CAR SERVICE MANAGEMENT SYSTEM

A MINI-PROJECT REPORT

Submitted by

1. SAICHUPAK E 240701456
2. KISHORE KUMAR 240701268
3. MOHAMMED ABDUL KADAR 240701329

in partial fulfilment of the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

An Autonomous Institute

CHENNAI

NOVEMBER 2025

BONAFIDE CERTIFICATE

Certified that this project **“CAR SERVICE MANAGEMENT SYSTEM”** is the Bonafide work of **“SAICHUPAK E”,”KISHORE KUMAR J”,”MOHAMMED ABDUL KADAR”** who carried out the project work under my supervision.

SIGNATURE

Mr. B.DEEPA

ASSISTANT PROFESSOR

Dept. of Computer Science and Engg,

Rajalakshmi Engineering College

Chennai

This mini project report is submitted for the viva voce examination to be held on

____

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ABSTRACT

The **Car Service Booking Management System** is a Java-based desktop application designed to streamline and automate the process of car service scheduling and management. The system enables customers to conveniently
book vehicle servicing appointments, while allowing administrators to manage bookings, update service details, and maintain customer records efficiently.

ACKNOWLEDGEMENT

We express our sincere thanks to our beloved and honourable chairman **MR. S. MEGANATHAN** and the chairperson **DR. M.THANGAM MEGANATHAN** for their timely support and encouragement.

We are greatly indebted to our respected and honourable principal

Dr. S.N. MURUGESAN for his able support and guidance.

No words of gratitude will suffice for the unquestioning support extended to us by our Head Of The Department **Dr. E.M. MALATHY** and our Deputy Head Of The Department **Dr. J. MANORANJINI** for being ever supporting force during our project work

We also extend our sincere and hearty thanks to our internal guide

Mrs. B. DEEPA, for her valuable guidance and motivation during the completion of this project.

Our sincere thanks to our family members, friends and other staff members of computer science engineering.

1. **SAICHUPAK E**
2. **KISHORE KUMAR J**
3. **MOHAMED ABDUL KADAR**

TABLE OF CONTENTS

LIST of figures

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The **Car Service Booking Management System** is designed to simplify and digitize the process of booking and managing car servicing appointments. This project enables customers to easily schedule maintenance or repair services for their vehicles and allows service centres to manage bookings efficiently. The system provides all necessary details, such as customer information, service type, car model, and service cost, in a structured and user-friendly manner. It helps both customers and administrators by saving time, reducing manual errors, and improving overall service efficiency.

1.2 SCOPE OF THE WORK

The scope of this project is to develop an effective and reliable system that allows customers to book car service appointments conveniently and enables administrators to manage these bookings efficiently. The system provides functionalities such as adding new bookings, updating existing details, deleting completed or cancelled services, and viewing all service records. It is designed to be used by automobile service centres to improve their workflow and by customers to easily access service information anytime. The project can be extended in the future to include features like online payment, SMS/email notifications, and service reminders.

1.3 PROBLEM STATEMENT

In many service centres, booking and managing car services are still performed manually, which often leads to errors, data mismanagement, and customer dissatisfaction. Customers face inconvenience in booking appointments due to limited accessibility and poor record management. There is also a lack of a centralized system to monitor service details, track vehicle history, and manage customer information efficiently. Therefore, a computerized **Car Service Booking Management System** is required to overcome these issues and ensure a smooth and organized service management process

1.4 AIM AND OBJECTIVES OF THE PROJECT

The main aim of the **Car Service Booking Management System** is to automate and streamline the car service booking process. The system ensures easy management of customer and service details, thereby enhancing service quality and customer satisfaction.

CHAPTER 2

SYSTEM SPECIFICATIONS

2.1 HARDWARE SPECIFICATIONS

Processor	:	Intel i5
Memory Size	:	8GB (Minimum)
HDD	:	1 TB (Minimum)

2.2 SOFTWARE SPECIFICATIONS

Operating System	**:**	**WINDOWS 11**
Front - End	**:**	**JAVA SWING**
Back - End	**:**	**MYSQL**
Language	**:**	**JAVA,SQL**

CHAPTER

3 MODULE DESCRIPTION

This application consists of two modules. When the program runs, it will ask for a confirmation to the login window. The person who interacts can login as an Administrator or as a User. The description of the modules are as follows:

1. **Admin login**

When the person who interacts tries to login as Admin then he needs to login with his username and password. The administrator only has the power to change and manipulate the data in the database.

2. **User login**

When the person tries to login as a user then he/she will be prompted to enter the number of symptoms and the final result will be printed in the form of table.

CHAPTER 4 SAMPLE

CODING

```
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Component;
import java.awt.GridLayout;
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.BorderFactory;
import javax.swing.JButton;
import
javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import
javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.JTextField;
import javax.swing.SwingUtilities;
import javax.swing.table.DefaultTableModel;
```

```
public class CarServiceBookingSystem extends JFrame {
```

```
static final String DB_URL =
"jdbc:mysql://localhost:3306/"; static final String
DB_NAME = "car_service_db";
static final String DB_USER = "root";
static final String DB_PASS = "s123";
public JTextField tfCustomerName;
public JTextField tfCarModel;
public JTextField
tfServiceCost; public
JTextField tfPhone;
public JTextField tfAddress;
public JTextField tfDOB;
public JTextField tfSearch;
public JComboBox<String> cbFuelType;
public JComboBox<String> cbServiceType;
public JComboBox<String> cbPaymentMethod;
public JTable table;
public DefaultTableModel
tableModel; public String
loggedInUser;

public CarServiceBookingSystem(String var1) {
  this.loggedInUser = var1;
  this.initializeDatabase();
  this.setTitle("Car Service Booking System - " + var1);
  this.setSize(1000, 700);
  this.setDefaultCloseOperation(3);
  this.setLocationRelativeTo((Component)null);
  this.setLayout(new BorderLayout());
  JPanel var2 = new JPanel(new BorderLayout());
```

```
var2.setBackground(new Color(41, 128, 185));
```

```
  JLabel var3 = new JLabel("Logged in as: " + var1);
  var3.setForeground(Color.WHITE);
  var2.add(var3, "West");
  JButton var4 = new
  JButton("Logout");
  var4.setBackground(Color.RED);
  var4.setForeground(Color.WHITE);
  var4.addActionListener((var1x) -> {
   this.logout();
  });
  var2.add(var4, "East");
  this.add(var2, "North");
  JPanel var5 = new JPanel(new GridLayout(10, 2, 5, 5));
  var5.setBorder(BorderFactory.createTitledBorder("Booking
  Details")); var5.add(new JLabel("Customer Name:"));
  this.tfCustomerName = new JTextField();
  var5.add(this.tfCustomerName);
  var5.add(new JLabel("Car Model:"));
  this.tfCarModel = new JTextField();
  var5.add(this.tfCarModel);
  var5.add(new JLabel("Service Type:"));
  this.cbServiceType = new JComboBox(new String[]{"Full Service", "Oil Change",
"Engine Check", "Tire Change"});
  var5.add(this.cbServiceType);
  var5.add(new JLabel("Fuel
  Type:"));
  this.cbFuelType = new JComboBox(new String[]{"Petrol", "Diesel", "Electric",
"Hybrid"});
  var5.add(this.cbFuelType);
  var5.add(new JLabel("Service Cost:"));
  this.tfServiceCost = new JTextField();
```

```
    var5.add(this.tfServiceCost);

    var5.add(new JLabel("Phone Number:"));

    this.tfPhone = new JTextField();

    var5.add(this.tfPhone);

    var5.add(new

    JLabel("Address:"));

    this.tfAddress = new

    JTextField();

    var5.add(this.tfAddress);

    var5.add(new JLabel("Date of Birth:"));

    this.tfDOB = new JTextField();

    var5.add(this.tfDOB);

    var5.add(new JLabel("Payment Method:"));

    this.cbPaymentMethod = new JComboBox(new String[]{"Cash", "Credit Card",
"UPI", "Net Banking"});

    var5.add(this.cbPaymentMethod);

    JButton var6 = new JButton("Add Booking");

    var6.setBackground(new Color(46, 204, 113));

    var6.setForeground(Color.WHITE);

    var6.addActionListener((var1x) ->

      { this.addBooking();

    });

    var5.add(var6);

    this.add(var5, "West");

    this.tableModel = new DefaultTableModel();

    this.tableModel.setColumnIdentifiers(new String[]{"ID", "Name", "Car Model",
"Service Type", "Fuel Type", "Cost", "Phone", "Address", "DOB", "Payment", "Status"});

    this.table = new JTable(this.tableModel);

    this.add(new JScrollPane(this.table),

    "Center"); JPanel var7 = new JPanel();

    this.tfSearch = new JTextField(15);
```

```
  JButton var8 = new JButton("Search");
  JButton var9 = new JButton("Show
  All"); JButton var10 = new
  JButton("Delete"); var7.add(new
  JLabel("Search:"));
  var7.add(this.tfSearch);
  var7.add(var8);
  var7.add(var9);
  var7.add(var10);
  this.add(var7,
  "South");
  var8.addActionListener((var1x) ->
    { this.searchBookings();
  });
  var9.addActionListener((var1x) ->
    { this.loadBookings();
  });
  var10.addActionListener((var1x)
    -> { this.deleteBooking();
  });
  this.loadBookings();
}

public void
  logout() {
  this.dispose();
  SwingUtilities.invokeLater(() -> {
    (new
    LoginPage()).setVisible(true);
  });
```

```
}
```

```
  public Connection getConnection(boolean var1) throws SQLException {
    String var2 = var1 ? "Jdbc:mysql://localhost:3306/car_service_db" :
"jdbc:mysql://localhost:3306/";
    return DriverManager.getConnection(var2, "root", "s123");
  }

  public void
    initializeDatabase() {
    Connection var1;
    Statement var2;
    try {
      var1 = this.getConnection(false);

      try {
        var2 = var1.createStatement();

        try {
          var2.executeUpdate("CREATE DATABASE IF NOT EXISTS car_service_db");
          System.out.println("Database checked/created successfully!");
        } catch (Throwable var12)
          { if (var2 != null) {
            try {
              var2.close();
            } catch (Throwable var8) {
              var12.addSuppressed(var8);
            }
          }

          throw var12;
        }

        if (var2 !=
          null) {
          var2.close();
        }
      } catch (Throwable
        var13) { if (var1 != null)
        {
          try {
            var1.close();
          } catch (Throwable var7) {
            var13.addSuppressed(var7);
          }
        }

        throw var13;
      }

      if (var1 != null)
        { var1.close();
      }
    } catch (SQLException var14) {
      JOptionPane.showMessageDialog(this, "Database Creation Error: " +
var14.getMessage());
    }

    try {
      var1 = this.getConnection(true);
      try {
        var2 = var1.createStatement();


        try {
          String var3 = " CREATE TABLE IF NOT EXISTS bookings (\n    customer_id
INT AUTO_INCREMENT PRIMARY KEY,\n
customer_name VARCHAR(100) NOT
NULL,\n     car_model VARCHAR(100) NOT NULL,\n     service_type VARCHAR(100)
NOT NULL,\n     fuel_type VARCHAR(50) NOT NULL,\n    service_cost DOUBLE
NOT NULL,\n     phone_number VARCHAR(20) NOT NULL,\n          address
VARCHAR(200) NOT NULL,\n    date_of_birth VARCHAR(20) NOT NULL,\n
payment_method VARCHAR(50) NOT NULL,\n             deleted BOOLEAN DEFAULT
FALSE\n )\n";

          var2.executeUpdate(var3);

          System.out.println("Table checked/created successfully!");

          ResultSet var4 = var2.executeQuery("SELECT COUNT(*) FROM bookings");

          var4.next();

          if (var4.getInt(1) == 0) {

            var2.executeUpdate(" INSERT INTO bookings (customer_name,
car_model, service_type, fuel_type, service_cost, phone_number, address,
date_of_birth,
payment_method)\n VALUES ('Arun Kumar', 'Honda City', 'Full Service', 'Petrol',
4500, '9876543210', 'Chennai', '15-08-1998', 'UPI'),\n ('Deepa R', 'Hyundai i20', 'Oil
Change', 'Diesel', 1800, '9998844221', 'Coimbatore', '05-06-1995', 'Cash')\n");

            System.out.println("Sample data inserted!");

          }

        } catch (Throwable

          var9) { if (var2 != null)

          {

            try {

              var2.close();

            } catch (Throwable var6) {

              var9.addSuppressed(var6);

            }
}
```

```
          throw var9;
        }

        if (var2 !=
          null) {
          var2.close();
        }
      } catch (Throwable
        var10) { if (var1 != null)
        {
          try {
            var1.close();
          } catch (Throwable var5) {
            var10.addSuppressed(var5);
          }
        }

        throw var10;
      }

      if (var1 != null)
        { var1.close();
      }
    } catch (SQLException var11) {
      JOptionPane.showMessageDialog(this, "Table Creation Error: " +
var11.getMessage());
    }

  }
```

```
 public void addBooking()
  { try {
    Connection var1 = this.getConnection(true);


    try {
      String var2 = "INSERT INTO bookings (customer_name, car_model,
service_type, fuel_type, service_cost, phone_number, address, date_of_birth,
payment_method) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)";
      PreparedStatement var3 =
      var1.prepareStatement(var2); var3.setString(1,
      this.tfCustomerName.getText()); var3.setString(2,
      this.tfCarModel.getText());
      var3.setString(3, this.cbServiceType.getSelectedItem().toString());
      var3.setString(4, this.cbFuelType.getSelectedItem().toString());
      var3.setDouble(5, Double.parseDouble(this.tfServiceCost.getText()));
      var3.setString(6, this.tfPhone.getText());
      var3.setString(7, this.tfAddress.getText());
      var3.setString(8, this.tfDOB.getText());
      var3.setString(9, this.cbPaymentMethod.getSelectedItem().toString());
      var3.executeUpdate();
      JOptionPane.showMessageDialog(this, "Booking Added Successfully!");
      this.loadBookings();
      this.clearFields();
    } catch (Throwable var5)
     { if (var1 != null) {
       try {
         var1.close();
       } catch (Throwable var4) {
         var5.addSuppressed(var4);
```

```
          }
        }

        throw var5;
      }

      if (var1 != null)
        { var1.close();
      }
    } catch (Exception var6) {
      JOptionPane.showMessageDialog(this, "Error adding booking: " +
var6.getMessage());
    }

  }

  public void loadBookings() {
    try {
      Connection var1 = this.getConnection(true);

      try {
        Statement var2 = var1.createStatement();

        try {
          ResultSet var3 = var2.executeQuery("SELECT * FROM bookings");

          try {
            this.tableModel.setRowCount(0);
            while(var3.next()) {
              this.tableModel.addRow(new Object[]{var3.getInt("customer_id"),
var3.getString("customer_name"), var3.getString("car_model"),
var3.getString("service_type"), var3.getString("fuel_type"),
var3.getDouble("service_cost"), var3.getString("phone_number"),
var3.getString("address"), var3.getString("date_of_birth"),
var3.getString("payment_method"), var3.getBoolean("deleted") ? "Deleted" :
"Active"});
            }
          } catch (Throwable
            var9) { if (var3 != null)
            {
              try {
                var3.close();
              } catch (Throwable var8) {
                var9.addSuppressed(var8);
              }
            }

            throw var9;
          }

          if (var3 != null) {
            var3.close();
          }
        } catch (Throwable var10)
          { if (var2 != null) {
            try {
              var2.close();
            } catch (Throwable var7) {
```

```
          var10.addSuppressed(var7);
        }
      }

      throw var10;
    }

    if (var2 !=
      null) {
      var2.close();
    }
  } catch (Throwable
    var11) { if (var1 != null)
    {
      try {
        var1.close();
      } catch (Throwable var6) {
        var11.addSuppressed(var6);
      }
    }

    throw var11;
  }

  if (var1 != null)
    { var1.close();
  }
} catch (Exception var12) {
  JOptionPane.showMessageDialog(this, "Error loading data: " +
  var12.getMessage());
}
```

```
  }


  public void searchBookings() {
    String var1 = this.tfSearch.getText().trim().toLowerCase();


    try {
      Connection var2 = this.getConnection(true);


      try {
        String var3 = "SELECT * FROM bookings WHERE LOWER(customer_name) LIKE ?
OR customer_id LIKE ?";
        PreparedStatement var4 =
        var2.prepareStatement(var3); var4.setString(1, "%" +
        var1 + "%");
        var4.setString(2, "%" + var1 + "%");
        ResultSet var5 = var4.executeQuery();
        this.tableModel.setRowCount(0);


        while(var5.next()) {
          this.tableModel.addRow(new Object[]{var5.getInt("customer_id"),
var5.getString("customer_name"), var5.getString("car_model"),
var5.getString("service_type"), var5.getString("fuel_type"),
var5.getDouble("service_cost"), var5.getString("phone_number"),
var5.getString("address"), var5.getString("date_of_birth"),
var5.getString("payment_method"), var5.getBoolean("deleted") ? "Deleted" :
"Active"});
        }
      } catch (Throwable var7)
        { if (var2 != null) {
          try {
            var2.close();
        } catch (Throwable var6) {
          var7.addSuppressed(var6);
        }
      }

      throw var7;
    }

    if (var2 != null)
      { var2.close();
    }
  } catch (Exception var8) {
    JOptionPane.showMessageDialog(this, "Search Error: " + var8.getMessage());
  }

}

public void deleteBooking() {
  int var1 =
  this.table.getSelectedRow(); if (var1
  == -1) {
    JOptionPane.showMessageDialog(this, "Select a booking to delete.");
  } else {
    int var2 = (Integer)this.tableModel.getValueAt(var1, 0);

    try {
      Connection var3 = this.getConnection(true);

      try {
        PreparedStatement var4 = var3.prepareStatement("UPDATE bookings SET
deleted = TRUE WHERE customer_id = ?");

        var4.setInt(1, var2);

        var4.executeUpdate();

        JOptionPane.showMessageDialog(this, "Booking Marked as Deleted!");

        this.loadBookings();

      } catch (Throwable

        var7) { if (var3 != null)

        {

          try {

            var3.close();

          } catch (Throwable var6) {

            var7.addSuppressed(var6);

          }

        }


        throw var7;

      }


      if (var3 !=

        null) {

        var3.close();

      }

    } catch (Exception var8) {

      JOptionPane.showMessageDialog(this, "Delete Error: " + var8.getMessage());

    }


  }

}
```

```
  public void clearFields() {
    this.tfCustomerName.setText("");
    this.tfCarModel.setText("");
    this.tfServiceCost.setText("");
    this.tfPhone.setText("");
    this.tfAddress.setText("
    ");
    this.tfDOB.setText("");
    this.cbFuelType.setSelectedIndex(0);
    this.cbServiceType.setSelectedIndex(
    0);
    this.cbPaymentMethod.setSelectedIndex(0);
  }

  public static void main(String[] var0) {
    SwingUtilities.invokeLater(() -> {
      (new LoginPage()).setVisible(true);
    });
  }
}
```

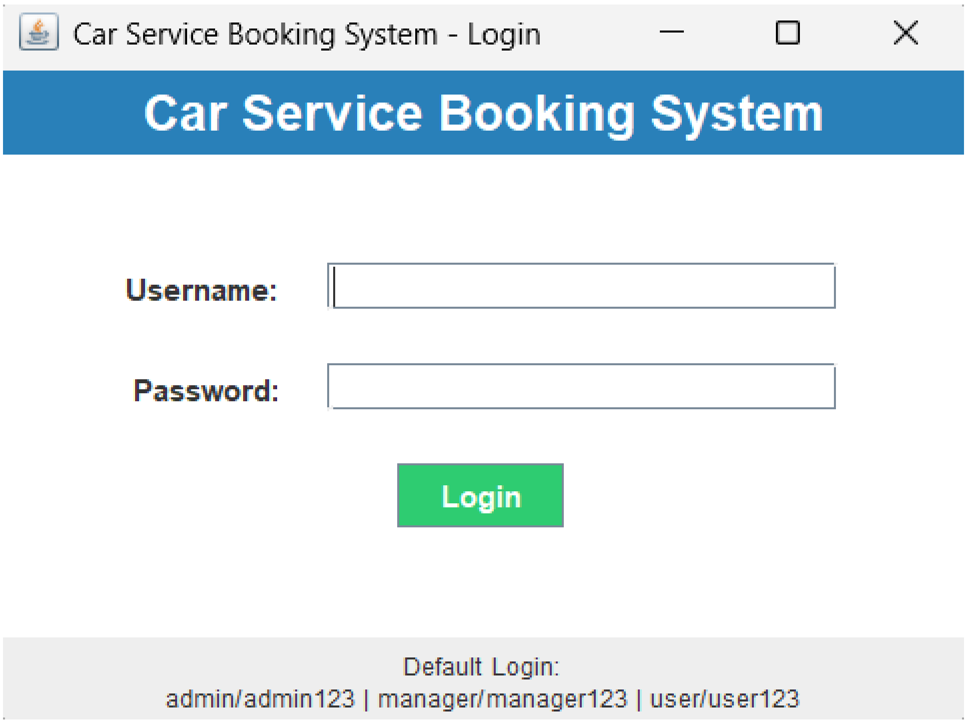
Sample 2

Sample 2 depicts the booking part of the code, where it displays booking details and enter user data and store it in database

CHAPTER 5

SCREEN SHOTS

Fig 5.1 Introduction page

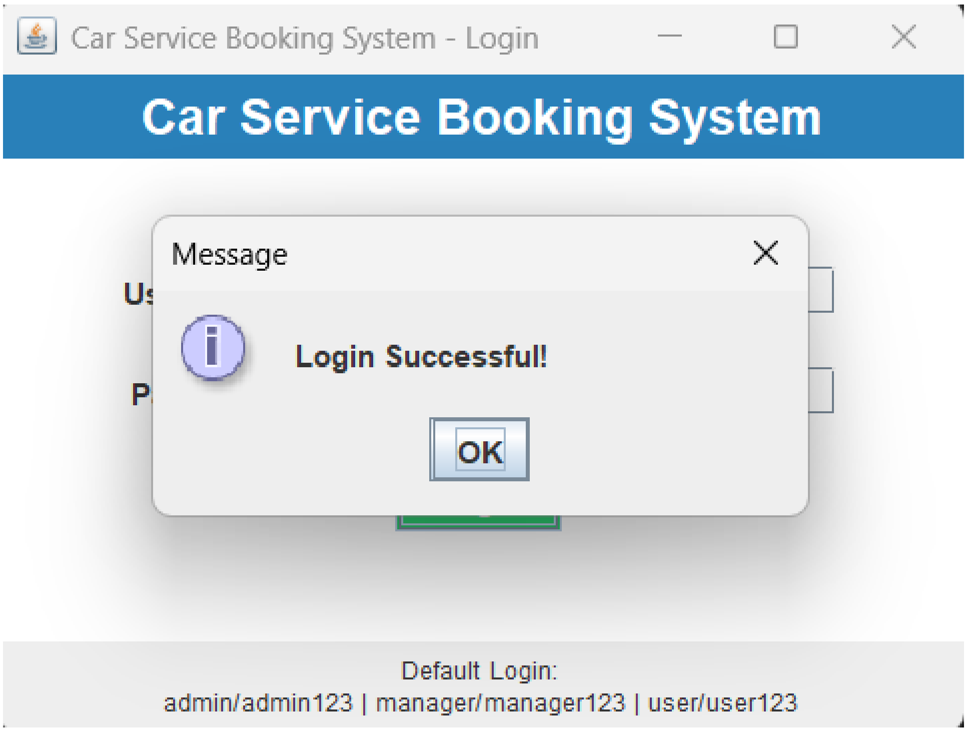


Car Service Booking System - Login

Car Service Booking System

Default Login:
admin/admin123 | manager/manager123 | user/user123

Fig 5.2 Customer details



Car Service Booking System - Login

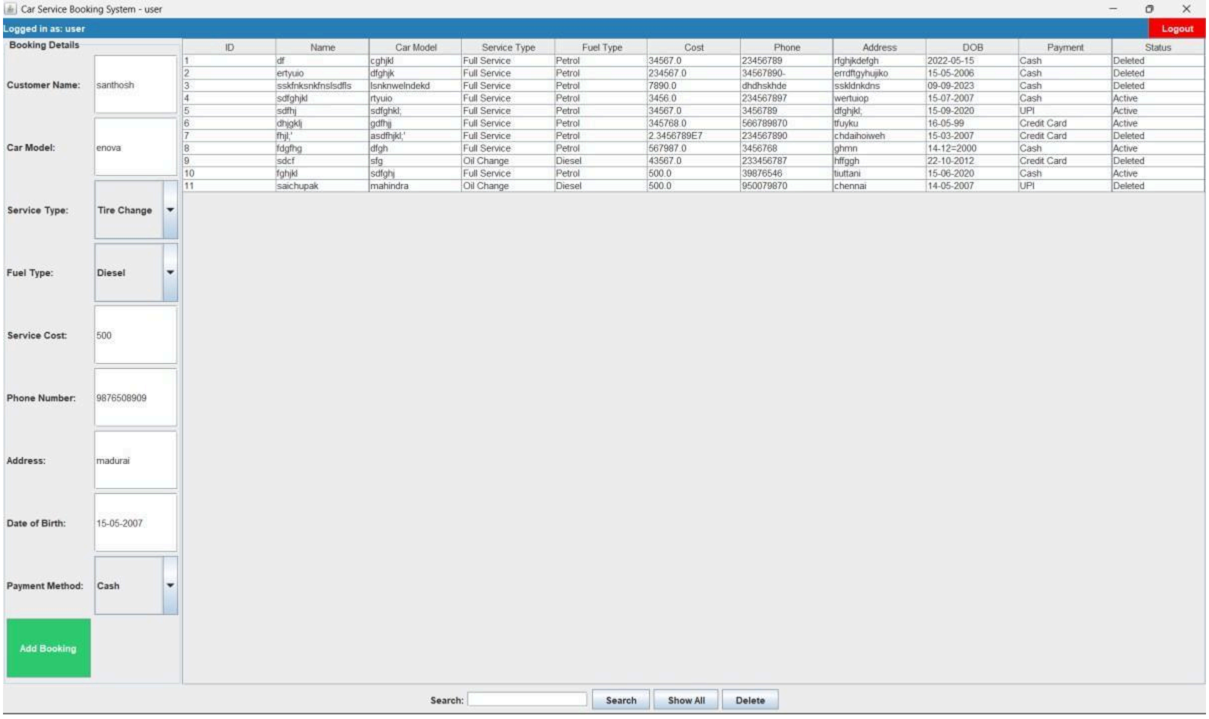
Car Service Booking System

Message

Login Successful!

Default Login:
admin/admin123 | manager/manager123 | user/user123

Fig 5.3 Booking Log



Booking Details

Customer Name: santhosh

Car Model: enova

Service Type: Tire Change

Fuel Type: Diesel

Service Cost: 500

Phone Number: 9876508909

Address: madurai

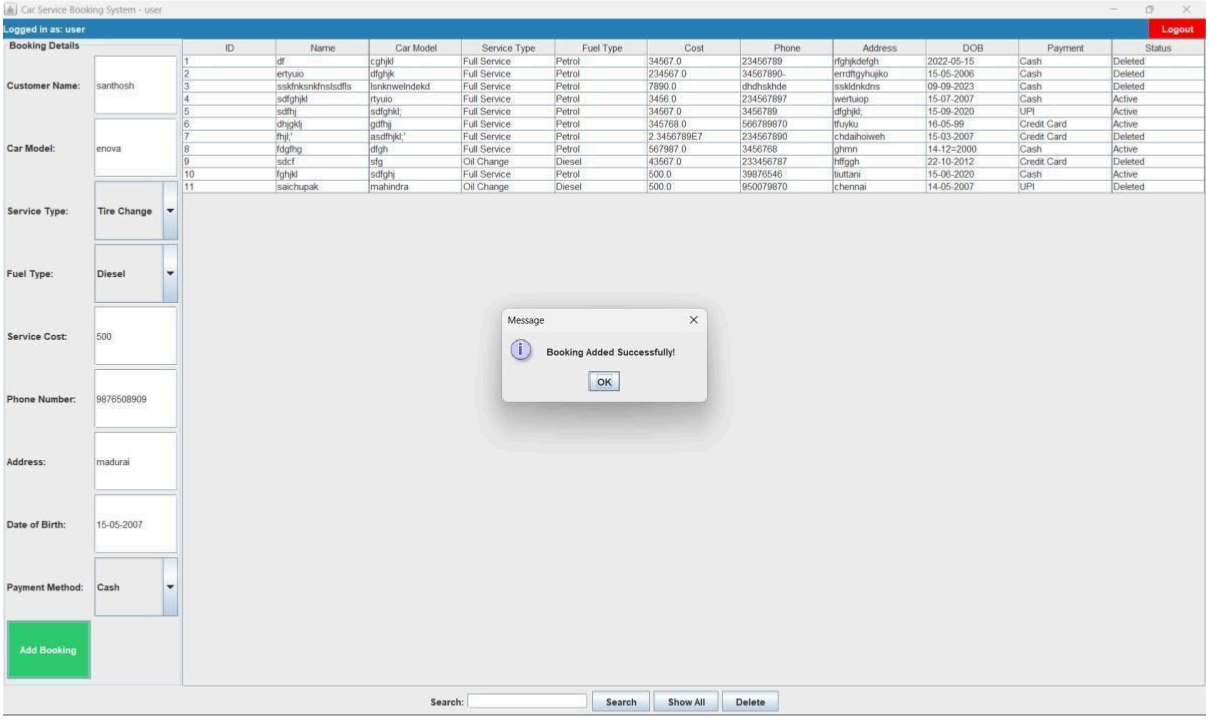
Date of Birth: 15-05-2007

Payment Method: Cash

ID	Name	Car Model	Service Type	Fuel Type	Cost	Phone	Address	DOB	Payment	Status
1	df	cghjkl	Full Service	Petrol	34567.0	23456789	rfghjkdefgh	2022-05-15	Cash	Deleted
2	ertyuio	dfghjk	Full Service	Petrol	234567.0	34567890-	errdftgyhujiko	15-05-2006	Cash	Deleted
3	sskfnksnkfnslsdfls	lsnknwelndekd	Full Service	Petrol	7890.0	dhdhskhde	sskldnkdns	09-09-2023	Cash	Deleted
4	sdfghjkl	rtyuio	Full Service	Petrol	3456.0	234567897	wertuiop	15-07-2007	Cash	Active
5	sdfhj	sdfghkl;	Full Service	Petrol	34567.0	3456789	dfghjkl;	15-09-2020	UPI	Active
6	dhjgklj	gdfhjj	Full Service	Petrol	345768.0	566789870	tfuyku	16-05-99	Credit Card	Active
7	fhjl;'	asdfhjkl;'	Full Service	Petrol	2.3456789E7	234567890	chdaihoiweh	15-03-2007	Credit Card	Deleted
8	fdgfhg	dfgh	Full Service	Petrol	567987.0	3456768	ghmn	14-12=2000	Cash	Active
9	sdcf	sfg	Oil Change	Diesel	43567.0	233456787	hffggh	22-10-2012	Credit Card	Deleted
10	fghjkl	sdfghj	Full Service	Petrol	500.0	39876546	tiuttani	15-06-2020	Cash	Active
11	saichupak	mahindra	Oil Change	Diesel	500.0	950079870	chennai	14-05-2007	UPI	Deleted

Search:

Fig 5.4 Booking creation



Booking Added Successfully!

Booking Details

Customer Name: santhosh

Car Model: enova

Service Type: Tire Change

Fuel Type: Diesel

Service Cost: 500

Phone Number: 9876508909

Address: madurai

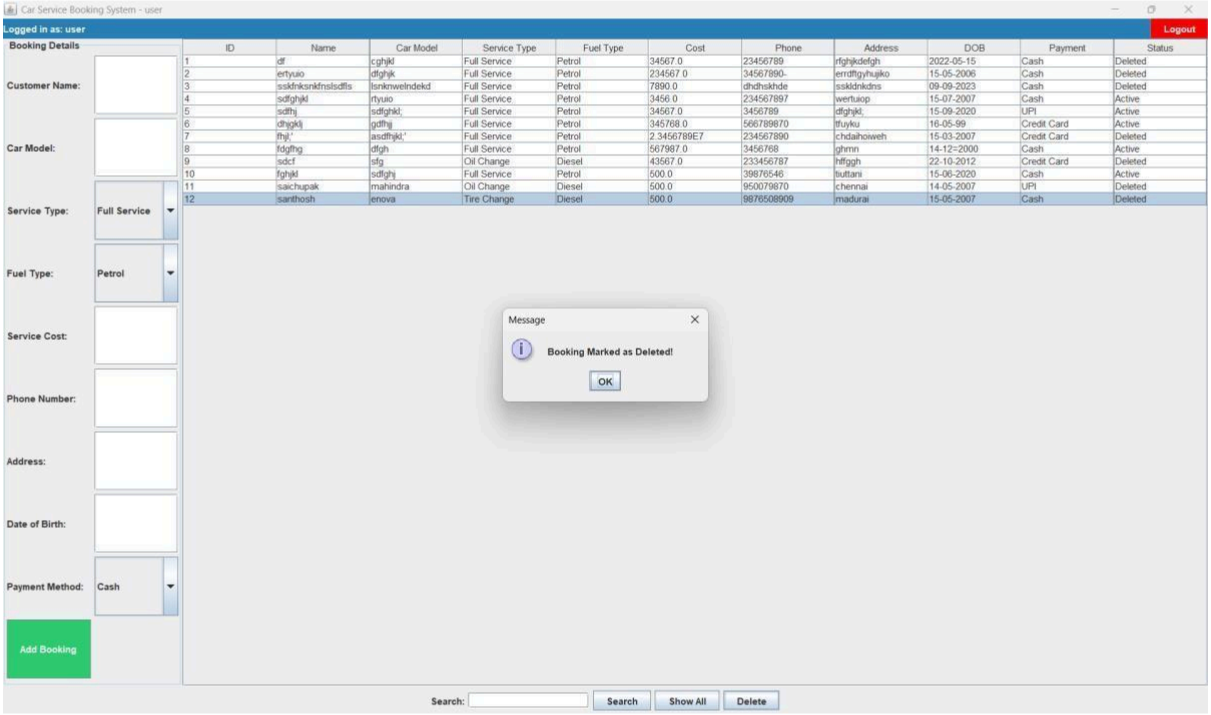
Date of Birth: 15-05-2007

Payment Method: Cash

ID	Name	Car Model	Service Type	Fuel Type	Cost	Phone	Address	DOB	Payment	Status
1	df	cghjkl	Full Service	Petrol	34567.0	23456789	rfghjkdefgh	2022-05-15	Cash	Deleted
2	ertyuio	dfghjk	Full Service	Petrol	234567.0	34567890-	errdftgyhujiko	15-05-2006	Cash	Deleted
3	sskfnksnkfnslsdfls	lsnknwelndekd	Full Service	Petrol	7890.0	dhdhskhde	sskldnkdns	09-09-2023	Cash	Deleted
4	sdfghjkl	rtyuio	Full Service	Petrol	3456.0	234567897	wertuiop	15-07-2007	Cash	Active
5	sdfhj	sdfghkl;	Full Service	Petrol	34567.0	3456789	dfghjkl;	15-09-2020	UPI	Active
6	dhjgklj	gdfhjj	Full Service	Petrol	345768.0	566789870	tfuyku	16-05-99	Credit Card	Active
7	fhjl;'	asdfhjkl;'	Full Service	Petrol	2.3456789E7	234567890	chdaihoiweh	15-03-2007	Credit Card	Deleted
8	fdgfhg	dfgh	Full Service	Petrol	567987.0	3456768	ghmn	14-12=2000	Cash	Active
9	sdcf	sfg	Oil Change	Diesel	43567.0	233456787	hffggh	22-10-2012	Credit Card	Deleted
10	fghjkl	sdfghj	Full Service	Petrol	500.0	39876546	tiuttani	15-06-2020	Cash	Active
11	saichupak	mahindra	Oil Change	Diesel	500.0	950079870	chennai	14-05-2007	UPI	Deleted

Search:

Fig 5.5 Deletion of Booking



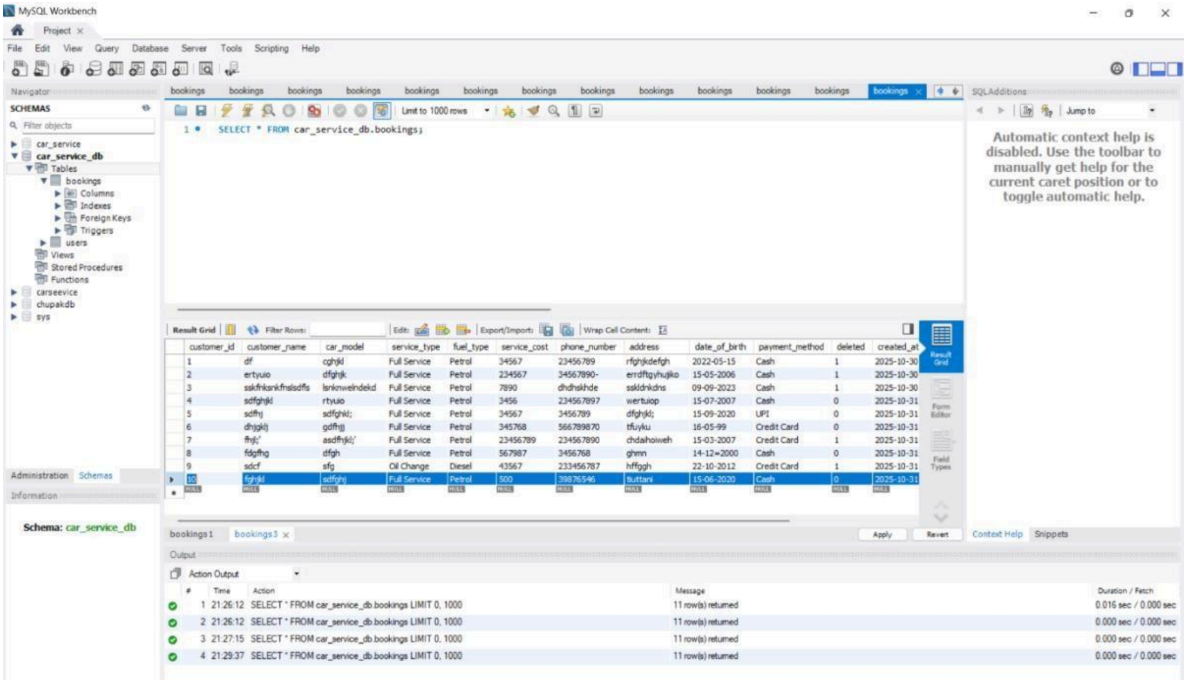
Customer Name:	
Car Model:	
Service Type:	Full Service
Fuel Type:	Petrol
Service Cost:	
Phone Number:	
Address:	
Date of Birth:	
Payment Method:	Cash

ID	Name	Car Model	Service Type	Fuel Type	Cost	Phone	Address	DOB	Payment	Status
1	df	cghjkl	Full Service	Petrol	34567.0	23456789	rfghjkdefgh	2022-05-15	Cash	Deleted
2	ertyuio	dfghjk	Full Service	Petrol	234567.0	34567890-	errdftgyhujiko	15-05-2006	Cash	Deleted
3	sskfnksnkfnslsdfls	lsnknwelndekd	Full Service	Petrol	7890.0	dhdhskhde	sskldnkdns	09-09-2023	Cash	Deleted
4	sdfghjkl	rtyuio	Full Service	Petrol	3456.0	234567897	wertuiop	15-07-2007	Cash	Active
5	sdfhj	sdfghkl;	Full Service	Petrol	34567.0	3456789	dfghjkl;	15-09-2020	UPI	Active
6	dhjgklj	gdfhjj	Full Service	Petrol	345768.0	566789870	tfuyku	16-05-99	Credit Card	Active
7	fhjl;'	asdfhjkl;'	Full Service	Petrol	2.3456789E7	234567890	chdaihoiweh	15-03-2007	Credit Card	Deleted
8	fdgfhg	dfgh	Full Service	Petrol	567987.0	3456768	ghmn	14-12=2000	Cash	Active
9	sdcf	sfg	Oil Change	Diesel	43567.0	233456787	hffggh	22-10-2012	Credit Card	Deleted
10	fghjkl	sdfghj	Full Service	Petrol	500.0	39876546	tiuttani	15-06-2020	Cash	Active
11	saichupak	mahindra	Oil Change	Diesel	500.0	950079870	chennai	14-05-2007	UPI	Deleted
12	santhosh	enova	Tire Change	Diesel	500.0	9876508909	madurai	15-05-2007	Cash	Deleted

Booking Marked as Deleted!

Search:

Fig 5.6 Database creation



Schema: car_service_db

```
1 • SELECT * FROM car_service_db.bookings;
```

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

customer_id	customer_name	car_model	service_type	fuel_type	service_cost	phone_number	address	date_of_birth	payment_method	deleted	created_at
1	df	cghjkl	Full Service	Petrol	34567	23456789	rfghjkdefgh	2022-05-15	Cash	1	2025-10-30
2	ertyuio	dfghjk	Full Service	Petrol	234567	34567890-	errdftgyhujiko	15-05-2006	Cash	1	2025-10-30
3	sskfnksnkfnslsdfls	lsnknwelndekd	Full Service	Petrol	7890	dhdhskhde	sskldnkdns	09-09-2023	Cash	1	2025-10-30
4	sdfghjkl	rtyuio	Full Service	Petrol	3456	234567897	wertuiop	15-07-2007	Cash	0	2025-10-31
5	sdfhj	sdfghkl;	Full Service	Petrol	34567	3456789	dfghjkl;	15-09-2020	UPI	0	2025-10-31
6	dhjgklj	gdfhjj	Full Service	Petrol	345768	566789870	tfuyku	16-05-99	Credit Card	0	2025-10-31
7	fhjl;'	asdfhjkl;'	Full Service	Petrol	23456789	234567890	chdaihoiweh	15-03-2007	Credit Card	1	2025-10-31
8	fdgfhg	dfgh	Full Service	Petrol	567987	3456768	ghmn	14-12=2000	Cash	0	2025-10-31
9	sdcf	sfg	Oil Change	Diesel	43567	233456787	hffggh	22-10-2012	Credit Card	1	2025-10-31
10	fghjkl	sdfghj	Full Service	Petrol	500	39876546	tiuttani	15-06-2020	Cash	0	2025-10-31
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

#	Time	Action	Message	Duration / Fetch
1	21:26:12	SELECT * FROM car_service_db.bookings LIMIT 0, 1000	11 row(s) returned	0.016 sec / 0.000 sec
2	21:26:12	SELECT * FROM car_service_db.bookings LIMIT 0, 1000	11 row(s) returned	0.000 sec / 0.000 sec
3	21:27:15	SELECT * FROM car_service_db.bookings LIMIT 0, 1000	11 row(s) returned	0.000 sec / 0.000 sec
4	21:29:37	SELECT * FROM car_service_db.bookings LIMIT 0, 1000	11 row(s) returned	0.000 sec / 0.000 sec

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

The **Car Service Booking System** efficiently automates the process of scheduling and managing car service appointments, reducing manual effort and errors. It provides a user-friendly interface for customers and administrators to handle bookings smoothly. The system enhances accuracy, saves time, and improves service quality using Java and MySQL integration. In the future, it can be enhanced with online payments, mobile app access, and notification features. Overall, it offers a reliable, scalable, and efficient solution for modern automobile service centres.