

**RiderProvider – Car Rental System**

**MINI PROJECT REPORT**

By

**Jatin Naik (RA2311026010670)**  
**Agamjot Kaur Choudhary (RA2311026010681)**  
**Aditi Onkar (RA2311026010693)**  
**Pushkal Tiwari (RA2311026010697)**  
**Syon Mukherjee (RA2311026010701)**  
**Shreyans Modi (RA2311026010720)**

Under the guidance of

**Dr. JEYASUDHA J**

*In partial fulfillment for the Course*

of

21CSE271T-Programming in Java

In CINTEL



**FACULTY OF ENGINEERING AND TECHNOLOGY**

**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

**Kattankulathur, Chengalpattu District**

MAY 2025

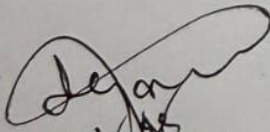
---

# SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

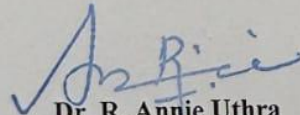
## BONAFIDE CERTIFICATE

Certified that Mini project report titled "Car Rental System" is the bona fide work of Jatin Naik [RA2311026010670], Agamjot Kaur Choudhary [RA2311026010681], Aditi Onkar [RA2311026010693], Pushkal Tiwari [RA2311026010697], Syon Mukherjee [RA2311026010701] and Shreyans Modi [RA2311026010720] who carried out the minor project under my supervision. Certified further, that to the best of my knowledge, the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.



**Dr. Jayasudha J**  
Guide

Assistant Professor  
Department of Computational Intelligence  
SRMIST



**Dr. R. Annie Uthra**  
Head of the Department  
Professor & Head

Department of Computational Intelligence  
SRMIST

## ACKNOWLEDGEMENT

We express our heartfelt thanks to our honorable **Vice Chancellor Dr. C. MUTHAMIZHCHELVAN**, for being the beacon in all our endeavors.

We would like to express my warmth of gratitude to our **Registrar Dr. S. Ponnusamy**, for his encouragement.

We express our profound gratitude to our **Dean (College of Engineering and Technology) Dr. T. V. Gopal**, for bringing out novelty in all executions.

We would like to express my heartfelt thanks to Chairperson, School of Computing **Dr. Revathi Venkataraman**, for imparting confidence to complete my course project

We are highly thankful to my Course project Faculty **Dr. Jeyasudha J, Assistant Professor** for his assistance, timely suggestion and guidance throughout the duration of this course project.

We extend my gratitude to our **HOD Dr. R. Annie Uthra, Department of Computational Intelligence** and my Departmental colleagues for their Support.

Finally, we thank our parents and friends near and dear ones who directly and indirectly contributed to the successful completion of our project.

Above all, I thank the almighty for showering his blessings on me to complete my Course project.

## TABLE OF CONTENTS

Chapter No.	Chapter Title	Page No.
1	ABSTRACT	5
2	PROBLEM DEFINITION	6
3	SOFTWARE ARCHITECTURE	7
4	FEATURES	8
5	SCHEMA DIAGRAM	9
6	SYSTEM REQUIREMENTS	10
7	LIST OF MODULES	11
8	CODING	12 - 45
9	SCREENSHOTS	46 - 48
10	FUTURE PROSPECTS	49
11	CONCLUSION	50

## ABSTRACT

The Vehicle Rental Management System – **RiderProvider** is designed to revolutionize the traditional rental process by automating reservation management, inventory tracking, and customer interactions through an intuitive Java Swing desktop application. This system empowers administrators to efficiently oversee the entire rental operation while providing customers with a seamless booking experience.

The system effectively manages comprehensive details of vehicles, customers, bookings, and payment records in a centralized database. Users can easily register accounts, browse through categorized vehicle options, view real-time availability, complete bookings, and access their reservation history. The interface prioritizes user experience with intuitive navigation and responsive design principles.

Administrative capabilities include monitoring vehicle status, tracking reservations, and generating insightful reports on rental patterns and revenue streams. The system incorporates robust authentication mechanisms to ensure data security and privacy protection for all stored information.

By transitioning from manual record-keeping to this computerized solution, rental businesses can dramatically increase operational efficiency, reduce human error, enhance customer satisfaction, and ultimately improve profitability. The system's modular architecture allows for future expansions and feature enhancements to accommodate evolving business requirements.

This project represents a significant advancement in rental management technology, offering both practical utility for business operations and convenient service delivery for end-users.

## **PROBLEM DEFINITION**

The conventional vehicle rental process relies heavily on manual operations requiring physical presence, extensive paperwork, and time-consuming procedures. Customers typically must visit rental locations in person, complete multiple forms, and navigate through disjointed booking systems. This traditional approach creates numerous inefficiencies including extended waiting times, limited visibility of vehicle availability, tedious documentation, and potential for human error in record-keeping and billing calculations.

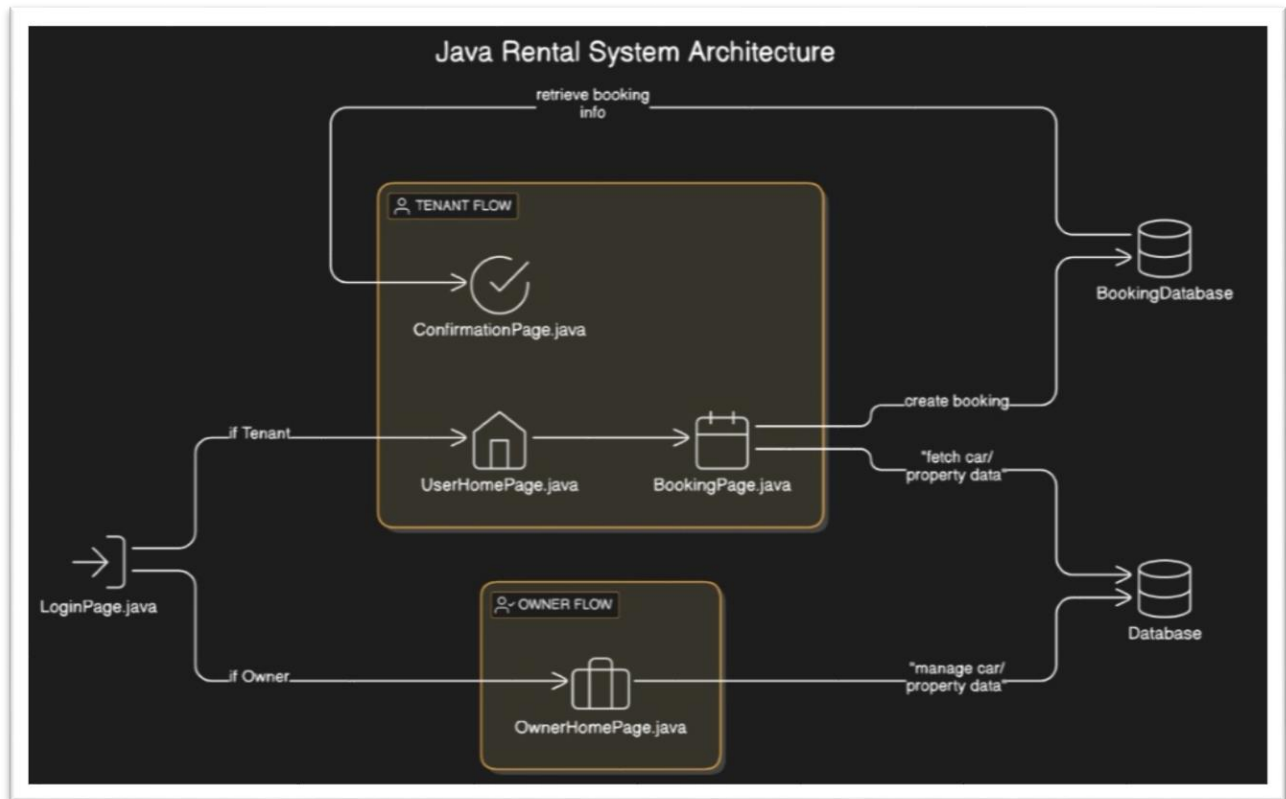
The proposed Vehicle Rental Management System addresses these challenges by creating a comprehensive digital solution that automates and streamlines the entire rental experience. The system replaces manual processes with an integrated platform where customers can seamlessly browse available vehicles, make informed selections based on real-time inventory, and complete bookings without physical presence requirements.

The system features dual access levels: administrative and customer interfaces. Administrators can effectively manage the vehicle fleet, monitor reservation status, update vehicle information, and access customer records through a secure dashboard. Meanwhile, customers benefit from intuitive account creation, personalized booking management, transparent pricing, and convenient access to their rental history.

By digitalizing the rental workflow, the system eliminates paperwork redundancies, reduces processing time, minimizes human error in calculations, and provides a centralized database for all rental operations. The comprehensive solution enhances operational efficiency for rental businesses while simultaneously improving customer satisfaction through convenience, transparency, and reliability in the rental experience.

This transformation from a fragmented manual process to an integrated digital system represents a significant advancement in rental service delivery, benefiting both service providers and customers through modernized, efficient vehicle rental management.

## SYSTEM ARCHITECTURE DIAGRAM



## **FEATURES:**

### **User Management**

- Secure user registration and authentication
- Profile management with rental history tracking

### **Booking System**

- Calendar-based reservation interface
- Transparent pricing structure with automatic calculation
- Duration selection with flexible pickup/return times

### **Reservation Management**

- Booking confirmation with unique reference number
- Modification and cancellation options (subject to policy)

### **Administrative Controls**

- Fleet inventory management dashboard
- Customer database with search functionality
- Reservation tracking and approval workflow

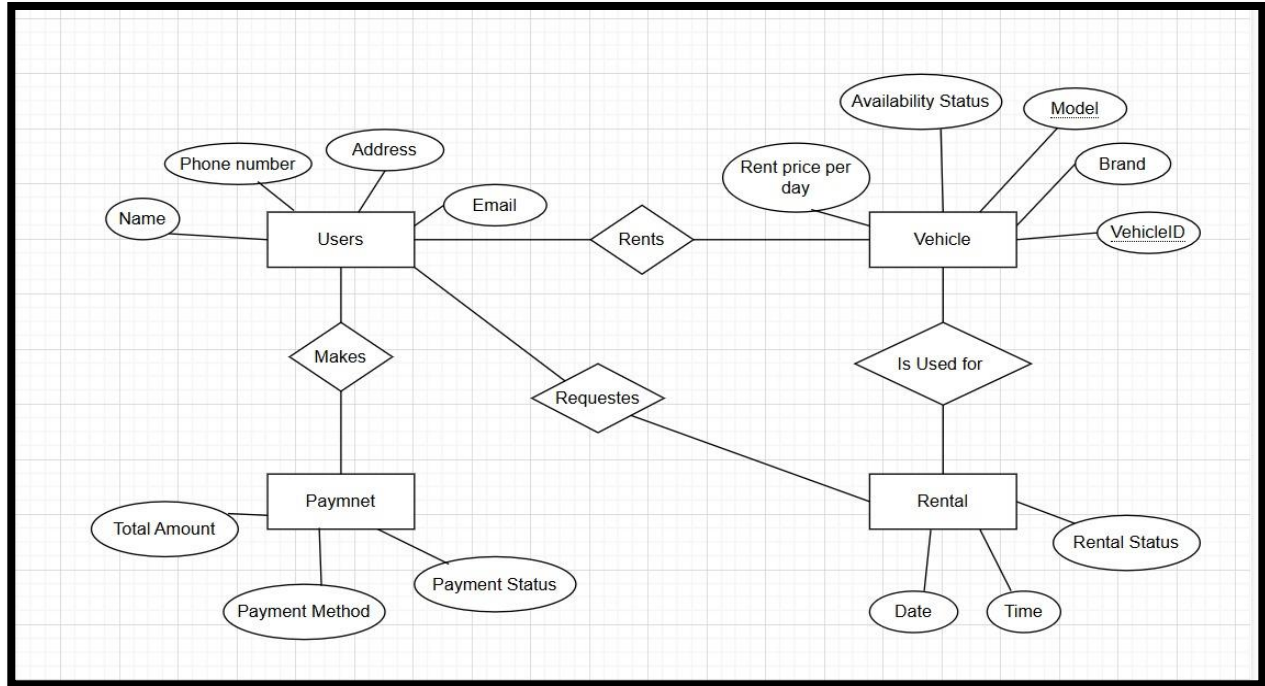
### **Reporting**

- Booking history records accessible to users
- Vehicle utilization analytics for administrators
- Revenue reporting by vehicle category

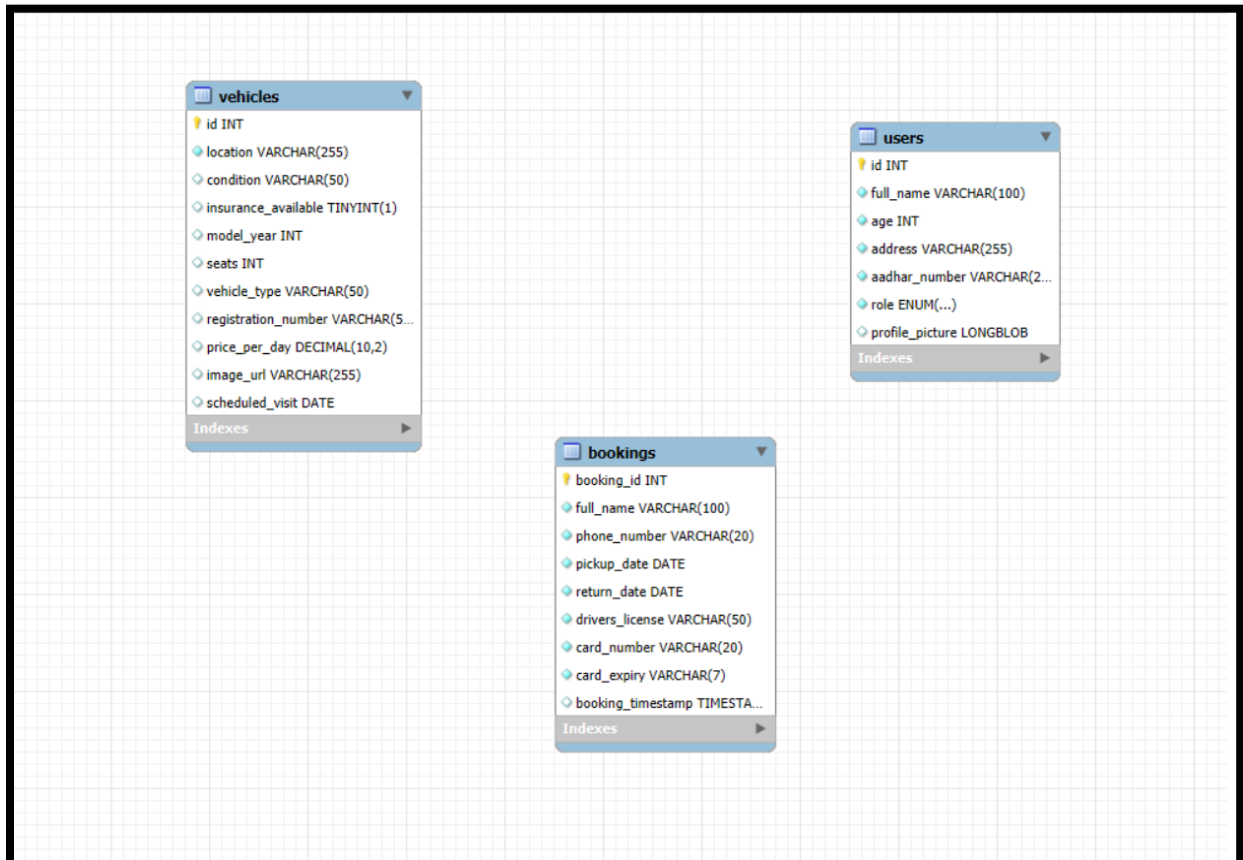
This system eliminates paper-based processes while providing a streamlined rental experience. Administrators gain comprehensive operational oversight while customers enjoy convenient self-service booking capabilities. The solution balances robust functionality with user-friendly design to serve both business requirements and customer expectations in today's digital rental marketplace.



## ER diagram:



## Relational Schema:



# **SYSTEM REQUIREMENTS**

## **Operating System**

- Ubuntu Server 20.04 LTS or Windows Server 2019

## **Database Management System**

- MySQL 8.0 or higher
- Database backup and recovery tools

## **Frontend**

- Java SE Development Kit (JDK) 11 or higher
- Java Swing for GUI components

## **Backend**

- Java
- Springboot

## **Database**

- MySQL 8.0 or higher

## **Development Tools**

- Version control system (Git)
- IDE compatible with Java development
- Database management interface (MySQL Workbench)
- API testing tools

# LIST OF MODULES

## 1. Authentication Module

- **Login Component:** Validates user credentials, manages session creation, and handles authentication errors.
- **Signup Component:** Captures new user information, validates input fields, and creates user accounts in the database.

## 2. Home Module

- **Dashboard Component:** Presents overview of the system with navigation options to all key features.
- **Quick Search:** Enables users to quickly find vehicles based on basic criteria.

## 3. Vehicle Listing Module

- **Filtered Vehicle Display:** Shows available vehicles based on selected category.

## 4. Booking Module

- **Date Selection:** Enables users to select pickup and return dates/times.
- **Vehicle Details Confirmation:** Displays complete information about the selected vehicle.
- **Pricing Calculator:** Automatically calculates rental costs based on duration and vehicle type.
- **Additional Options:** Offers insurance, add-ons, and special requests.

## 5. Confirmation Module

- **Booking Summary:** Shows complete details of the finalized booking.
- **Reference Number Generation:** Creates unique booking identifiers for future reference.

## 6. My Bookings Module

- **Booking History:** Displays all past and upcoming bookings for the logged-in user.
- **Booking Details:** Allows users to view complete information about any specific booking.

## CODING

### SQL:

tables:-

(saved from LoginPage)

```
CREATE TABLE users (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  full_name VARCHAR(100) NOT NULL,  
  age INT NOT NULL,  
  address VARCHAR(255) NOT NULL,  
  aadhar_number VARCHAR(20) UNIQUE NOT NULL,  
  role ENUM('Tenant', 'Owner') NOT NULL,  
  profile_picture LONGBLOB  
);
```

(Saved from OwnerHomePage and displayed in UserHomePage as list of available vehicles)

```
CREATE TABLE vehicles (  
  id INT AUTO_INCREMENT PRIMARY KEY,          -- Unique identifier for the vehicle  
  location VARCHAR(255) NOT NULL,              -- Vehicle location  
  condition VARCHAR(50),                       -- Vehicle condition (Excellent, Good, Fair, Poor)  
  insurance_available BOOLEAN,                 -- Whether insurance is available  
  model_year INT,                             -- Vehicle model year  
  seats INT,                                  -- Number of seats  
  vehicle_type VARCHAR(50),                   -- Type of vehicle (Car, Bike, SUV, Van)  
  registration_number VARCHAR(50),            -- Vehicle registration number  
  price_per_day DECIMAL(10, 2),               -- Price per day  
  image_url VARCHAR(255),                     -- Path to the uploaded image (for preview)  
  scheduled_visit DATE                        -- Scheduled visit date  
);
```

(Saved from BookingPage)

```
CREATE TABLE bookings (  
  booking_id INT AUTO_INCREMENT PRIMARY KEY,  
  full_name VARCHAR(100) NOT NULL,  
  phone_number VARCHAR(20) NOT NULL,  
  pickup_date DATE NOT NULL,  
  return_date DATE NOT NULL,  
  drivers_license VARCHAR(50) NOT NULL,  
  card_number VARCHAR(20) NOT NULL,  
  card_expiry VARCHAR(7) NOT NULL, -- Format: MM/YY  
  booking_timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

## Java:

### Login.java

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.io.*;
import java.sql.*;

import javax.swing.border.TitledBorder;

public class LoginPage extends JFrame {
    // Color palette
    private static final Color PRIMARY_COLOR = new Color(54, 116, 181);
    private static final Color SECONDARY_COLOR = new Color(87, 143, 202);
    private static final Color ACCENT_COLOR = new Color(161, 227, 249);
    private static final Color BACKGROUND_COLOR = new Color(209, 248, 239);
    private static final int PADDING = 15;

    private JTextField nameField, ageField, addressField, aadharField;
    private JComboBox<String> roleComboBox;
    private File selectedImageFile;

    public LoginPage() {
        setTitle("RiderProvider - Login");
        setSize(500, 750);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
        setLayout(new BorderLayout(0, 10));
        getContentPane().setBackground(BACKGROUND_COLOR);

        JPanel header = createHeader();
        add(header, BorderLayout.NORTH);

        JPanel contentPanel = new JPanel();
        contentPanel.setLayout(new BoxLayout(contentPanel, BoxLayout.Y_AXIS));
        contentPanel.setBorder(BorderFactory.createEmptyBorder(PADDING, PADDING, PADDING, PADDING));
        contentPanel.setBackground(BACKGROUND_COLOR);

        JPanel rolePanel = createRoleSelectionPanel();
        contentPanel.add(rolePanel);
        contentPanel.add(Box.createVerticalStrut(20));

        JPanel formPanel = createFormPanel();
        contentPanel.add(formPanel);

        add(contentPanel, BorderLayout.CENTER);

        JPanel bottomPanel = createBottomPanel();
        add(bottomPanel, BorderLayout.SOUTH);
    }
}
```

```

    setVisible(true);
}

private JPanel createHeader() {
    JPanel header = new JPanel(new BorderLayout());
    header.setBackground(PRIMARY_COLOR);
    header.setBorder(BorderFactory.createEmptyBorder(15, 20, 15, 20));
    header.setPreferredSize(new Dimension(getWidth(), 80));

    JPanel logoPanel = new JPanel(new FlowLayout(FlowLayout.LEFT, 15, 0));
    logoPanel.setOpaque(false);

    ImageIcon originalIcon = new ImageIcon("logo.png");
    Image scaledImage = originalIcon.getImage().getScaledInstance(50, 50, Image.SCALE_SMOOTH);
    ImageIcon scaledIcon = new ImageIcon(scaledImage);

    JLabel logo = new JLabel(scaledIcon);
    JLabel companyName = new JLabel("RiderProvider");
    companyName.setFont(new Font("Arial", Font.BOLD, 24));
    companyName.setForeground(Color.WHITE);

    logoPanel.add(logo);
    logoPanel.add(companyName);

    header.add(logoPanel, BorderLayout.WEST);

    return header;
}

private JPanel createRoleSelectionPanel() {
    JPanel rolePanel = new JPanel();
    rolePanel.setLayout(new BoxLayout(rolePanel, BoxLayout.Y_AXIS));
    rolePanel.setBorder(BorderFactory.createTitledBorder(
        BorderFactory.createLineBorder(SECONDARY_COLOR, 2, true), "User Role"));
    rolePanel.setBackground(BACKGROUND_COLOR);
    rolePanel.setAlignmentX(Component.LEFT_ALIGNMENT);

    JLabel selectRole = new JLabel("Select Role:");
    selectRole.setFont(new Font("Arial", Font.BOLD, 16));
    selectRole.setForeground(PRIMARY_COLOR);
    selectRole.setAlignmentX(Component.LEFT_ALIGNMENT);
    selectRole.setBorder(BorderFactory.createEmptyBorder(5, 5, 8, 0));

    String[] roles = {"Select", "Tenant", "Owner"};
    roleComboBox = new JComboBox<>(roles);
    roleComboBox.setFont(new Font("Arial", Font.PLAIN, 16));
    roleComboBox.setBackground(Color.WHITE);
    roleComboBox.setForeground(PRIMARY_COLOR);
    roleComboBox.setMaximumSize(new Dimension(Integer.MAX_VALUE,
        roleComboBox.getPreferredSize().height + 5));
    roleComboBox.setAlignmentX(Component.LEFT_ALIGNMENT);
}

```

```

rolePanel.add(selectRole);
rolePanel.add(roleComboBox);

return rolePanel;
}

private JPanel createFormPanel() {
    JPanel formPanel = new JPanel();
    formPanel.setLayout(new BoxLayout(formPanel, BoxLayout.Y_AXIS));
    formPanel.setBorder(BorderFactory.createTitledBorder(
        BorderFactory.createLineBorder(SECONDARY_COLOR, 2, true), "User Information"));
    formPanel.setBackground(BACKGROUND_COLOR);
    formPanel.setAlignmentX(Component.LEFT_ALIGNMENT);

    nameField = createFormField("Full Name");
    ageField = createFormField("Age");
    addressField = createFormField("Address");
    aadharField = createFormField("Aadhar Number");

    JButton uploadPhotoButton = new JButton("Upload Profile Picture");
    uploadPhotoButton.setFont(new Font("Arial", Font.BOLD, 16));
    uploadPhotoButton.setAlignmentX(Component.LEFT_ALIGNMENT);
    uploadPhotoButton.setMaximumSize(new Dimension(Integer.MAX_VALUE,
uploadPhotoButton.getPreferredSize().height + 5));
    uploadPhotoButton.setBackground(PRIMARY_COLOR);
    uploadPhotoButton.setForeground(Color.BLUE);
    uploadPhotoButton.setFocusPainted(false);
    uploadPhotoButton.setCursor(new Cursor(Cursor.HAND_CURSOR));

    uploadPhotoButton.addActionListener(e -> {
        JFileChooser fileChooser = new JFileChooser();
        fileChooser.setDialogTitle("Select Profile Picture");
        fileChooser.setAcceptAllFileFilterUsed(false);
        fileChooser.addChoosableFileFilter(new javax.swing.filechooser.FileNameExtensionFilter("Image files",
"jpg", "jpeg", "png", "bmp", "gif"));

        int result = fileChooser.showOpenDialog(this);
        if (result == JFileChooser.APPROVE_OPTION) {
            selectedImageFile = fileChooser.getSelectedFile();
            JOptionPane.showMessageDialog(this, "Selected File: " + selectedImageFile.getName(), "Photo Selected",
JOptionPane.INFORMATION_MESSAGE);
        }
    });

    formPanel.add(nameField);
    formPanel.add(Box.createVerticalStrut(PADDING));
    formPanel.add(ageField);
    formPanel.add(Box.createVerticalStrut(PADDING));
    formPanel.add(addressField);
    formPanel.add(Box.createVerticalStrut(PADDING));
    formPanel.add(aadharField);
    formPanel.add(Box.createVerticalStrut(PADDING));
    formPanel.add(uploadPhotoButton);

```

```

    return formPanel;
}

private JTextField createFormField(String title) {
    JTextField field = new JTextField();
    Font titleFont = new Font("Arial", Font.BOLD, 16);
    field.setFont(new Font("Arial", Font.PLAIN, 16));
    field.setBackground(Color.WHITE);
    field.setForeground(new Color(50, 50, 50));

    TitledBorder titledBorder = BorderFactory.createTitledBorder(
        BorderFactory.createLineBorder(SECONDARY_COLOR), title);
    titledBorder.setTitleFont(titleFont);
    titledBorder.setTitleColor(PRIMARY_COLOR);

    field.setBorder(BorderFactory.createCompoundBorder(
        titledBorder,
        BorderFactory.createEmptyBorder(8, 8, 8, 8)));
    field.setAlignmentX(Component.LEFT_ALIGNMENT);
    field.setMaximumSize(new Dimension(Integer.MAX_VALUE, field.getPreferredSize().height + 20));
    return field;
}

private JPanel createBottomPanel() {
    JPanel bottomPanel = new JPanel(new FlowLayout(FlowLayout.CENTER));
    bottomPanel.setBorder(BorderFactory.createEmptyBorder(5, PADDING, PADDING, PADDING));
    bottomPanel.setBackground(BACKGROUND_COLOR);

    JButton submitButton = new JButton("Submit");
    submitButton.setPreferredSize(new Dimension(200, 50));
    submitButton.setBackground(PRIMARY_COLOR);
    submitButton.setForeground(Color.WHITE);
    submitButton.setFocusPainted(false);
    submitButton.setBorderPainted(false);
    submitButton.setCursor(new Cursor(Cursor.HAND_CURSOR));
    submitButton.setFont(new Font("Arial", Font.BOLD, 18));

    submitButton.addActionListener(e -> {
        try {
            String selectedRole = (String) roleComboBox.getSelectedItem();
            if (selectedRole.equals("Select")) {
                JOptionPane.showMessageDialog(this, "Please select a valid role.", "Error",
JOptionPane.ERROR_MESSAGE);
                return;
            }

            String name = nameField.getText();
            int age = Integer.parseInt(ageField.getText());
            String address = addressField.getText();
            String aadhar = aadharField.getText();

            Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/RiderProvider", "root",

```



```
"Jatin#Naik@31");
```

```
PreparedStatement stmt = conn.prepareStatement(  
    "INSERT INTO users (full_name, age, address, aadhar_number, role, profile_picture) VALUES (?, ?,  
    ?, ?, ?, ?)");
```

```
    stmt.setString(1, name);  
    stmt.setInt(2, age);  
    stmt.setString(3, address);  
    stmt.setString(4, aadhar);  
    stmt.setString(5, selectedRole);
```

```
    if (selectedImageFile != null) {  
        FileInputStream fis = new FileInputStream(selectedImageFile);  
        stmt.setBinaryStream(6, fis, (int) selectedImageFile.length());  
    } else {  
        stmt.setNull(6, Types.BLOB);  
    }  
}
```

```
int result = stmt.executeUpdate();  
if (result > 0) {  
    JOptionPane.showMessageDialog(this, "User registered successfully!");  
    if (selectedRole.equals("Tenant")) {  
        new UserHomePage();  
    } else {  
        new OwnerHomePage();  
    }  
    dispose();  
}
```

```
stmt.close();  
conn.close();
```

```
    } catch (Exception ex) {  
        ex.printStackTrace();  
        JOptionPane.showMessageDialog(this, "Error: " + ex.getMessage(), "Database Error",  
JOptionPane.ERROR_MESSAGE);  
    }  
});
```

```
bottomPanel.add(submitButton);  
return bottomPanel;  
}
```

```
public static void main(String[] args) {  
    try {  
        UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

```
SwingUtilities.invokeLater(() -> new LoginPage());  
}
```

```
}
```

## ConformationPage.java

```
import javax.swing.*;
import java.awt.*;

public class ConfirmationPage extends JFrame {

    // Color constants
    private static final Color PRIMARY_COLOR = new Color(54, 116, 181); // #3674B5
    private static final Color SECONDARY_COLOR = new Color(87, 143, 202); // #578FCA
    private static final Color ACCENT_COLOR = new Color(161, 227, 249); // #A1E3F9
    private static final Color BACKGROUND_COLOR = new Color(209, 248, 239); // #D1F8EF
    private static final int PADDING = 15;

    public ConfirmationPage(String name, String phone, String pickupDate, String returnDate, String license, String
card, String expiry) {
        setTitle("Booking Confirmation");
        setSize(500, 500);
        setLocationRelativeTo(null);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // Main panel with padding and background
        JPanel mainPanel = new JPanel();
        mainPanel.setLayout(new BorderLayout(PADDING, PADDING));
        mainPanel.setBackground(BACKGROUND_COLOR);
        mainPanel.setBorder(BorderFactory.createEmptyBorder(PADDING, PADDING, PADDING, PADDING));

        // Add custom styled header
        mainPanel.add(createHeader(), BorderLayout.NORTH);

        // Info Panel
        JPanel infoPanel = new JPanel(new GridLayout(7, 1, 10, 10));
        infoPanel.setOpaque(false); // so background color shows through

        infoPanel.add(new JLabel("Name: " + name));
        infoPanel.add(new JLabel("Phone: " + phone));
        infoPanel.add(new JLabel("Pickup Date: " + pickupDate));
        infoPanel.add(new JLabel("Return Date: " + returnDate));
        infoPanel.add(new JLabel("License Number: " + license));
        infoPanel.add(new JLabel("Card Number: " + card));
        infoPanel.add(new JLabel("Card Expiry Date: " + expiry));

        // Style the labels
        for (Component c : infoPanel.getComponents()) {
            c.setFont(new Font("SansSerif", Font.PLAIN, 16));
        }

        mainPanel.add(infoPanel, BorderLayout.CENTER);
    }
}
```

```

    add(mainPanel);
    setVisible(true);
}

private JPanel createHeader() {
    JPanel header = new JPanel(new BorderLayout());
    header.setBackground(PRIMARY_COLOR);
    header.setBorder(BorderFactory.createEmptyBorder(PADDING, PADDING, PADDING, PADDING));
    header.setPreferredSize(new Dimension(getWidth(), 80));

    // Left: Logo and Company Name
    JPanel logoPanel = new JPanel(new FlowLayout(FlowLayout.LEFT, 15, 0));
    logoPanel.setOpaque(false);

    ImageIcon originalIcon = new ImageIcon(getClass().getResource("/logo.png"));
    Image scaledImage = originalIcon.getImage().getScaledInstance(50, 50, Image.SCALE_SMOOTH);
    ImageIcon scaledIcon = new ImageIcon(scaledImage);
    JLabel logo = new JLabel(scaledIcon);

    JLabel companyName = new JLabel("RiderProvider");
    companyName.setFont(new Font("Arial", Font.BOLD, 24));
    companyName.setForeground(Color.WHITE);

    logoPanel.add(logo);
    logoPanel.add(companyName);
    header.add(logoPanel, BorderLayout.WEST);

    // Center: Page Title
    JLabel pageTitle = new JLabel("☑ Booking Confirmed!", SwingConstants.CENTER);
    pageTitle.setFont(new Font("Arial", Font.BOLD, 24));
    pageTitle.setForeground(Color.WHITE);
    header.add(pageTitle, BorderLayout.CENTER);

    // Right: Empty or optional Profile Button
    JPanel profilePanel = new JPanel(new FlowLayout(FlowLayout.RIGHT));
    profilePanel.setOpaque(false);
    header.add(profilePanel, BorderLayout.EAST);

    return header;
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(() -> new ConfirmationPage(
        "Tabitha",           // name
        "1234567890",        // phone
        "2025-05-04",        // pickupDate
        "2025-05-07",        // returnDate
        "TN01-123456",        // license
        "1234-5678-9012",    // card
        "11/25"              // expiry
    ));
}

```

```
}  
}
```

## BookingPage.java

```
import javax.swing.*.*;  
import java.awt.*.*;  
import java.awt.event.*;  
import java.awt.image.BufferedImage;  
import javax.swing.border.*;  
import java.time.LocalDate;  
import java.time.format.DateTimeFormatter;  
import java.sql.*;  
  
public class BookingPage {  
    // Color palette and Fonts (same as original)  
    private static final Color PRIMARY_COLOR = new Color(54, 116, 181);    // #3674B5  
    private static final Color SECONDARY_COLOR = new Color(87, 143, 202); // #578FCA  
    private static final Color ACCENT_COLOR = new Color(161, 227, 249);    // #A1E3F9  
    private static final Color BACKGROUND_COLOR = new Color(209, 248, 239); // #D1F8EF  
    private static final Color TEXT_COLOR = new Color(33, 33, 33);  
  
    private static final Font TITLE_FONT = new Font("Segoe UI", Font.BOLD, 28);  
    private static final Font SUBTITLE_FONT = new Font("Segoe UI", Font.BOLD, 18);  
    private static final Font REGULAR_FONT = new Font("Segoe UI", Font.PLAIN, 14);  
  
    private JFrame frame;  
    private UserHomePage homePageRef; // Reference to home page to return back  
  
    public BookingPage(String carModel, String price, String year, String location, UserHomePage homePage) {  
        this.homePageRef = homePage;  
  
        try {  
            UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
  
        frame = new JFrame("RiderProvider - Book Your Ride");  
        frame.setSize(900, 700);  
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        frame.setLocationRelativeTo(null);  
  
        // Main panel  
        JPanel mainPanel = new JPanel(new BorderLayout());  
        mainPanel.setBackground(BACKGROUND_COLOR);  
  
        // Header panel  
        JPanel headerPanel = createHeaderPanel();  
  
        // Booking form panel  
        JPanel contentPanel = new JPanel();
```

```

contentPanel.setLayout(new BoxLayout(contentPanel, BoxLayout.Y_AXIS));
contentPanel.setBackground(BACKGROUND_COLOR);
contentPanel.setBorder(BorderFactory.createEmptyBorder(20, 40, 20, 40));

// Car details summary panel
JPanel carSummaryPanel = createCarSummaryPanel(carModel, price, year, location);

// Booking form
JPanel bookingFormPanel = createBookingFormPanel();

// Add panels to content
contentPanel.add(carSummaryPanel);
contentPanel.add(Box.createRigidArea(new Dimension(0, 20)));
contentPanel.add(bookingFormPanel);

// Add all to main panel
mainPanel.add(headerPanel, BorderLayout.NORTH);
mainPanel.add(new JScrollPane(contentPanel), BorderLayout.CENTER);

frame.setContentPane(mainPanel);
frame.setVisible(true);
}

private JPanel createHeaderPanel() {
    JPanel header = new JPanel(new BorderLayout());
    header.setBackground(PRIMARY_COLOR);
    header.setBorder(BorderFactory.createEmptyBorder(15, 20, 15, 20));
    header.setPreferredSize(new Dimension(900, 80));

    // Panel for logo and company name
    JPanel logoPanel = new JPanel(new FlowLayout(FlowLayout.LEFT, 15, 0));
    logoPanel.setOpaque(false);

    // Create placeholder logo
    ImageIcon originalIcon = new ImageIcon("logo.png");
    Image scaledImage = originalIcon.getImage().getScaledInstance(50, 50, Image.SCALE_SMOOTH);
    ImageIcon scaledIcon = new ImageIcon(scaledImage);
    JLabel logo = new JLabel(scaledIcon);

    // Company name label
    JLabel companyName = new JLabel("RiderProvider");
    companyName.setFont(new Font("Segoe UI", Font.BOLD, 24));
    companyName.setForeground(Color.WHITE);

    logoPanel.add(logo);
    logoPanel.add(companyName);
    header.add(logoPanel, BorderLayout.WEST);

    // Page title
    JLabel pageTitle = new JLabel("Book Your Ride", SwingConstants.CENTER);
    pageTitle.setFont(TITLE_FONT);
    pageTitle.setForeground(Color.WHITE);
    header.add(pageTitle, BorderLayout.CENTER);
}

```

```

// Back button
JPanel navPanel = new JPanel(new FlowLayout(FlowLayout.RIGHT));
navPanel.setOpaque(false);
JButton backButton = new JButton("Back to Listings");
backButton.setFont(new Font("Segoe UI", Font.BOLD, 16));
backButton.setBackground(SECONDARY_COLOR);
backButton.setForeground(Color.WHITE);
backButton.setFocusPainted(false);
backButton.setCursor(new Cursor(Cursor.HAND_CURSOR));

// Add action to return to UserHomePage
backButton.addActionListener(e -> {
    frame.dispose(); // Close current frame
    if (homePageRef != null) {
        homePageRef.showFrame(); // Show the home page frame
    }
});

navPanel.add(backButton);
header.add(navPanel, BorderLayout.EAST);

return header;
}

private JPanel createCarSummaryPanel(String carModel, String price, String year, String location) {
    JPanel panel = new JPanel();
    panel.setLayout(new BoxLayout(panel, BoxLayout.Y_AXIS));
    panel.setBackground(Color.WHITE);
    panel.setBorder(BorderFactory.createCompoundBorder(
        new LineBorder(ACCENT_COLOR, 1, true),
        BorderFactory.createEmptyBorder(20, 20, 20, 20)
    ));
    panel.setAlignmentX(Component.LEFT_ALIGNMENT);

    // Title for the summary
    JLabel summaryTitle = new JLabel("Booking Summary");
    summaryTitle.setFont(SUBTITLE_FONT);
    summaryTitle.setForeground(PRIMARY_COLOR);
    summaryTitle.setAlignmentX(Component.LEFT_ALIGNMENT);

    // Car info panel with image and details
    JPanel carInfoPanel = new JPanel(new BorderLayout(20, 0));
    carInfoPanel.setBackground(Color.WHITE);
    carInfoPanel.setAlignmentX(Component.LEFT_ALIGNMENT);

    // Car image
    JLabel carImage = new JLabel(createCarPlaceholderIcon(120, 90));
    carImage.setBorder(BorderFactory.createLineBorder(new Color(220, 220, 220), 1));

    // Car details panel
    JPanel detailsPanel = new JPanel(new GridLayout(4, 1, 0, 5));
    detailsPanel.setBackground(Color.WHITE);

```

```
JLabel modelLabel = new JLabel("Model: " + carModel);
modelLabel.setFont(new Font("Segoe UI", Font.BOLD, 16));
modelLabel.setForeground(TEXT_COLOR);
```

```
JLabel priceLabel = new JLabel("Rental Price: " + price);
priceLabel.setFont(REGULAR_FONT);
priceLabel.setForeground(TEXT_COLOR);
```

```
JLabel yearLabel = new JLabel("Year: " + year);
yearLabel.setFont(REGULAR_FONT);
yearLabel.setForeground(TEXT_COLOR);
```

```
JLabel locationLabel = new JLabel("Pickup Location: " + location);
locationLabel.setFont(REGULAR_FONT);
locationLabel.setForeground(TEXT_COLOR);
```

```
detailsPanel.add(modelLabel);
detailsPanel.add(priceLabel);
detailsPanel.add(yearLabel);
detailsPanel.add(locationLabel);
```

```
carInfoPanel.add(carImage, BorderLayout.WEST);
carInfoPanel.add(detailsPanel, BorderLayout.CENTER);
```

```
panel.add(summaryTitle);
panel.add(Box.createRigidArea(new Dimension(0, 15)));
panel.add(carInfoPanel);
```

```
return panel;
```

```
}
```

```
private JPanel createBookingFormPanel() {
    JPanel formPanel = new JPanel();
    formPanel.setLayout(new BoxLayout(formPanel, BoxLayout.Y_AXIS));
    formPanel.setBackground(Color.WHITE);
    formPanel.setBorder(BorderFactory.createCompoundBorder(
        new LineBorder(ACCENT_COLOR, 1, true),
        BorderFactory.createEmptyBorder(20, 20, 20, 20)
    ));
    formPanel.setAlignmentX(Component.LEFT_ALIGNMENT);
```

```
// Form title
```

```
JLabel formTitle = new JLabel("Booking Details");
formTitle.setFont(SUBTITLE_FONT);
formTitle.setForeground(PRIMARY_COLOR);
formTitle.setAlignmentX(Component.LEFT_ALIGNMENT);
```

```
// Date selection panel
```

```
JPanel datePanel = new JPanel(new GridLayout(2, 2, 15, 15));
datePanel.setBackground(Color.WHITE);
datePanel.setAlignmentX(Component.LEFT_ALIGNMENT);
```

```

// Date labels
JLabel pickupLabel = new JLabel("Pickup Date:");
pickupLabel.setFont(new Font("Segoe UI", Font.BOLD, 14));
pickupLabel.setForeground(TEXT_COLOR);

JLabel returnLabel = new JLabel("Return Date:");
returnLabel.setFont(new Font("Segoe UI", Font.BOLD, 14));
returnLabel.setForeground(TEXT_COLOR);

// Date pickers - using JTextField as placeholder
JTextField pickupDate = new JTextField(LocalDate.now().format(DateTimeFormatter.ofPattern("yyyy-MM-dd")));
pickupDate.setFont(REGULAR_FONT);

JTextField returnDate = new
TextField(LocalDate.now().plusDays(3).format(DateTimeFormatter.ofPattern("yyyy-MM-dd")));
returnDate.setFont(REGULAR_FONT);

datePanel.add(pickupLabel);
datePanel.add(pickupDate);
datePanel.add(returnLabel);
datePanel.add(returnDate);

// Driver details panel
JPanel driverPanel = new JPanel(new GridLayout(3, 2, 15, 15));
driverPanel.setBackground(Color.WHITE);
driverPanel.setAlignmentX(Component.LEFT_ALIGNMENT);

// Labels
JLabel nameLabel = new JLabel("Full Name:");
nameLabel.setFont(new Font("Segoe UI", Font.BOLD, 14));
nameLabel.setForeground(TEXT_COLOR);

JLabel phoneLabel = new JLabel("Phone Number:");
phoneLabel.setFont(new Font("Segoe UI", Font.BOLD, 14));
phoneLabel.setForeground(TEXT_COLOR);

JLabel licenseLabel = new JLabel("Driver's License #:");
licenseLabel.setFont(new Font("Segoe UI", Font.BOLD, 14));
licenseLabel.setForeground(TEXT_COLOR);

// Text fields
JTextField nameField = new JTextField();
nameField.setFont(REGULAR_FONT);

JTextField phoneField = new JTextField();
phoneField.setFont(REGULAR_FONT);

JTextField licenseField = new JTextField();
licenseField.setFont(REGULAR_FONT);

driverPanel.add(nameLabel);
driverPanel.add(nameField);

```



```
driverPanel.add(phoneLabel);
driverPanel.add(phoneField);
driverPanel.add(licenseLabel);
driverPanel.add(licenseField);
```

```
// Payment section
```

```
JPanel paymentPanel = new JPanel(new GridLayout(2, 2, 15, 15));
paymentPanel.setBackground(Color.WHITE);
paymentPanel.setAlignmentX(Component.LEFT_ALIGNMENT);
```

```
JLabel cardLabel = new JLabel("Card Number:");
cardLabel.setFont(new Font("Segoe UI", Font.BOLD, 14));
cardLabel.setForeground(TEXT_COLOR);
```

```
JLabel expiryLabel = new JLabel("Expiry (MM/YY):");
expiryLabel.setFont(new Font("Segoe UI", Font.BOLD, 14));
expiryLabel.setForeground(TEXT_COLOR);
```

```
JTextField cardField = new JTextField();
cardField.setFont(REGULAR_FONT);
```

```
JTextField expiryField = new JTextField();
expiryField.setFont(REGULAR_FONT);
```

```
paymentPanel.add(cardLabel);
paymentPanel.add(cardField);
paymentPanel.add(expiryLabel);
paymentPanel.add(expiryField);
```

```
// Terms and conditions checkbox
```

```
JPanel termsPanel = new JPanel(new FlowLayout(FlowLayout.LEFT));
termsPanel.setBackground(Color.WHITE);
termsPanel.setAlignmentX(Component.LEFT_ALIGNMENT);
```

```
JCheckBox termsCheckbox = new JCheckBox("I agree to the terms and conditions");
termsCheckbox.setFont(REGULAR_FONT);
termsCheckbox.setBackground(Color.WHITE);
```

```
termsPanel.add(termsCheckbox);
```

```
// Button panel
```

```
JPanel buttonPanel = new JPanel(new FlowLayout(FlowLayout.RIGHT));
buttonPanel.setBackground(Color.WHITE);
buttonPanel.setAlignmentX(Component.LEFT_ALIGNMENT);
```

```
JButton cancelButton = new JButton("Cancel");
cancelButton.setFont(new Font("Segoe UI", Font.BOLD, 14));
cancelButton.setForeground(TEXT_COLOR);
cancelButton.setBackground(Color.WHITE);
cancelButton.setBorder(new LineBorder(SECONDARY_COLOR, 1));
cancelButton.setPreferredSize(new Dimension(120, 40));
cancelButton.setCursor(new Cursor(Cursor.HAND_CURSOR));
cancelButton.setFocusPainted(false);
```

```

// Add action to return to UserHomePage
cancelButton.addActionListener(e -> {
    frame.dispose(); // Close current frame
    if (homePageRef != null) {
        homePageRef.showFrame(); // Show the home page frame
    }
});

JButton confirmButton = new JButton("Confirm Booking");
confirmButton.setFont(new Font("Segoe UI", Font.BOLD, 14));
confirmButton.setForeground(Color.WHITE);
confirmButton.setBackground(PRIMARY_COLOR);
confirmButton.setBorderPainted(false);
confirmButton.setPreferredSize(new Dimension(150, 40));
confirmButton.setCursor(new Cursor(Cursor.HAND_CURSOR));
confirmButton.setFocusPainted(false);

confirmButton.addActionListener(e -> {
    if (!termsCheckbox.isSelected()) {
        JOptionPane.showMessageDialog(frame,
            "Please agree to the terms and conditions to proceed.",
            "Terms Required",
            JOptionPane.WARNING_MESSAGE);
        return;
    }

    if (nameField.getText().trim().isEmpty() ||
        phoneField.getText().trim().isEmpty() ||
        licenseField.getText().trim().isEmpty() ||
        cardField.getText().trim().isEmpty() ||
        expiryField.getText().trim().isEmpty())
    {
        JOptionPane.showMessageDialog(frame, "Please fill in all the fields before confirming.", "Incomplete
Form", JOptionPane.WARNING_MESSAGE);
        return;
    }
    frame.dispose(); // Close current frame
    new ConfirmationPage(
        nameField.getText(),
        phoneField.getText(),
        pickupDate.getText(),
        returnDate.getText(),
        licenseField.getText(),
        cardField.getText(),
        expiryField.getText()
    );
});
buttonPanel.add(cancelButton);
buttonPanel.add(Box.createRigidArea(new Dimension(10, 0)));
buttonPanel.add(confirmButton);

// Add everything to the form panel

```

```

formPanel.add(formTitle);
formPanel.add(Box.createRigidArea(new Dimension(0, 15)));
formPanel.add(new JLabel("Please select your rental dates:"));
formPanel.add(Box.createRigidArea(new Dimension(0, 5)));
formPanel.add(datePanel);
formPanel.add(Box.createRigidArea(new Dimension(0, 20)));
formPanel.add(new JLabel("Driver Information:"));
formPanel.add(Box.createRigidArea(new Dimension(0, 5)));
formPanel.add(driverPanel);
formPanel.add(Box.createRigidArea(new Dimension(0, 20)));
formPanel.add(new JLabel("Payment Details:"));
formPanel.add(Box.createRigidArea(new Dimension(0, 5)));
formPanel.add(paymentPanel);
formPanel.add(Box.createRigidArea(new Dimension(0, 15)));
formPanel.add(termsPanel);
formPanel.add(Box.createRigidArea(new Dimension(0, 15)));
formPanel.add(buttonPanel);

return formPanel;
}

// Utility methods for placeholder images
private ImageIcon createPlaceholderIcon(int width, int height) {
    BufferedImage img = new BufferedImage(width, height, BufferedImage.TYPE_INT_ARGB);
    Graphics2D g2d = img.createGraphics();

    g2d.setColor(SECONDARY_COLOR);
    g2d.fillRoundRect(0, 0, width, height, 10, 10);

    g2d.setColor(PRIMARY_COLOR);
    g2d.setFont(new Font("Segoe UI", Font.BOLD, 24));
    FontMetrics fm = g2d.getFontMetrics();
    String text = "RP";
    int textWidth = fm.stringWidth(text);
    int textHeight = fm.getHeight();
    g2d.drawString(text, (width - textWidth) / 2, height / 2 + textHeight / 4);

    g2d.dispose();
    return new ImageIcon(img);
}

private ImageIcon createCarPlaceholderIcon(int width, int height) {
    BufferedImage img = new BufferedImage(width, height, BufferedImage.TYPE_INT_ARGB);
    Graphics2D g2d = img.createGraphics();

    g2d.setColor(BACKGROUND_COLOR);
    g2d.fillRect(0, 0, width, height);

    g2d.setColor(SECONDARY_COLOR);
    g2d.setStroke(new BasicStroke(2));
    g2d.drawRoundRect(10, 5, width - 20, height - 10, 10, 10);

    // Draw simple car shape

```

```

int carWidth = width - 40;
int carHeight = height - 30;
int startX = 20;
int startY = 15;

g2d.setColor(PRIMARY_COLOR);
// Car body
g2d.fillRect(startX, startY + carHeight/3, carWidth, carHeight/2, 10, 10);
// Car top
g2d.fillRect(startX + carWidth/4, startY, carWidth/2, carHeight/3, 8, 8);
// Wheels
g2d.setColor(new Color(60, 60, 60));
g2d.fillOval(startX + carWidth/5, startY + carHeight*2/3, carWidth/5, carHeight/5);
g2d.fillOval(startX + carWidth*3/5, startY + carHeight*2/3, carWidth/5, carHeight/5);

g2d.dispose();
return new ImageIcon(img);
}

// Function to confirm booking and insert into the database
private void confirmBooking(String name, String phone, String license,
String pickupDate, String returnDate, String cardNumber, String expiry) {

try (Connection connection = DriverManager.getConnection(
"jdbc:mysql://localhost:3306/car_rental_db", "root", "password")) {

String query = "INSERT INTO bookings (name, phone, license, pickup_date, return_date, card_number,
expiry_date) " +
"VALUES (?, ?, ?, ?, ?, ?, ?)";

try (PreparedStatement pst = connection.prepareStatement(query)) {
pst.setString(1, name);
pst.setString(2, phone);
pst.setString(3, license);
pst.setString(4, pickupDate);
pst.setString(5, returnDate);
pst.setString(6, cardNumber);
pst.setString(7, expiry);

int rowsAffected = pst.executeUpdate();

if (rowsAffected > 0) {
JOptionPane.showMessageDialog(frame, "Booking Confirmed!");
frame.dispose();
if (homePageRef != null) {
homePageRef.showFrame();
}
} else {
JOptionPane.showMessageDialog(frame, "Booking failed. Please try again.");
}
}
} catch (SQLException e) {
e.printStackTrace();
}
}

```

```

        JOptionPane.showMessageDialog(frame,
            "An error occurred while processing your booking. Please try again.",
            "Error", JOptionPane.ERROR_MESSAGE);
    }
}

```

## Owner Home Page.java

```

import javax.swing.*;
import javax.swing.border.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;

public class OwnerHomePage extends JFrame {
    private static final Color PRIMARY_COLOR = new Color(54, 116, 181); // #3674B5
    private static final Color SECONDARY_COLOR = new Color(87, 143, 202); // #578FCA
    private static final Color ACCENT_COLOR = new Color(161, 227, 249); // #A1E3F9
    private static final Color BACKGROUND_COLOR = new Color(209, 248, 239); // #D1F8EF
    private static final int PADDING = 15;

    private JTextField locationField, modelYearField, seatsField, registrationNumberField, pricePerDayField;
    private JComboBox<String> conditionCombo, typeCombo;
    private JCheckBox insuranceCheck;

    public OwnerHomePage() {
        setTitle("RiderProvider - Vehicle Upload");
        setSize(1000, 750);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
        setLayout(new BorderLayout(0, 10));
        getContentPane().setBackground(BACKGROUND_COLOR);

        // Create and add header panel
        JPanel header = createHeader();
        add(header, BorderLayout.NORTH);

        // Main content area with form and preview
        JPanel mainContent = new JPanel(new BorderLayout(20, 0));
        mainContent.setBackground(BACKGROUND_COLOR);
        mainContent.setBorder(BorderFactory.createEmptyBorder(PADDING, PADDING, PADDING, PADDING));

        // Form panel on the left
        JPanel formPanel = createFormPanel();

        // Preview panel on the right
        JPanel previewPanel = createPreviewPanel();

        // Add panels to main content
        mainContent.add(formPanel, BorderLayout.CENTER);
    }
}

```

```

mainContent.add(previewPanel, BorderLayout.EAST);
add(mainContent, BorderLayout.CENTER);

// Bottom panel with submit button
JPanel bottomPanel = createBottomPanel();
add(bottomPanel, BorderLayout.SOUTH);

setVisible(true);
}

private JPanel createHeader() {
    JPanel header = new JPanel(new BorderLayout());
    header.setBackground(PRIMARY_COLOR);
    header.setBorder(BorderFactory.createEmptyBorder(15, 20, 15, 20));
    header.setPreferredSize(new Dimension(getWidth(), 80));

    // Panel for logo and company name
    JPanel logoPanel = new JPanel(new FlowLayout(FlowLayout.LEFT, 15, 0));
    logoPanel.setOpaque(false);

    // Scale the logo
    ImageIcon originalIcon = new ImageIcon("logo.png");
    Image scaledImage = originalIcon.getImage().getScaledInstance(50, 50, Image.SCALE_SMOOTH);
    ImageIcon scaledIcon = new ImageIcon(scaledImage);
    JLabel logo = new JLabel(scaledIcon);

    // Company name label
    JLabel companyName = new JLabel("RiderProvider");
    companyName.setFont(new Font("Arial", Font.BOLD, 24));
    companyName.setForeground(Color.WHITE);

    logoPanel.add(logo);
    logoPanel.add(companyName);
    header.add(logoPanel, BorderLayout.WEST);

    // Page title
    JLabel pageTitle = new JLabel("Upload Your Vehicle", SwingConstants.CENTER);
    pageTitle.setFont(new Font("Arial", Font.BOLD, 24));
    pageTitle.setForeground(Color.WHITE);
    header.add(pageTitle, BorderLayout.CENTER);

    // Profile button
    JPanel profilePanel = new JPanel(new FlowLayout(FlowLayout.RIGHT));
    profilePanel.setOpaque(false);
    JButton profileButton = new JButton("My Profile");
    profileButton.setFont(new Font("Arial", Font.BOLD, 16));
    profileButton.setBackground(SECONDARY_COLOR);
    profileButton.setForeground(Color.WHITE);
    profileButton.setFocusPainted(false);
    profileButton.setCursor(new Cursor(Cursor.HAND_CURSOR));
    profilePanel.add(profileButton);
    header.add(profilePanel, BorderLayout.EAST);
}

```

```

    return header;
}
private JPanel createFormPanel() {
    JPanel formWrapper = new JPanel(new BorderLayout());
    formWrapper.setBackground(BACKGROUND_COLOR);

    JLabel formTitle = new JLabel("Vehicle Details");
    formTitle.setFont(new Font("Arial", Font.BOLD, 20));
    formTitle.setForeground(PRIMARY_COLOR);
    formTitle.setBorder(BorderFactory.createEmptyBorder(0, 0, 10, 0));
    formWrapper.add(formTitle, BorderLayout.NORTH);

    // Create the form with GridBagLayout for better control
    JPanel form = new JPanel(new GridBagLayout());
    form.setBackground(Color.WHITE);
    form.setBorder(BorderFactory.createCompoundBorder(
        BorderFactory.createLineBorder(SECONDARY_COLOR, 2),
        BorderFactory.createEmptyBorder(20, 20, 20, 20)
    ));

    GridBagConstraints gbc = new GridBagConstraints();
    gbc.fill = GridBagConstraints.HORIZONTAL;
    gbc.insets = new Insets(8, 8, 8, 8);
    gbc.weightx = 0.3;

    // Row 1: Location
    locationField = createStyledTextField();
    addFormRow(form, gbc, "Location:", locationField);

    // Row 2: Car Condition
    String[] conditions = {"Excellent", "Good", "Fair", "Poor"};
    conditionCombo = createStyledComboBox(conditions);
    addFormRow(form, gbc, "Car Condition:", conditionCombo);

    // Row 3: Upload Images
    JButton chooseFilesBtn = createStyledButton("Choose Files");
    addFormRow(form, gbc, "Upload Images:", chooseFilesBtn);

    // Row 4: Insurance
    JPanel insurancePanel = new JPanel(new FlowLayout(FlowLayout.LEFT, 0, 0));
    insurancePanel.setBackground(Color.WHITE);
    insuranceCheck = new JCheckBox("Yes");
    insuranceCheck.setFont(new Font("Arial", Font.PLAIN, 16));
    insuranceCheck.setBackground(Color.WHITE);
    insurancePanel.add(insuranceCheck);
    addFormRow(form, gbc, "Insurance Available:", insurancePanel);

    // Row 5: Model Year
    modelYearField = createStyledTextField();
    addFormRow(form, gbc, "Model Year:", modelYearField);

    // Row 6: Seats
    String[] seats = {"2", "4", "5", "7", "8+"};

```

```
seatsField = createStyledTextField();
addFormRow(form, gbc, "Seats:", seatsField);
```

```
// Row 7: Vehicle Type
```

```
String[] vehicleTypes = {"Car", "Bike", "SUV", "Van"};
typeCombo = createStyledComboBox(vehicleTypes);
addFormRow(form, gbc, "Vehicle Type:", typeCombo);
```

```
// Row 8: Registration Number
```

```
registrationNumberField = createStyledTextField();
addFormRow(form, gbc, "Registration Number:", registrationNumberField);
```

```
// Row 9: Price/Day
```

```
pricePerDayField = createStyledTextField();
addFormRow(form, gbc, "Price/Day:", pricePerDayField);
```

```
formWrapper.add(new JScrollPane(form), BorderLayout.CENTER);
return formWrapper;
```

```
}
```

```
private void addFormRow(JPanel panel, GridBagConstraints gbc, String labelText, Component component) {
```

```
    gbc.gridx = 0;
    gbc.gridy++;
    JLabel label = new JLabel(labelText);
    label.setFont(new Font("Arial", Font.BOLD, 16));
    label.setForeground(PRIMARY_COLOR);
    panel.add(label, gbc);
```

```
    gbc.gridx = 1;
    gbc.weightx = 0.7;
    panel.add(component, gbc);
    gbc.weightx = 0.3;
```

```
}
```

```
private JTextField createStyledTextField() {
```

```
    JTextField field = new JTextField();
    field.setFont(new Font("Arial", Font.PLAIN, 16));
    field.setPreferredSize(new Dimension(200, 35));
    field.setBorder(BorderFactory.createCompoundBorder(
        BorderFactory.createLineBorder(SECONDARY_COLOR),
        BorderFactory.createEmptyBorder(5, 8, 5, 8)
    ));
    return field;
```

```
}
```

```
private JComboBox<String> createStyledComboBox(String[] items) {
```

```
    JComboBox<String> comboBox = new JComboBox<>(items);
    comboBox.setFont(new Font("Arial", Font.PLAIN, 16));
    comboBox.setPreferredSize(new Dimension(200, 35));
    comboBox.setBackground(Color.WHITE);
    comboBox.setForeground(new Color(50, 50, 50));
    return comboBox;
```

```
}
```



```

private JButton createStyledButton(String text) {
    JButton button = new JButton(text);
    button.setFont(new Font("Arial", Font.BOLD, 16));
    button.setPreferredSize(new Dimension(200, 35));
    button.setBackground(ACCENT_COLOR);
    button.setForeground(new Color(50, 50, 50));
    button.setFocusPainted(false);
    button.setCursor(new Cursor(Cursor.HAND_CURSOR));
    return button;
}

private JPanel createPreviewPanel() {
    JPanel previewPanel = new JPanel(new BorderLayout());
    previewPanel.setPreferredSize(new Dimension(300, 500));
    previewPanel.setBackground(Color.WHITE);
    previewPanel.setBorder(BorderFactory.createCompoundBorder(
        BorderFactory.createLineBorder(SECONDARY_COLOR, 2),
        BorderFactory.createEmptyBorder(15, 15, 15, 15)
    ));

    JLabel previewTitle = new JLabel("Vehicle Preview", SwingConstants.CENTER);
    previewTitle.setFont(new Font("Arial", Font.BOLD, 20));
    previewTitle.setForeground(PRIMARY_COLOR);
    previewTitle.setBorder(BorderFactory.createEmptyBorder(0, 0, 10, 0));
    previewPanel.add(previewTitle, BorderLayout.NORTH);

    // Preview image placeholder
    JPanel imagePlaceholder = new JPanel();
    imagePlaceholder.setBackground(new Color(240, 240, 240));
    imagePlaceholder.setPreferredSize(new Dimension(270, 200));
    imagePlaceholder.setBorder(BorderFactory.createLineBorder(SECONDARY_COLOR));

    JLabel placeholderText = new JLabel("Vehicle Image Preview", SwingConstants.CENTER);
    placeholderText.setFont(new Font("Arial", Font.ITALIC, 16));
    placeholderText.setForeground(SECONDARY_COLOR);
    imagePlaceholder.add(placeholderText);

    JPanel centerPanel = new JPanel(new FlowLayout(FlowLayout.CENTER));
    centerPanel.setBackground(Color.WHITE);
    centerPanel.add(imagePlaceholder);
    previewPanel.add(centerPanel, BorderLayout.CENTER);

    // Preview info
    JPanel infoPanel = new JPanel();
    infoPanel.setLayout(new BoxLayout(infoPanel, BoxLayout.Y_AXIS));
    infoPanel.setBackground(Color.WHITE);
    infoPanel.setBorder(BorderFactory.createEmptyBorder(10, 5, 5, 5));

    addPreviewInfoRow(infoPanel, "Vehicle Type:", "Not selected");
    addPreviewInfoRow(infoPanel, "Condition:", "Not specified");
    addPreviewInfoRow(infoPanel, "Location:", "Not specified");
    addPreviewInfoRow(infoPanel, "Price/Day:", "₹0");

```

```

        previewPanel.add(infoPanel, BorderLayout.SOUTH);

        return previewPanel;
    }

    private void addPreviewInfoRow(JPanel panel, String label, String value) {
        JPanel rowPanel = new JPanel(new BorderLayout(10, 0));
        rowPanel.setBackground(Color.WHITE);
        rowPanel.setBorder(BorderFactory.createEmptyBorder(5, 0, 5, 0));

        JLabel labelComponent = new JLabel(label);
        labelComponent.setFont(new Font("Arial", Font.BOLD, 16));
        labelComponent.setForeground(PRIMARY_COLOR);

        JLabel valueComponent = new JLabel(value);
        valueComponent.setFont(new Font("Arial", Font.PLAIN, 16));

        rowPanel.add(labelComponent, BorderLayout.WEST);
        rowPanel.add(valueComponent, BorderLayout.CENTER);

        panel.add(rowPanel);
    }

    private JPanel createBottomPanel() {
        JPanel bottomPanel = new JPanel(new FlowLayout(FlowLayout.CENTER));
        bottomPanel.setBorder(BorderFactory.createEmptyBorder(5, PADDING, PADDING, PADDING));
        bottomPanel.setBackground(BACKGROUND_COLOR);

        JButton submitButton = new JButton("Upload Vehicle");
        submitButton.setPreferredSize(new Dimension(250, 50));
        submitButton.setBackground(PRIMARY_COLOR);
        submitButton.setForeground(Color.WHITE);
        submitButton.setFocusPainted(false);
        submitButton.setBorderPainted(false);
        submitButton.setCursor(new Cursor(Cursor.HAND_CURSOR));
        submitButton.setFont(new Font("Arial", Font.BOLD, 18));

        submitButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                uploadVehicle();
            }
        });

        bottomPanel.add(submitButton);

        return bottomPanel;
    }

    private void uploadVehicle() {
        // Get form values
        String location = locationField.getText();
    }

```

```

String condition = (String) conditionCombo.getSelectedItemAt();
boolean insuranceAvailable = insuranceCheck.isSelected();
int modelYear = Integer.parseInt(modelYearField.getText());
int seats = Integer.parseInt(seatsField.getText());
String vehicleType = (String) typeCombo.getSelectedItemAt();
String registrationNumber = registrationNumberField.getText();
double pricePerDay = Double.parseDouble(pricePerDayField.getText());

// Call a method to save the data to the database
saveVehicleToDatabase(location, condition, insuranceAvailable, modelYear, seats, vehicleType,
registrationNumber, pricePerDay);
}

private void saveVehicleToDatabase(String location, String condition, boolean insuranceAvailable, int
modelYear, int seats, String vehicleType, String registrationNumber, double pricePerDay) {
// Example connection setup, assuming you have a JDBC setup
String url = "jdbc:mysql://localhost:3306/car_rental_db";
String user = "root";
String password = "your_password";

String sql = "INSERT INTO vehicles (location, condition, insurance_available, model_year, seats, vehicle_type,
registration_number, price_per_day) " +
"VALUES (?, ?, ?, ?, ?, ?, ?, ?)";

try (Connection conn = DriverManager.getConnection(url, user, password);
PreparedStatement stmt = conn.prepareStatement(sql)) {

stmt.setString(1, location);
stmt.setString(2, condition);
stmt.setBoolean(3, insuranceAvailable);
stmt.setInt(4, modelYear);
stmt.setInt(5, seats);
stmt.setString(6, vehicleType);
stmt.setString(7, registrationNumber);
stmt.setDouble(8, pricePerDay);

stmt.executeUpdate();
JOptionPane.showMessageDialog(this, "Vehicle uploaded successfully!");
} catch (SQLException e) {
e.printStackTrace();
JOptionPane.showMessageDialog(this, "Error uploading vehicle.");
}
}

public static void main(String[] args) {
new OwnerHomePage();
}
}

```

### User Home Page.java

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

```

```

import java.awt.image.BufferedImage;
import java.awt.font.FontRenderContext;
import javax.swing.border.*;

public class UserHomePage {
    // Your color palette
    private static final Color PRIMARY_COLOR = new Color(54, 116, 181);    // #3674B5
    private static final Color SECONDARY_COLOR = new Color(87, 143, 202); // #578FCA
    private static final Color ACCENT_COLOR = new Color(161, 227, 249);    // #A1E3F9
    private static final Color BACKGROUND_COLOR = new Color(209, 248, 239); // #D1F8EF

    // Additional colors for contrast and highlights
    private static final Color TEXT_COLOR = new Color(33, 33, 33);
    private static final Color BUTTON_HOVER_COLOR = new Color(45, 95, 148);
    private static final Font TITLE_FONT = new Font("Segoe UI", Font.BOLD, 28);
    private static final Font SUBTITLE_FONT = new Font("Segoe UI", Font.BOLD, 18);
    private static final Font REGULAR_FONT = new Font("Segoe UI", Font.PLAIN, 14);

    private JFrame frame;

    public static void main(String[] args) {
        SwingUtilities.invokeLater(UserHomePage::new);
    }

    public UserHomePage() {
        try {
            UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
        } catch (Exception e) {
            e.printStackTrace();
        }

        frame = new JFrame("RiderProvider - Home (Tenant)");
        frame.setSize(900, 700);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setLocationRelativeTo(null);

        // Main panel with background color
        JPanel mainPanel = new JPanel(new BorderLayout(0, 15));
        mainPanel.setBackground(BACKGROUND_COLOR);
        mainPanel.setBorder(BorderFactory.createEmptyBorder(0, 0, 15, 0));

        // Enhanced header panel
        JPanel headerPanel = createHeaderPanel();

        // Improved filter panel
        JPanel filterPanel = createFilterPanel();

        // Container panel for content with padding

```

```

JPanel contentPanel = new JPanel(new BorderLayout(0, 15));
contentPanel.setBackground(BACKGROUND_COLOR);
contentPanel.setBorder(BorderFactory.createEmptyBorder(0, 15, 15, 15));
contentPanel.add(filterPanel, BorderLayout.NORTH);

// Scrollable car list with improved styling
JPanel carListPanel = createCarListPanel();
JScrollPane scrollPane = new JScrollPane(carListPanel);
scrollPane.setBorder(BorderFactory.createEmptyBorder());
scrollPane.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_AS_NEEDED);
scrollPane.getVerticalScrollBar().setUnitIncrement(16);
scrollPane.setBackground(BACKGROUND_COLOR);
contentPanel.add(scrollPane, BorderLayout.CENTER);

// Add components to main panel
mainPanel.add(headerPanel, BorderLayout.NORTH);
mainPanel.add(contentPanel, BorderLayout.CENTER);

frame.setContentPane(mainPanel);
frame.setVisible(true);
}

private JPanel createHeaderPanel() {
    JPanel header = new JPanel(new BorderLayout());
    header.setBackground(PRIMARY_COLOR);
    header.setBorder(BorderFactory.createEmptyBorder(15, 20, 15, 20));
    header.setPreferredSize(new Dimension(900, 80));

    // Panel for logo and company name
    JPanel logoPanel = new JPanel(new FlowLayout(FlowLayout.LEFT, 15, 0));
    logoPanel.setOpaque(false);

    // Create placeholder logo
    ImageIcon originalIcon = new ImageIcon("logo.png");
    Image scaledImage = originalIcon.getImage().getScaledInstance(50, 50, Image.SCALE_SMOOTH);
    ImageIcon scaledIcon = new ImageIcon(scaledImage);
    JLabel logo = new JLabel(scaledIcon);

    // Company name label
    JLabel companyName = new JLabel("RiderProvider");
    companyName.setFont(new Font("Segoe UI", Font.BOLD, 24));
    companyName.setForeground(Color.WHITE);

    logoPanel.add(logo);
    logoPanel.add(companyName);
    header.add(logoPanel, BorderLayout.WEST);

    // Page title

```

```
JLabel pageTitle = new JLabel("Available Rides", SwingConstants.CENTER);
pageTitle.setFont(TITLE_FONT);
pageTitle.setForeground(Color.WHITE);
header.add(pageTitle, BorderLayout.CENTER);
```

```
// Profile button
```

```
JPanel profilePanel = new JPanel(new FlowLayout(FlowLayout.RIGHT));
profilePanel.setOpaque(false);
JButton profileButton = new JButton("My Profile");
profileButton.setFont(new Font("Segoe UI", Font.BOLD, 16));
profileButton.setBackground(SECONDARY_COLOR);
profileButton.setForeground(Color.WHITE);
profileButton.setFocusPainted(false);
profileButton.setCursor(new Cursor(Cursor.HAND_CURSOR));
profilePanel.add(profileButton);
header.add(profilePanel, BorderLayout.EAST);
```

```
return header;
```

```
}
```

```
private JPanel createFilterPanel() {
```

```
JPanel filterPanel = new JPanel();
filterPanel.setLayout(new BoxLayout(filterPanel, BoxLayout.Y_AXIS));
filterPanel.setBackground(Color.WHITE);
filterPanel.setBorder(BorderFactory.createCompoundBorder(
    new LineBorder(ACCENT_COLOR, 1, true),
    BorderFactory.createEmptyBorder(15, 15, 15, 15)
));
```

```
// Filter title
```

```
JLabel filterTitle = new JLabel("Search Rides");
filterTitle.setFont(SUBTITLE_FONT);
filterTitle.setForeground(PRIMARY_COLOR);
filterTitle.setAlignmentX(Component.LEFT_ALIGNMENT);
```

```
// Filter options panel
```

```
JPanel optionsPanel = new JPanel(new GridLayout(2, 4, 15, 10));
optionsPanel.setBackground(Color.WHITE);
optionsPanel.setAlignmentX(Component.LEFT_ALIGNMENT);
optionsPanel.setBorder(BorderFactory.createEmptyBorder(10, 0, 10, 0));
```

```
// Styled combo boxes
```

```
JComboBox<String> locationFilter = createStyledComboBox(new String[] {"All Locations",
"Chennai", "Bangalore", "Hyderabad"});
```

```
JComboBox<String> passengerFilter = createStyledComboBox(new String[] {"Any Passengers", "2
Passengers", "4 Passengers", "6+ Passengers"});
```

```
JComboBox<String> priceFilter = createStyledComboBox(new String[] {"Any Price", "< ₹1000",
"₹1000 - ₹2000", "> ₹2000"});
```

```
JComboBox<String> yearFilter = createStyledComboBox(new String[]{"Any Year", "2020+", "2015-2020", "Before 2015"});
```

```
// Labels with custom styling
```

```
JLabel locLabel = createFilterLabel("Location");  
JLabel passLabel = createFilterLabel("Passengers");  
JLabel priceLabel = createFilterLabel("Price Range");  
JLabel yearLabel = createFilterLabel("Model Year");
```

```
optionsPanel.add(locLabel);  
optionsPanel.add(locationFilter);  
optionsPanel.add(passLabel);  
optionsPanel.add(passengerFilter);  
optionsPanel.add(priceLabel);  
optionsPanel.add(priceFilter);  
optionsPanel.add(yearLabel);  
optionsPanel.add(yearFilter);
```

```
// Button panel
```

```
JPanel buttonPanel = new JPanel(new FlowLayout(FlowLayout.RIGHT));  
buttonPanel.setBackground(Color.WHITE);  
buttonPanel.setAlignmentX(Component.LEFT_ALIGNMENT);
```

```
JButton resetButton = new JButton("Reset");  
resetButton.setFont(REGULAR_FONT);  
resetButton.setForeground(TEXT_COLOR);  
resetButton.setBackground(Color.WHITE);  
resetButton.setBorder(new LineBorder(SECONDARY_COLOR, 1));  
resetButton.setFocusPainted(false);  
resetButton.setCursor(new Cursor(Cursor.HAND_CURSOR));  
resetButton.setPreferredSize(new Dimension(100, 35));
```

```
JButton searchButton = new JButton("Search");  
searchButton.setFont(new Font("Segoe UI", Font.BOLD, 14));  
searchButton.setForeground(Color.WHITE);  
searchButton.setBackground(PRIMARY_COLOR);  
searchButton.setBorderPainted(false);  
searchButton.setFocusPainted(false);  
searchButton.setCursor(new Cursor(Cursor.HAND_CURSOR));  
searchButton.setPreferredSize(new Dimension(120, 35));
```

```
buttonPanel.add(resetButton);  
buttonPanel.add(Box.createRigidArea(new Dimension(10, 0)));  
buttonPanel.add(searchButton);
```

```
filterPanel.add(filterTitle);  
filterPanel.add(Box.createRigidArea(new Dimension(0, 5)));  
filterPanel.add(optionsPanel);
```

```

filterPanel.add(Box.createRigidArea(new Dimension(0, 5)));
filterPanel.add(buttonPanel);

return filterPanel;
}

private JComboBox<String> createStyledComboBox(String[] items) {
    JComboBox<String> comboBox = new JComboBox<>(items);
    comboBox.setFont(REGULAR_FONT);
    comboBox.setBackground(Color.WHITE);
    comboBox.setBorder(BorderFactory.createCompoundBorder(
        new LineBorder(new Color(200, 200, 200), 1),
        BorderFactory.createEmptyBorder(5, 5, 5, 5)
    ));
    ((JComponent) comboBox.getRenderer()).setOpaque(true);
    return comboBox;
}

private JLabel createFilterLabel(String text) {
    JLabel label = new JLabel(text);
    label.setFont(new Font("Segoe UI", Font.BOLD, 13));
    label.setForeground(new Color(80, 80, 80));
    return label;
}

private JPanel createCarListPanel() {
    JPanel carListPanel = new JPanel();
    carListPanel.setLayout(new BoxLayout(carListPanel, BoxLayout.Y_AXIS));
    carListPanel.setBackground(BACKGROUND_COLOR);
    carListPanel.setBorder(BorderFactory.createEmptyBorder(5, 0, 5, 0));

    // Car card models
    String[][] carModels = {
        {"Honda City", "₹1500/day", "2019", "4", "Chennai", "Petrol", "Automatic"},
        {"Toyota Innova", "₹2200/day", "2021", "7", "Bangalore", "Diesel", "Manual"},
        {"Hyundai Verna", "₹1800/day", "2020", "5", "Hyderabad", "Petrol", "Automatic"}
    };

    for (String[] car : carModels) {
        JPanel carCard = createCarCard(car);
        carListPanel.add(carCard);
        carListPanel.add(Box.createRigidArea(new Dimension(0, 15)));
    }

    return carListPanel;
}

private JPanel createCarCard(String[] carDetails) {

```



```
JPanel carCard = new JPanel(new BorderLayout(15, 0));
carCard.setBackground(Color.WHITE);
carCard.setBorder(BorderFactory.createCompoundBorder(
    new LineBorder(new Color(220, 220, 220), 1, true),
    BorderFactory.createEmptyBorder(15, 15, 15, 15)
));

// Car image panel
JPanel imagePanel = new JPanel(new BorderLayout());
imagePanel.setBackground(Color.WHITE);
imagePanel.setPreferredSize(new Dimension(150, 120));

// Create a placeholder for the car image
JLabel carImage = new JLabel();
carImage.setIcon(createCarPlaceholderIcon(150, 120));
carImage.setHorizontalAlignment(JLabel.CENTER);
imagePanel.add(carImage, BorderLayout.CENTER);

// Car info panel
JPanel infoPanel = new JPanel();
infoPanel.setLayout(new BoxLayout(infoPanel, BoxLayout.Y_AXIS));
infoPanel.setBackground(Color.WHITE);

// Car title with model
JLabel carTitle = new JLabel(carDetails[0]);
carTitle.setFont(SUBTITLE_FONT);
carTitle.setForeground(PRIMARY_COLOR);
carTitle.setAlignmentX(Component.LEFT_ALIGNMENT);

// Car details panel
JPanel detailsGrid = new JPanel(new GridLayout(2, 2, 30, 5));
detailsGrid.setBackground(Color.WHITE);
detailsGrid.setAlignmentX(Component.LEFT_ALIGNMENT);

// Car specs with icons
JLabel priceLabel = createDetailLabel("Price: " + carDetails[1]);
JLabel yearLabel = createDetailLabel("Year: " + carDetails[2]);
JLabel passengersLabel = createDetailLabel("Passengers: " + carDetails[3]);
JLabel locationLabel = createDetailLabel("Location: " + carDetails[4]);

detailsGrid.add(priceLabel);
detailsGrid.add(yearLabel);
detailsGrid.add(passengersLabel);
detailsGrid.add(locationLabel);

// Features panel
JPanel featuresPanel = new JPanel(new FlowLayout(FlowLayout.LEFT, 10, 0));
featuresPanel.setBackground(Color.WHITE);
```

```

featuresPanel.setAlignmentX(Component.LEFT_ALIGNMENT);

JLabel fuelType = createFeatureLabel(carDetails[5]);
JLabel transmission = createFeatureLabel(carDetails[6]);
JLabel insurance = createFeatureLabel("Insured");

featuresPanel.add(fuelType);
featuresPanel.add(transmission);
featuresPanel.add(insurance);

// Button panel
JPanel buttonPanel = new JPanel(new FlowLayout(FlowLayout.LEFT, 10, 0));
buttonPanel.setBackground(Color.WHITE);
buttonPanel.setAlignmentX(Component.LEFT_ALIGNMENT);

JButton specButton = createStyledButton("View Details", SECONDARY_COLOR, Color.WHITE);
specButton.addActionListener(e -> {
    JOptionPane.showMessageDialog(
        null,
        "Car Details:\nModel: " + carDetails[0] +
        "\nFuel: " + carDetails[5] +
        "\nTransmission: " + carDetails[6] +
        "\nInsurance: Yes\nCondition: Good\nLocation: " + carDetails[4] +
        "\nModel Year: " + carDetails[2],
        "Specifications",
        JOptionPane.INFORMATION_MESSAGE
    );
});

JButton bookButton = createStyledButton("Book Now", PRIMARY_COLOR, Color.WHITE);
// Modified action listener to open BookingPage
bookButton.addActionListener(e -> {
    try {
        // Hide current frame
        frame.setVisible(false);

        // Open the BookingPage with car details
        new BookingPage(carDetails[0], carDetails[1], carDetails[2], carDetails[4], this);
    } catch (Exception ex) {
        ex.printStackTrace();
        JOptionPane.showMessageDialog(
            null,
            "Error opening booking page: " + ex.getMessage(),
            "Error",
            JOptionPane.ERROR_MESSAGE
        );
    }
});

```

```

buttonPanel.add(specButton);
buttonPanel.add(bookButton);

// Add all components to info panel
infoPanel.add(carTitle);
infoPanel.add(Box.createRigidArea(new Dimension(0, 10)));
infoPanel.add(detailsGrid);
infoPanel.add(Box.createRigidArea(new Dimension(0, 10)));
infoPanel.add(featuresPanel);
infoPanel.add(Box.createRigidArea(new Dimension(0, 15)));
infoPanel.add(buttonPanel);

// Add panels to card
carCard.add(imagePanel, BorderLayout.WEST);
carCard.add(infoPanel, BorderLayout.CENTER);

return carCard;
}

// Method to make the frame visible again
public void showFrame() {
    frame.setVisible(true);
}

private JLabel createDetailLabel(String text) {
    JLabel label = new JLabel(text);
    label.setFont(REGULAR_FONT);
    label.setForeground(TEXT_COLOR);
    return label;
}

private JLabel createFeatureLabel(String text) {
    JLabel label = new JLabel(text);
    label.setFont(new Font("Segoe UI", Font.PLAIN, 12));
    label.setForeground(TEXT_COLOR);
    label.setBackground(ACCENT_COLOR.brighter());
    label.setOpaque(true);
    label.setBorder(BorderFactory.createEmptyBorder(3, 8, 3, 8));
    return label;
}

private JButton createStyledButton(String text, Color bgColor, Color fgColor) {
    JButton button = new JButton(text);
    button.setFont(new Font("Segoe UI", Font.BOLD, 13));
    button.setForeground(fgColor);
    button.setBackground(bgColor);
    button.setBorderPainted(false);

```

```

button.setFocusPainted(false);
button.setCursor(new Cursor(Cursor.HAND_CURSOR));

button.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseEntered(MouseEvent e) {
        button.setBackground(bgColor.darker());
    }

    @Override
    public void mouseExited(MouseEvent e) {
        button.setBackground(bgColor);
    }
});

return button;
}

// Utility methods for placeholder images
private ImageIcon createPlaceholderIcon(int width, int height) {
    BufferedImage img = new BufferedImage(width, height, BufferedImage.TYPE_INT_ARGB);
    Graphics2D g2d = img.createGraphics();

    g2d.setColor(SECONDARY_COLOR);
    g2d.fillRoundRect(0, 0, width, height, 10, 10);

    g2d.setColor(PRIMARY_COLOR);
    g2d.setFont(new Font("Segoe UI", Font.BOLD, 24));
    FontMetrics fm = g2d.getFontMetrics();
    String text = "RP";
    int textWidth = fm.stringWidth(text);
    int textHeight = fm.getHeight();
    g2d.drawString(text, (width - textWidth) / 2, height / 2 + textHeight / 4);

    g2d.dispose();
    return new ImageIcon(img);
}

private ImageIcon createCarPlaceholderIcon(int width, int height) {
    BufferedImage img = new BufferedImage(width, height, BufferedImage.TYPE_INT_ARGB);
    Graphics2D g2d = img.createGraphics();

    g2d.setColor(BACKGROUND_COLOR);
    g2d.fillRect(0, 0, width, height);

    g2d.setColor(SECONDARY_COLOR);
    g2d.setStroke(new BasicStroke(2));
    g2d.drawRoundRect(10, 20, width - 20, height - 40, 20, 20);

```

```

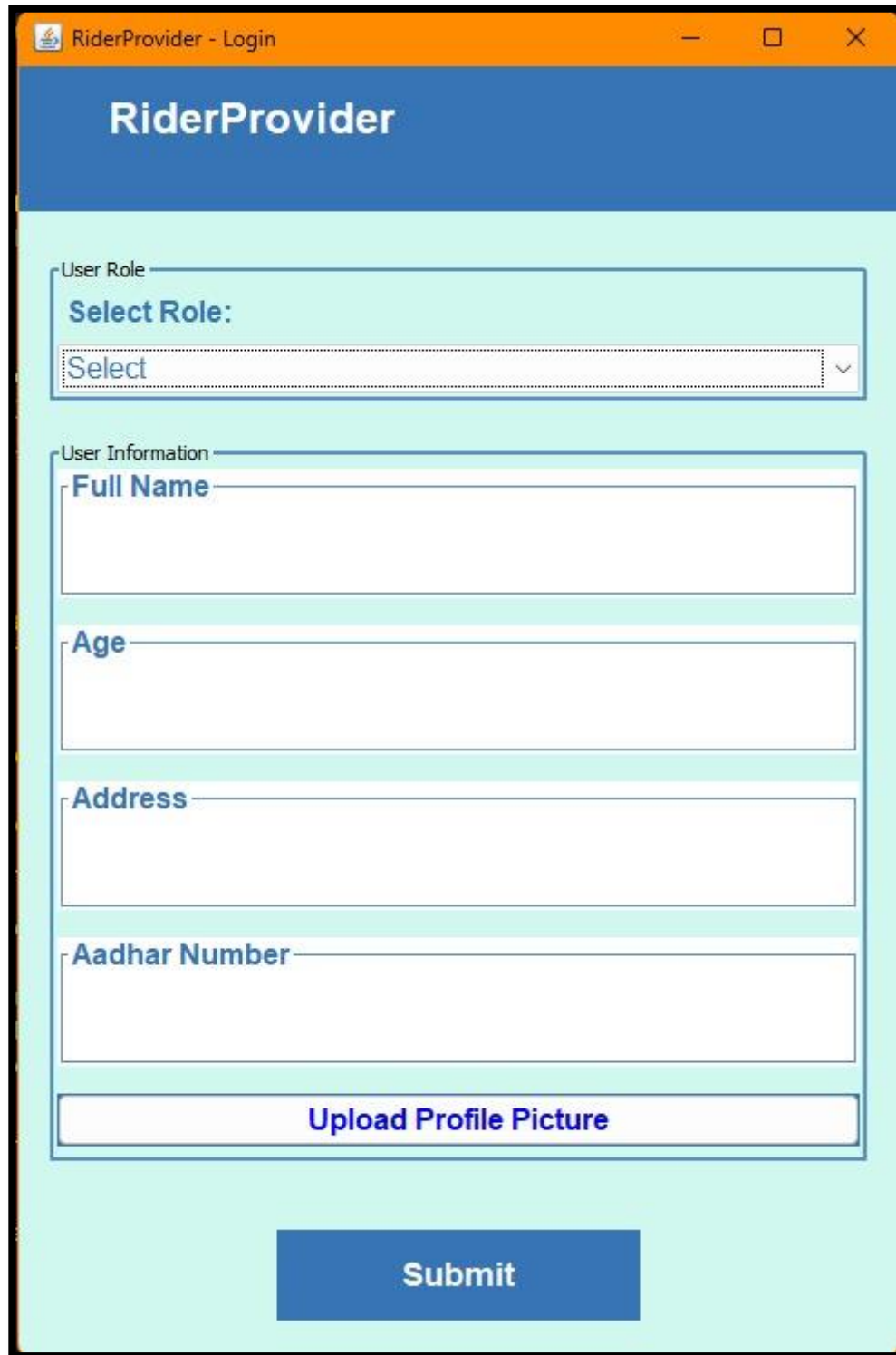
// Draw simple car shape
int carWidth = width - 40;
int carHeight = height - 60;
int startX = 20;
int startY = 30;

g2d.setColor(PRIMARY_COLOR);
// Car body
g2d.fillRoundRect(startX, startY + carHeight/3, carWidth, carHeight/2, 20, 20);
// Car top
g2d.fillRoundRect(startX + carWidth/4, startY, carWidth/2, carHeight/3, 10, 10);
// Wheels
g2d.setColor(new Color(60, 60, 60));
g2d.fillOval(startX + carWidth/5, startY + carHeight*2/3, carWidth/5, carHeight/5);
g2d.fillOval(startX + carWidth*3/5, startY + carHeight*2/3, carWidth/5, carHeight/5);

g2d.dispose();
return new ImageIcon(img);
}
private Image createCircularAvatar(int width, int height) {
    BufferedImage img = new BufferedImage(width, height, BufferedImage.TYPE_INT_ARGB);
    Graphics2D g2d = img.createGraphics();
    g2d.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
RenderingHints.VALUE_ANTIALIAS_ON);
    // Draw circle background
    g2d.setColor(ACCENT_COLOR);
    g2d.fillOval(0, 0, width, height);
    // Draw text
    g2d.setColor(PRIMARY_COLOR);
    g2d.setFont(new Font("Segoe UI", Font.BOLD, 16));
    FontMetrics fm = g2d.getFontMetrics();
    String text = "JD";
    int textWidth = fm.stringWidth(text);
    int textHeight = fm.getHeight();
    g2d.drawString(text, (width - textWidth) / 2, height / 2 + textHeight / 4);
    g2d.dispose();
    return img;
}}

```

## SCREENSHOTS



The screenshot shows a web browser window titled "RiderProvider - Login". The page has a blue header with the text "RiderProvider". Below the header, the form is divided into two main sections: "User Role" and "User Information".

**User Role**

**Select Role:**

Select

**User Information**

**Full Name**

**Age**

**Address**

**Aadhar Number**

**Upload Profile Picture**

**Submit**

RiderProvider - Home (tenant)

RiderProvider

Available Rides

Search Rides

Location

All Locations

Passengers

Any Passengers

Price Range


Any Price

Model Year

Any Year

Reset

Search



Honda City

Price: ₹1500/day

Passengers: 4


Petrol Automatic Insured

View Details

Book Now

Year: 2019

Location: Chennai



Toyota Innova

Price: ₹2200/day

Passengers: 7


Diesel Manual Insured

View Details

Book Now

Year: 2021

Location: Bangalore



Hyundai Verna

Price: ₹1800/day

Passengers: 5

Petrol Automatic Insured

View Details

Book Now

Year: 2020

Location: Hyderabad

RiderProvider - Vehicle Upload

RiderProvider

Upload Your Vehicle

My Profile

Vehicle Details

Location:

Car Condition:

Excellent

Upload Images:

Choose Files

Insurance Available:

☐ Yes

Model Year:

Seats:

Vehicle Type:

Car

Registration Number:

Price/Day:

Vehicle Preview

Vehicle Image Preview

Vehicle Type: Not selected  
Condition: Not specified  
Location: Not specified  
Price/Day: ₹0

Upload Vehicle





## **FUTURE PROSPECTS**

The Vehicle Rental Management System is designed with scalability and future enhancement in mind. The following prospects represent potential expansions to meet evolving business needs and technological advancements:

### **Mobile Application Integration**

Development of companion mobile applications for iOS and Android platforms would extend accessibility, allowing customers to book vehicles and manage reservations directly from their smartphones.

### **GPS Tracking Implementation**

Integration of GPS tracking technology would enable real-time vehicle location monitoring, enhancing fleet management capabilities and providing additional security features.

### **Digital Documentation**

Implementation of digital contracts and e-signature functionality would further streamline the rental process, eliminating paper documentation entirely.

### **Advanced Analytics Dashboard**

Expanding the reporting capabilities with predictive analytics would help optimize pricing strategies, vehicle procurement decisions, and maintenance scheduling.

### **Payment Gateway Expansion**

Supporting multiple payment methods including digital wallets and cryptocurrency options would cater to diverse customer preferences.

### **Customer Loyalty Program**

Development of a points-based loyalty system would encourage repeat business through rewards and special offers for frequent customers.

## CONCLUSION

**RiderProvider** successfully transforms the traditional manual rental process into a streamlined digital experience. By implementing this Java Swing-based application with a robust database infrastructure, rental businesses can achieve significant improvements in operational efficiency and customer service quality.

The system effectively addresses key challenges in the vehicle rental industry through its comprehensive module structure. The user-friendly interface simplifies the booking process for customers while providing administrators with powerful tools to manage inventory, track reservations, and monitor business performance.

Key accomplishments of this system include:

- Elimination of paper-based processes and manual record-keeping
- Reduction in administrative workload through automation
- Enhanced customer experience through self-service capabilities
- Improved inventory management with real-time availability tracking
- Comprehensive reporting for informed business decision-making

The modular architecture ensures the system can adapt to changing business requirements and technological advancements without requiring complete redevelopment. This flexibility represents a significant advantage over rigid legacy systems.

Through the implementation of this Vehicle Rental Management System, businesses can expect to see measurable improvements in customer satisfaction, operational efficiency, and ultimately, profitability. The system stands as a practical solution to real-world challenges in the vehicle rental industry while providing a solid foundation for future technological enhancements.

## References:

1. Oracle. (n.d.). Java Swing Documentation. Oracle. Retrieved from <https://docs.oracle.com/javase/8/docs/technotes/guides/swing/>
2. Oracle. (n.d.). Java SE Documentation. Oracle. Retrieved from <https://docs.oracle.com/javase/8/docs/>
3. GeeksforGeeks. (n.d.). Java Swing Tutorial. Retrieved from <https://www.geeksforgeeks.org/java-swing/>
4. TutorialsPoint. (n.d.). Java Swing Tutorial. Retrieved from <https://www.tutorialspoint.com/swing/index.htm>
5. Javatpoint. (n.d.). Java Swing Tutorial. Retrieved from <https://www.javatpoint.com/java-swing>
6. Bro Code. (2021). Java Swing GUI Full Course – YouTube. Retrieved from <https://www.youtube.com/watch?v=Kmgo00avvEw>
7. CodeAcademy. (n.d.). Java Swing GUI Playlist – YouTube. Retrieved from <https://www.youtube.com/@codeacademy>
8. Oracle. (n.d.). Java Documentation & API Specification. Oracle. Retrieved from <https://docs.oracle.com/en/java/>
9. Codecademy. (n.d.). Learn Java. Retrieved from <https://www.codecademy.com/learn/learn-java>
10. Coursera (Duke University). (n.d.). Java Programming and Software Engineering Fundamentals. Retrieved from <https://www.coursera.org/specializations/java-programming>
11. Udemy. (n.d.). Java Programming Masterclass for Software Developers. Retrieved from <https://www.udemy.com/course/java-the-complete-java-developer-course/>
12. Telusko. (2021). Java Complete Course – YouTube. Retrieved from <https://www.youtube.com/watch?v=8cm1x4bC610>
13. Bro Code. (2021). Java Full Course – YouTube. Retrieved from [https://www.youtube.com/watch?v=xk4\\_1vDrzzo](https://www.youtube.com/watch?v=xk4_1vDrzzo)