCarID.Text = carID;
            lblCarModel.Text = model;
            lblCarType.Text = carType;
            lblFName.Text = firstName;
            lblLName.Text = lastName;
        }
    }
}

```

```

lblPhNumber.Text = telephone;
lblCost.Text = "$" + totalcost;

if (cardnumber.Length >= 4)
{
    string lastFourChars = cardnumber.Substring(cardnumber.Length - 4);
    string maskedText = new string('X', cardnumber.Length - 4) + lastFourChars;
    lblCardNumber.Text = maskedText;
}
else
{
    string maskedText = new string('X', cardnumber.Length);
    lblCardNumber.Text = maskedText;
}
}
public void dbfun(string str, DataTable table)
{
    string constr = "Provider=Microsoft.ACE.OLEDB.12.0;" + "Data Source=ProjectTables.accdb";
    string sqlstr = str;
    OleDbDataAdapter da = new OleDbDataAdapter(sqlstr, constr);
    da.Fill(table);
    da.Dispose();
}
private void btnClose_Click(object sender, EventArgs e)
{
    Application.Exit();
}
}
}

```