

**JAVA PROGRAMMING (BCA 106-2)**

**ESE-III (JAVA PROJECT)**

**“PARK-ZILLA”**

SUBMITTED TO-

**DR. VARUNA GUPTA**

**SCHOOL OF SCIENCES**

**2023-2026**

**TEAM MEMBERS:**

**Manya Bajaj 23215031**

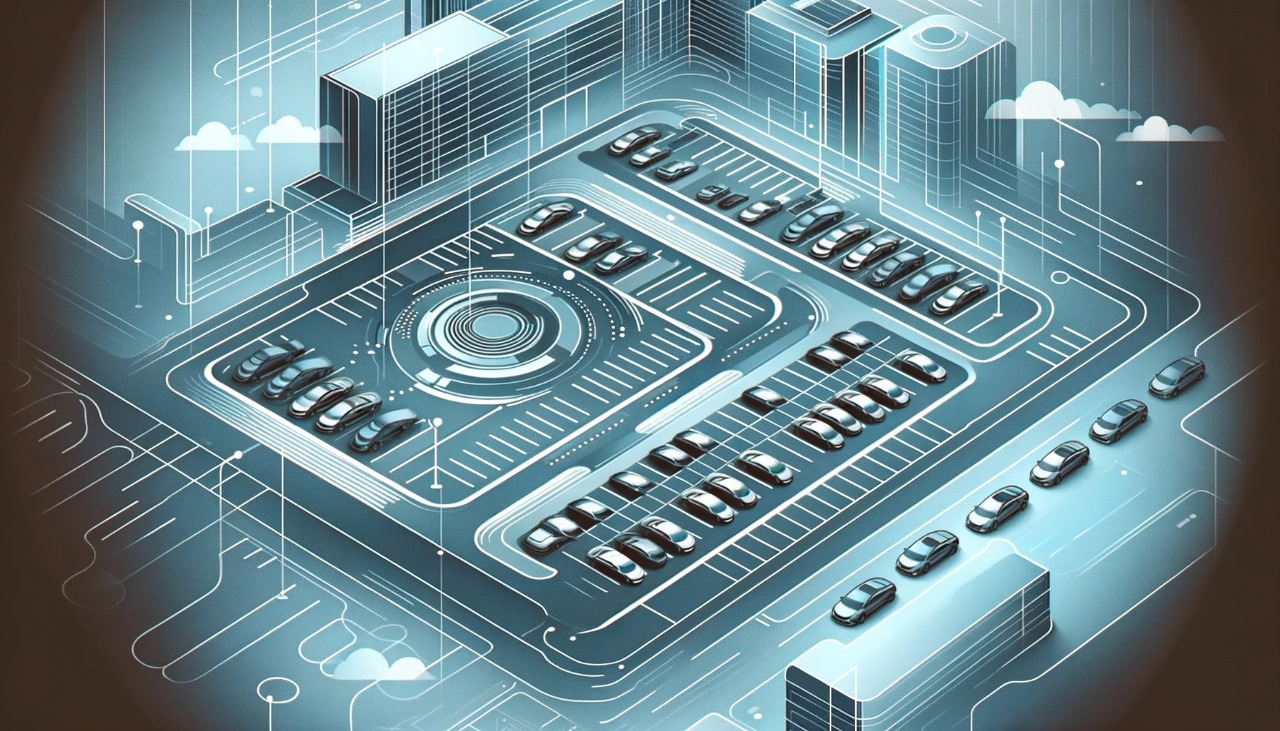
**Mayank Jha 23215032**

**Naman Sharma 23215035**

**Namrata Das 23215036**

**Aneesh Minj 23215072**

**Akshat Batra 23215073**

ABSTRACT:

**Park Zilla is a smart parking system developed to address the challenges of urban parking. The system allows users to pre-book parking slots, eliminating the hassle of finding parking spots on the go. By providing a convenient solution for parking management, Park Zilla aims to enhance the overall parking experience for vehicle owners.**

OBJECTIVES:

**1. On-Demand Parking Booking: Park Zilla enables users to book parking slots on demand, allowing them to secure parking spaces before reaching their destination. This feature helps users avoid the inconvenience of searching for parking spots in busy areas.**

**2. Real-Time Availability: The system provides real-time information on parking slot availability, helping users make informed decisions when booking parking spaces. This feature ensures that users can easily find parking spots without wasting time.**

**3. Convenient Payment Options: Park Zilla offers convenient payment options for booking parking slots, including online payment methods. This feature eliminates the need for cash transactions, making the parking process more convenient for users.**

**4. Integration with Navigation Systems: The system integrates with navigation systems to provide users with directions to the booked parking slot. This feature helps users navigate to their parking spots easily, especially in unfamiliar areas.**

**5. Parking Space Management: Park Zilla allows parking lot operators to manage parking spaces efficiently. Operators can monitor parking occupancy in real time, optimize parking operations, and ensure a smooth flow of vehicles within the parking lot.**

**Overall, Park Zilla aims to simplify the parking process and eliminate parking issues by providing a smart and efficient parking management solution.**

INTRODUCTION:

**In bustling urban environments, the spontaneity of planning a day out often comes with the challenge of finding suitable parking spots upon arrival. This scenario often leads to frustration and delays, detracting from the overall experience. To address this common urban woe, Park Zilla presents a solution—a smart parking system that allows users to pre-book parking slots.**

**Park Zilla's innovative approach aims to revolutionize the parking experience by providing a convenient and efficient way to secure parking spaces in advance. By leveraging technology, Park Zilla not only eliminates the stress of finding parking but also contributes to reducing congestion and improving traffic flow in urban areas.**

**With Park Zilla, users can plan their outings with confidence, knowing that their parking needs are taken care of. Whether it's a spontaneous visit to a popular spot or a planned event, Park Zilla ensures that parking is one less thing to worry about, making urban exploration more enjoyable and hassle-free.**

IMPLEMENTATION:

**HOME PAGE:**

**package project;**

**import javax.swing.\*;**

**import java.awt.\*;**

**import java.awt.event.ActionEvent;**

**import java.awt.event.ActionListener;**

**public class ParkingManagementSystemPage1 extends JFrame implements ActionListener {**

**private JButton customerButton;**

**private JButton adminButton;**

**public ParkingManagementSystemPage1() {**

**setTitle("Parking Management System");**

**setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);**

**setSize(300, 200);**

**setLocationRelativeTo(null);**

**customerButton = new JButton("Customer");**

**adminButton = new JButton("Admin");**

**customerButton.addActionListener(this);**

**adminButton.addActionListener(this);**

**Container contentPane = getContentPane();**

**contentPane.setLayout(new FlowLayout());**

**contentPane.add(customerButton);**

**contentPane.add(adminButton);**

**}**

**@Override**

**public void actionPerformed(ActionEvent e) {**

**if (e.getSource() == customerButton) {**

**LoginSignupPage loginSignupPage = new LoginSignupPage();**

**loginSignupPage.setVisible(true);**

**this.dispose();**

**} else if (e.getSource() == adminButton) {**

**AdminPage adminPage = new AdminPage();**

**adminPage.setVisible(true);**

**this.dispose();**

**}**

**}**

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

**SwingUtilities.invokeLater(() -> {**

**ParkingManagementSystemPage1 app = new ParkingManagementSystemPage1();**

**app.setVisible(true);**

**});**

**}**

**}**

**OUTPUT:**



**LOGIN/SIGNUP PAGE:**

package project;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.\*;

public class LoginSignupPage extends JFrame {

private JFrame frame;

private JTextField txtCustomerId;

private JTextField txtUsername;

private JPasswordField pwdPassword;

private JButton btnLogin;

private JButton btnSignUp;

private Connection connection;

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

LoginSignupPage window = new LoginSignupPage();

window.frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

public LoginSignupPage() {

initialize();

connectToDatabase();

}

private void initialize() {

frame = new JFrame();

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setMinimumSize(new Dimension(400, 200));

frame.getContentPane().setLayout(new BorderLayout());

JPanel loginPanel = new JPanel();

frame.getContentPane().add(loginPanel, BorderLayout.CENTER);

loginPanel.setLayout(new GridLayout(5, 2, 10, 10));

JLabel lblCustomerId = new JLabel("Customer ID:");

loginPanel.add(lblCustomerId);

txtCustomerId = new JTextField();

loginPanel.add(txtCustomerId);

txtCustomerId.setColumns(10);

JLabel lblUsername = new JLabel("Username:");

loginPanel.add(lblUsername);

txtUsername = new JTextField();

loginPanel.add(txtUsername);

txtUsername.setColumns(10);

JLabel lblPassword = new JLabel("Password:");

loginPanel.add(lblPassword);

pwdPassword = new JPasswordField();

loginPanel.add(pwdPassword);

pwdPassword.setColumns(10);

btnLogin = new JButton("Login");

loginPanel.add(btnLogin);

btnSignUp = new JButton("Sign Up");

loginPanel.add(btnSignUp);

btnLogin.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

login();

}

});

btnSignUp.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

signUp();

}

});

frame.pack();

frame.setLocationRelativeTo(null);

frame.setVisible(true);

}

private void connectToDatabase() {

String url = "jdbc:mysql://localhost:3307/parking\_management\_system";

String username = "root";

String password = "root";

try {

connection = DriverManager.getConnection(url, username, password);

System.out.println("Connected to the database.");

} catch (SQLException e) {

e.printStackTrace();

}

}

private void login() {

int customerId = Integer.parseInt(txtCustomerId.getText());

String username = txtUsername.getText();

String password = String.valueOf(pwdPassword.getPassword());

try {

PreparedStatement statement = connection.prepareStatement("SELECT \* FROM login WHERE username = ? AND password = ? AND customer\_id = ?");

statement.setString(1, username);

statement.setString(2, password);

statement.setInt(3, customerId);

ResultSet resultSet = statement.executeQuery();

if (resultSet.next()) {

JOptionPane.showMessageDialog(frame, "Login Successful!");

openUserPage(customerId);

} else {

JOptionPane.showMessageDialog(frame, "Invalid credentials. Please try again.");

}

} catch (SQLException e) {

e.printStackTrace();

}

}

private void signUp() {

int customerId = Integer.parseInt(txtCustomerId.getText());

String username = txtUsername.getText();

String password = String.valueOf(pwdPassword.getPassword());

try {

PreparedStatement checkStatement = connection.prepareStatement("SELECT \* FROM customer WHERE customer\_id = ?");

checkStatement.setInt(1, customerId);

ResultSet resultSet = checkStatement.executeQuery();

if (resultSet.next()) {

insertLoginCredentials(customerId, username, password);

} else {

insertCustomerAndLogin(customerId, username, password);

}

} catch (SQLException e) {

e.printStackTrace();

}

}

private void insertCustomerAndLogin(int customerId, String username, String password) throws SQLException {

PreparedStatement insertCustomerStatement = connection.prepareStatement("INSERT INTO customer (customer\_id) VALUES (?)");

insertCustomerStatement.setInt(1, customerId);

insertCustomerStatement.executeUpdate();

insertLoginCredentials(customerId, username, password);

}

private void insertLoginCredentials(int customerId, String username, String password) throws SQLException {

PreparedStatement insertStatement = connection.prepareStatement("INSERT INTO login (customer\_id, username, password) VALUES (?, ?, ?)");

insertStatement.setInt(1, customerId);

insertStatement.setString(2, username);

insertStatement.setString(3, password);

insertStatement.executeUpdate();

JOptionPane.showMessageDialog(frame, "Sign Up Successful!");

}

private void openUserPage(int customerId) {

SwingUtilities.invokeLater(() -> {

UserPage userPage = new UserPage(customerId);

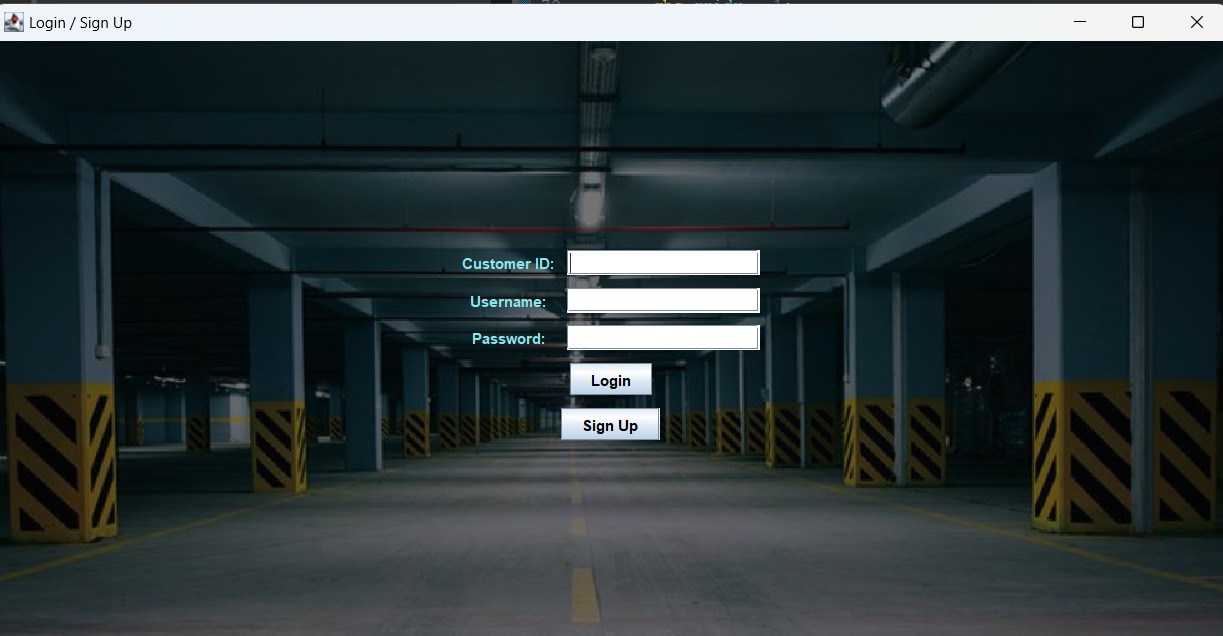
userPage.setVisible(true);

});

}

}

**OUTPUT:**



**USER PAGE:**

**package project;**

**import javax.swing.\*;**

**import java.awt.\*;**

**import java.awt.event.ActionEvent;**

**import java.awt.event.ActionListener;**

**import java.sql.\*;**

**public class UserPage extends JFrame implements ActionListener {**

**private JButton viewBookingsButton;**

**private JButton bookTwoWheelerButton;**

**private JButton bookFourWheelerButton;**

**private JButton logoutButton;**

**private static final String JDBC\_URL = "jdbc:mysql://localhost:3307/parking\_management\_system";**

**private static final String USERNAME = "root";**

**private static final String PASSWORD = "root";**

**private int customerId; // Field to store customer ID**

**public UserPage(int customerId) {**

**this.customerId = customerId;**

**setTitle("User Page");**

**setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);**

**setSize(400, 200);**

**setLocationRelativeTo(null); /**

**viewBookingsButton = new JButton("View My Bookings");**

**bookTwoWheelerButton = new JButton("Book Two Wheeler Slot");**

**bookFourWheelerButton = new JButton("Book Four Wheeler Slot");**

**logoutButton = new JButton("Logout");**

**viewBookingsButton.addActionListener(this);**

**bookTwoWheelerButton.addActionListener(this);**

**bookFourWheelerButton.addActionListener(this);**

**logoutButton.addActionListener(this);**

**Container contentPane = getContentPane();**

**contentPane.setLayout(new GridLayout(4, 1));**

**contentPane.add(viewBookingsButton);**

**contentPane.add(bookTwoWheelerButton);**

**contentPane.add(bookFourWheelerButton);**

**contentPane.add(logoutButton);**

**}**

**@Override**

**public void actionPerformed(ActionEvent e) {**

**if (e.getSource() == viewBookingsButton) {**

**String query = "SELECT \* FROM booked\_slots WHERE customer\_id = ?;";**

**try {**

**Connection connection = DriverManager.getConnection(JDBC\_URL, USERNAME, PASSWORD);**

**PreparedStatement statement = connection.prepareStatement(query);**

**statement.setInt(1, customerId);**

**ResultSet resultSet = statement.executeQuery();**

**StringBuilder resultStringBuilder = new StringBuilder();**

**while (resultSet.next()) {**

**int slotId = resultSet.getInt("slot\_id");**

**int customerId = resultSet.getInt("customer\_id");**

**resultStringBuilder.append("Slot ID: ").append(slotId).append(", Customer ID: ").append(customerId).append("\n");**

**}**

**JOptionPane.showMessageDialog(this, "MySQL Query Result:\n" + resultStringBuilder.toString());**

**resultSet.close();**

**statement.close();**

**connection.close();**

**} catch (SQLException ex) {**

**ex.printStackTrace();**

**JOptionPane.showMessageDialog(this, "Error retrieving booking data: " + ex.getMessage(), "Error", JOptionPane.ERROR\_MESSAGE);**

**}**

**} else if (e.getSource() == bookTwoWheelerButton) {**

**} else if (e.getSource() == bookFourWheelerButton) {**

**} else if (e.getSource() == logoutButton) {**

**}**

**}**

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

**try {**

**Class.forName("com.mysql.cj.jdbc.Driver");**

**} catch (ClassNotFoundException e) {**

**e.printStackTrace();**

**return;**

**}**

**int customerId = 1;**

**SwingUtilities.invokeLater(() -> {**

**UserPage userPage = new UserPage(customerId);**

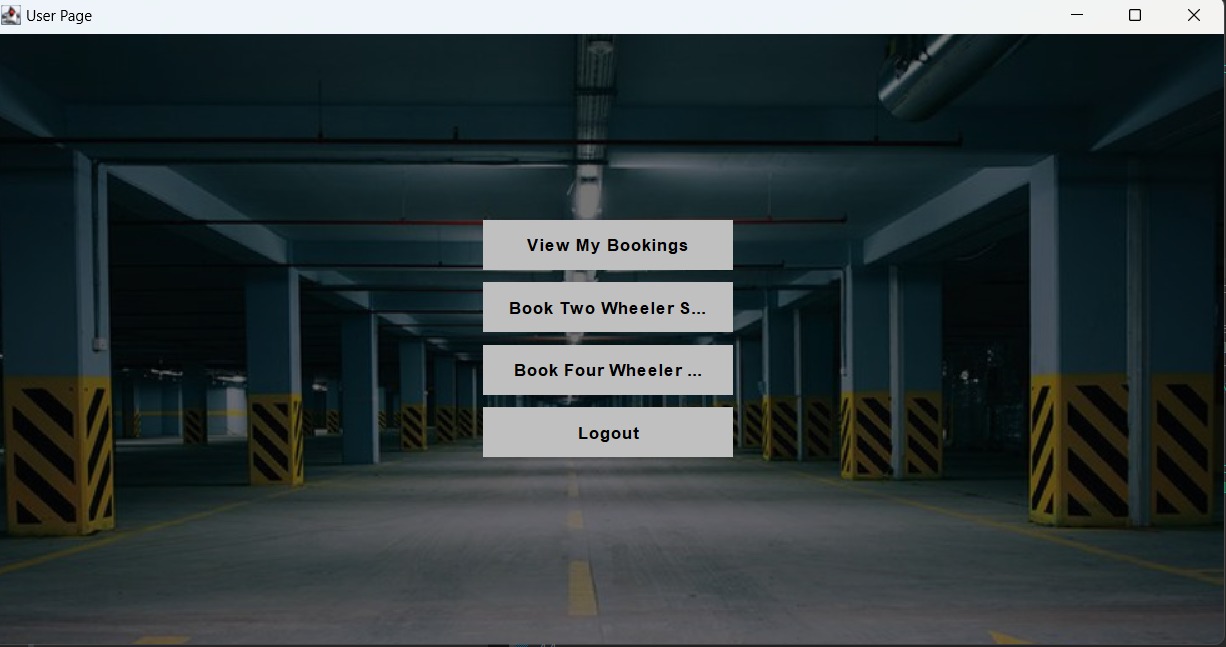
**userPage.setVisible(true);**

**});**

**}**

**}**

**OUTPUT:**



**FOUR WHEELERS:**

**package project;**

**import javax.swing.\*;**

**import javax.swing.table.DefaultTableCellRenderer;**

**import java.awt.\*;**

**import java.awt.event.ActionEvent;**

**import java.awt.event.ActionListener;**

**import java.sql.\*;**

**public class FourWheelers {**

**private static final int MAX\_SLOTS = 20;**

**private static boolean[] slots = new boolean[MAX\_SLOTS];**

**JFrame frame;**

**private JTextArea textArea;**

**private Connection connection;**

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

**EventQueue.invokeLater(new Runnable() {**

**public void run() {**

**try {**

**FourWheelers window = new FourWheelers();**

**window.frame.setVisible(true);**

**} catch (Exception e) {**

**e.printStackTrace();**

**}**

**}**

**});**

**}**

**public FourWheelers() {**

**initialize();**

**connectToDatabase();**

**}**

**private void initialize() {**

**frame = new JFrame();**

**frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);**

**frame.setMinimumSize(new Dimension(800, 600));**

**JTabbedPane tabbedPane = new JTabbedPane(JTabbedPane.TOP);**

**frame.getContentPane().add(tabbedPane, BorderLayout.CENTER);**

**JPanel bookingPanel = new JPanel(new BorderLayout());**

**tabbedPane.addTab("Booking System", null, bookingPanel, null);**

**JPanel buttonPanel = new JPanel();**

**buttonPanel.setLayout(new BoxLayout(buttonPanel, BoxLayout.Y\_AXIS));**

**bookingPanel.add(buttonPanel, BorderLayout.WEST);**

**JButton viewSlotsButton = createStyledButton("View Slots");**

**JButton viewBookingsButton = createStyledButton("View Bookings");**

**JButton paymentButton = createStyledButton("Make Payment");**

**JButton logoutButton = createStyledButton("Logout");**

**buttonPanel.add(Box.createVerticalStrut(20));**

**buttonPanel.add(viewSlotsButton);**

**buttonPanel.add(Box.createVerticalStrut(10));**

**buttonPanel.add(viewBookingsButton);**

**buttonPanel.add(Box.createVerticalStrut(10));**

**buttonPanel.add(paymentButton);**

**buttonPanel.add(Box.createVerticalStrut(10));**

**buttonPanel.add(logoutButton);**

**JPanel outputPanel = new JPanel() {**

**@Override**

**protected void paintComponent(Graphics g) {**

**super.paintComponent(g);**

**ImageIcon imageIcon = new ImageIcon("vehicle.jpg");**

**Image image = imageIcon.getImage();**

**g.drawImage(image, 0, 0, getWidth(), getHeight(), this);**

**}**

**};**

**outputPanel.setLayout(new BorderLayout());**

**bookingPanel.add(outputPanel, BorderLayout.CENTER);**

**textArea = new JTextArea();**

**textArea.setEditable(false);**

**textArea.setBackground(new Color(0, 0, 0, 0));**

**JScrollPane scrollPane = new JScrollPane(textArea);**

**scrollPane.getViewport().setOpaque(false);**

**outputPanel.add(scrollPane, BorderLayout.CENTER);**

**viewSlotsButton.addActionListener(new ActionListener() {**

**public void actionPerformed(ActionEvent e) {**

**viewSlots();**

**}**

**});**

**viewBookingsButton.addActionListener(new ActionListener() {**

**public void actionPerformed(ActionEvent e) {**

**viewBookings();**

**}**

**});**

**paymentButton.addActionListener(new ActionListener() {**

**public void actionPerformed(ActionEvent e) {**

**makePayment();**

**}**

**});**

**logoutButton.addActionListener(new ActionListener() {**

**public void actionPerformed(ActionEvent e) {**

**logout();**

**}**

**});**

**frame.pack();**

**frame.setLocationRelativeTo(null);**

**frame.setVisible(true);**

**}**

**private JButton createStyledButton(String text) {**

**JButton button = new JButton(text);**

**button.setAlignmentX(Component.LEFT\_ALIGNMENT);**

**button.setBackground(Color.LIGHT\_GRAY);**

**button.setForeground(Color.BLACK);**

**button.setBorder(BorderFactory.createEmptyBorder(10, 20, 10, 20));**

**button.setFocusPainted(false);**

**button.setFont(new Font("Arial", Font.BOLD, 14));**

**button.setCursor(new Cursor(Cursor.HAND\_CURSOR));**

**return button;**

**}**

**private void viewBookings() {**

**DefaultListModel<String> model = new DefaultListModel<>();**

**for (int i = 0; i < MAX\_SLOTS; i++) {**

**if (slots[i]) {**

**model.addElement("Slot " + (i + 1) + ": Booked");**

**}**

**}**

**JList<String> list = new JList<>(model);**

**list.setSelectionMode(ListSelectionModel.SINGLE\_SELECTION);**

**list.setVisibleRowCount(5);**

**JOptionPane.showMessageDialog(frame, new JScrollPane(list), "Bookings", JOptionPane.PLAIN\_MESSAGE);**

**}**

**private void viewSlots() {**

**try {**

**Statement statement = connection.createStatement();**

**ResultSet resultSet = statement.executeQuery("SELECT slot\_number FROM booked\_slots\_four\_wheelers");**

**while (resultSet.next()) {**

**int slotNumber = resultSet.getInt("slot\_number");**

**slots[slotNumber - 1] = true; // Mark slot as booked**

**}**

**Object[][] data = new Object[MAX\_SLOTS][2];**

**for (int i = 0; i < MAX\_SLOTS; i++) {**

**data[i][0] = "Slot " + (i + 1);**

**data[i][1] = slots[i] ? "Booked" : "Available";**

**}**

**String[] columnNames = {"Slot Number", "Status"};**

**JTable table = new JTable(data, columnNames) {**

**@Override**

**public boolean isCellEditable(int row, int column) {**

**return false;**

**}**

**};**

**table.setDefaultRenderer(Object.class, new DefaultTableCellRenderer() {**

**@Override**

**public Component getTableCellRendererComponent(JTable table, Object value, boolean isSelected, boolean hasFocus, int row, int column) {**

**Component cellComponent = super.getTableCellRendererComponent(table, value, isSelected, hasFocus, row, column);**

**if (value != null && value.toString().equals("Available")) {**

**cellComponent.setBackground(Color.GREEN);**

**} else {**

**cellComponent.setBackground(Color.RED);**

**}**

**return cellComponent;**

**}**

**});**

**table.setRowHeight(30);**

**table.setFont(new Font("Arial", Font.PLAIN, 14));**

**table.getTableHeader().setFont(new Font("Arial", Font.BOLD, 14));**

**JScrollPane scrollPane = new JScrollPane(table);**

**JOptionPane.showMessageDialog(frame, scrollPane, "Available Slots", JOptionPane.PLAIN\_MESSAGE);**

**} catch (SQLException ex) {**

**ex.printStackTrace();**

**JOptionPane.showMessageDialog(frame, "Failed to retrieve available slots", "Error", JOptionPane.ERROR\_MESSAGE);**

**}**

**}**

**private void makePayment() {**

**}**

**private void logout() {**

**try {**

**if (connection != null && !connection.isClosed()) {**

**connection.close();**

**System.out.println("Database connection closed.");**

**}**

**} catch (SQLException e) {**

**e.printStackTrace();**

**}**

**frame.dispose();**

**}**

**private void connectToDatabase() {**

**String url = "jdbc:mysql://localhost:3307/parking\_management\_system";**

**String username = "root";**

**String password = "root";**

**try {**

**connection = DriverManager.getConnection(url, username, password);**

**System.out.println("Connected to the database.");**

**} catch (SQLException e) {**

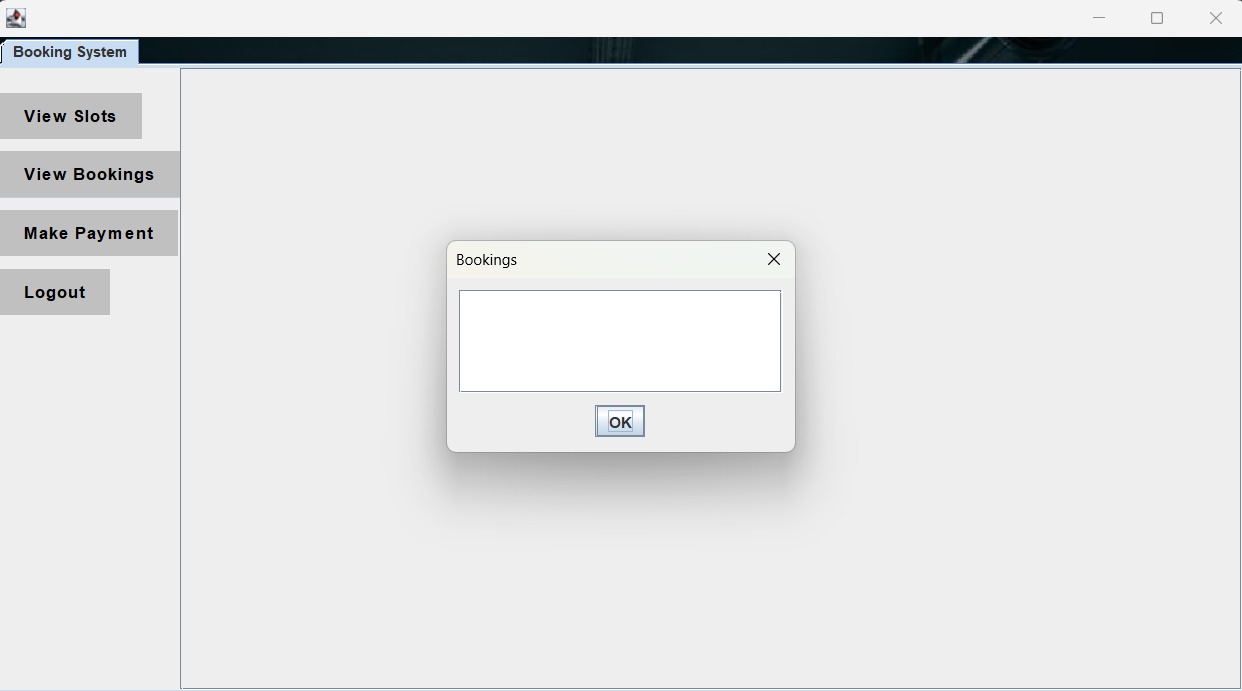
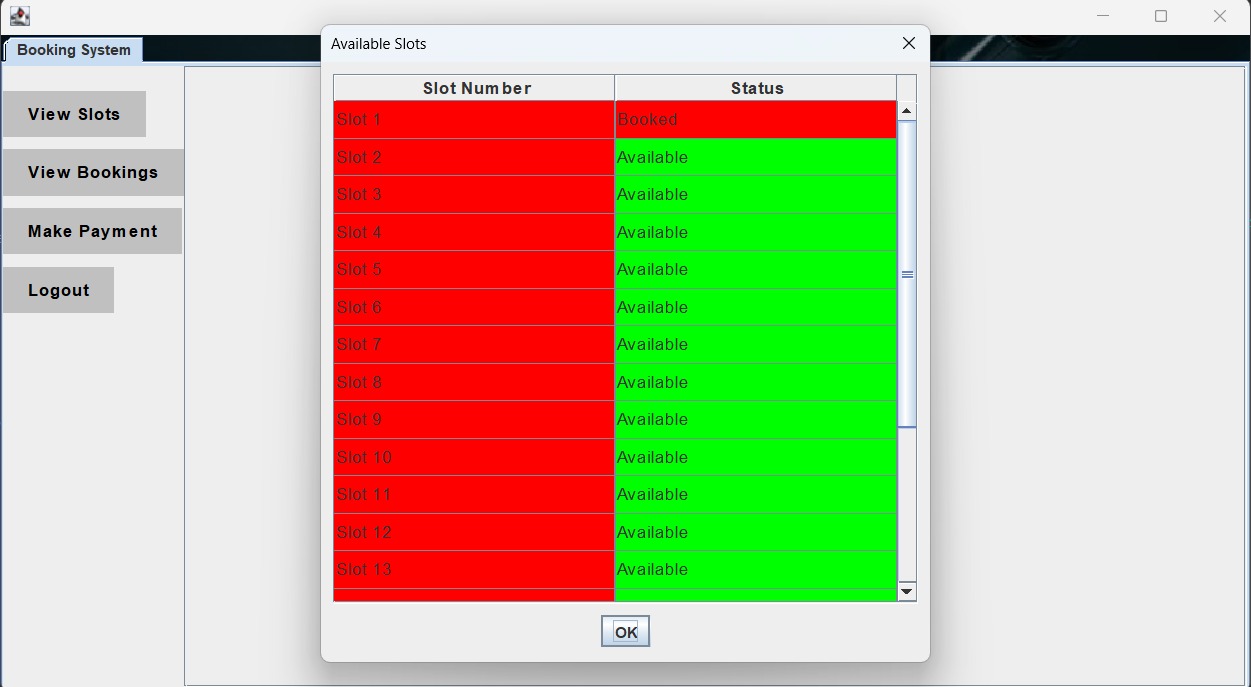
**e.printStackTrace();**

**}**

**}**

**}**

**OUTPUT:**



**TWO WHEELERS:**

**package project;**

**import javax.swing.\*;**

**import javax.swing.table.DefaultTableCellRenderer;**

**import java.awt.\*;**

**import java.awt.event.ActionEvent;**

**import java.awt.event.ActionListener;**

**import java.sql.\*;**

**public class TwoWheelers {**

**private static final int MAX\_SLOTS = 20;**

**private static boolean[] slots = new boolean[MAX\_SLOTS];**

**JFrame frame;**

**private JTextArea textArea;**

**private Connection connection;**

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

**EventQueue.invokeLater(new Runnable() {**

**public void run() {**

**try {**

**TwoWheelers window = new TwoWheelers();**

**window.frame.setVisible(true);**

**} catch (Exception e) {**

**e.printStackTrace();**

**}**

**}**

**});**

**}**

**public TwoWheelers() {**

**initialize();**

**connectToDatabase();**

**}**

**private void initialize() {**

**frame = new JFrame();**

**frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);**

**frame.setMinimumSize(new Dimension(800, 600));**

**JTabbedPane tabbedPane = new JTabbedPane(JTabbedPane.TOP);**

**frame.getContentPane().add(tabbedPane, BorderLayout.CENTER);**

**JPanel bookingPanel = new JPanel(new BorderLayout());**

**tabbedPane.addTab("Booking System", null, bookingPanel, null);**

**JPanel buttonPanel = new JPanel();**

**buttonPanel.setLayout(new BoxLayout(buttonPanel, BoxLayout.Y\_AXIS));**

**bookingPanel.add(buttonPanel, BorderLayout.WEST);**

**JButton viewSlotsButton = createStyledButton("View Slots");**

**JButton viewBookingsButton = createStyledButton("View Bookings");**

**JButton paymentButton = createStyledButton("Make Payment");**

**JButton logoutButton = createStyledButton("Logout");**

**buttonPanel.add(Box.createVerticalStrut(20));**

**buttonPanel.add(viewSlotsButton);**

**buttonPanel.add(Box.createVerticalStrut(10));**

**buttonPanel.add(viewBookingsButton);**

**buttonPanel.add(Box.createVerticalStrut(10));**

**buttonPanel.add(paymentButton);**

**buttonPanel.add(Box.createVerticalStrut(10));**

**buttonPanel.add(logoutButton);**

**JPanel outputPanel = new JPanel() {**

**@Override**

**protected void paintComponent(Graphics g) {**

**super.paintComponent(g);**

**ImageIcon imageIcon = new ImageIcon("vehicle.jpg");**

**Image image = imageIcon.getImage();**

**g.drawImage(image, 0, 0, getWidth(), getHeight(), this);**

**}**

**};**

**outputPanel.setLayout(new BorderLayout());**

**bookingPanel.add(outputPanel, BorderLayout.CENTER);**

**textArea = new JTextArea();**

**textArea.setEditable(false);**

**textArea.setBackground(new Color(0, 0, 0, 0));**

**JScrollPane scrollPane = new JScrollPane(textArea);**

**scrollPane.getViewport().setOpaque(false);**

**outputPanel.add(scrollPane, BorderLayout.CENTER);**

**viewSlotsButton.addActionListener(new ActionListener() {**

**public void actionPerformed(ActionEvent e) {**

**viewSlots();**

**}**

**});**

**viewBookingsButton.addActionListener(new ActionListener() {**

**public void actionPerformed(ActionEvent e) {**

**viewBookings();**

**}**

**});**

**paymentButton.addActionListener(new ActionListener() {**

**public void actionPerformed(ActionEvent e) {**

**makePayment();**

**}**

**});**

**logoutButton.addActionListener(new ActionListener() {**

**public void actionPerformed(ActionEvent e) {**

**logout();**

**}**

**});**

**frame.pack();**

**frame.setLocationRelativeTo(null);**

**frame.setVisible(true);**

**}**

**private JButton createStyledButton(String text) {**

**JButton button = new JButton(text);**

**button.setAlignmentX(Component.LEFT\_ALIGNMENT);**

**button.setBackground(Color.LIGHT\_GRAY);**

**button.setForeground(Color.BLACK);**

**button.setBorder(BorderFactory.createEmptyBorder(10, 20, 10, 20));**

**button.setFocusPainted(false);**

**button.setFont(new Font("Arial", Font.BOLD, 14));**

**button.setCursor(new Cursor(Cursor.HAND\_CURSOR));**

**return button;**

**}**

**private void viewBookings() {**

**DefaultListModel<String> model = new DefaultListModel<>();**

**for (int i = 0; i < MAX\_SLOTS; i++) {**

**if (slots[i]) {**

**model.addElement("Slot " + (i + 1) + ": Booked");**

**}**

**}**

**JList<String> list = new JList<>(model);**

**list.setSelectionMode(ListSelectionModel.SINGLE\_SELECTION);**

**list.setVisibleRowCount(5);**

**JOptionPane.showMessageDialog(frame, new JScrollPane(list), "Bookings", JOptionPane.PLAIN\_MESSAGE);**

**}**

**private void viewSlots() {**

**try {**

**Statement statement = connection.createStatement();**

**ResultSet resultSet = statement.executeQuery("SELECT slot\_number FROM booked\_slots");**

**while (resultSet.next()) {**

**int slotNumber = resultSet.getInt("slot\_number");**

**slots[slotNumber - 1] = true; // Mark slot as booked**

**}**

**Object[][] data = new Object[MAX\_SLOTS][2];**

**for (int i = 0; i < MAX\_SLOTS; i++) {**

**data[i][0] = "Slot " + (i + 1);**

**data[i][1] = slots[i] ? "Booked" : "Available";**

**}**

**String[] columnNames = {"Slot Number", "Status"};**

**JTable table = new JTable(data, columnNames) {**

**@Override**

**public boolean isCellEditable(int row, int column) {**

**return false;**

**}**

**};**

**table.setDefaultRenderer(Object.class, new DefaultTableCellRenderer() {**

**@Override**

**public Component getTableCellRendererComponent(JTable table, Object value, boolean isSelected, boolean hasFocus, int row, int column) {**

**Component cellComponent = super.getTableCellRendererComponent(table, value, isSelected, hasFocus, row, column);**

**if (value != null && value.toString().equals("Available")) {**

**cellComponent.setBackground(Color.GREEN);**

**} else {**

**cellComponent.setBackground(Color.RED);**

**}**

**return cellComponent;**

**}**

**});**

**table.setRowHeight(30);**

**table.setFont(new Font("Arial", Font.PLAIN, 14));**

**table.getTableHeader().setFont(new Font("Arial", Font.BOLD, 14));**

**JScrollPane scrollPane = new JScrollPane(table);**

**JOptionPane.showMessageDialog(frame, scrollPane, "Available Slots", JOptionPane.PLAIN\_MESSAGE);**

**} catch (SQLException ex) {**

**ex.printStackTrace();**

**JOptionPane.showMessageDialog(frame, "Failed to retrieve available slots", "Error", JOptionPane.ERROR\_MESSAGE);**

**}**

**}**

**private void makePayment() {**

**}**

**private void logout() {**

**try {**

**if (connection != null && !connection.isClosed()) {**

**connection.close();**

**System.out.println("Database connection closed.");**

**}**

**} catch (SQLException e) {**

**e.printStackTrace();**

**}**

**frame.dispose();**

**}**

**private void connectToDatabase() {**

**String url = "jdbc:mysql://localhost:3307/parking\_management\_system";**

**String username = "root";**

**String password = "root";**

**try {**

**connection = DriverManager.getConnection(url, username, password);**

**System.out.println("Connected to the database.");**

**} catch (SQLException e) {**

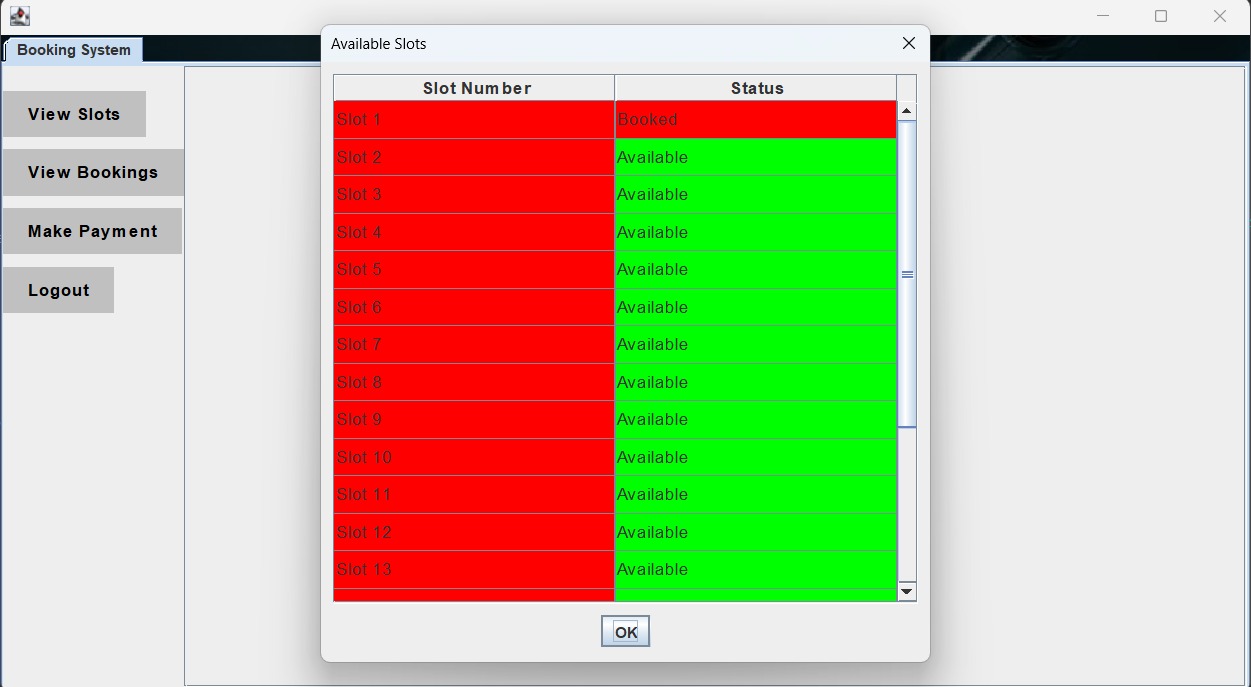
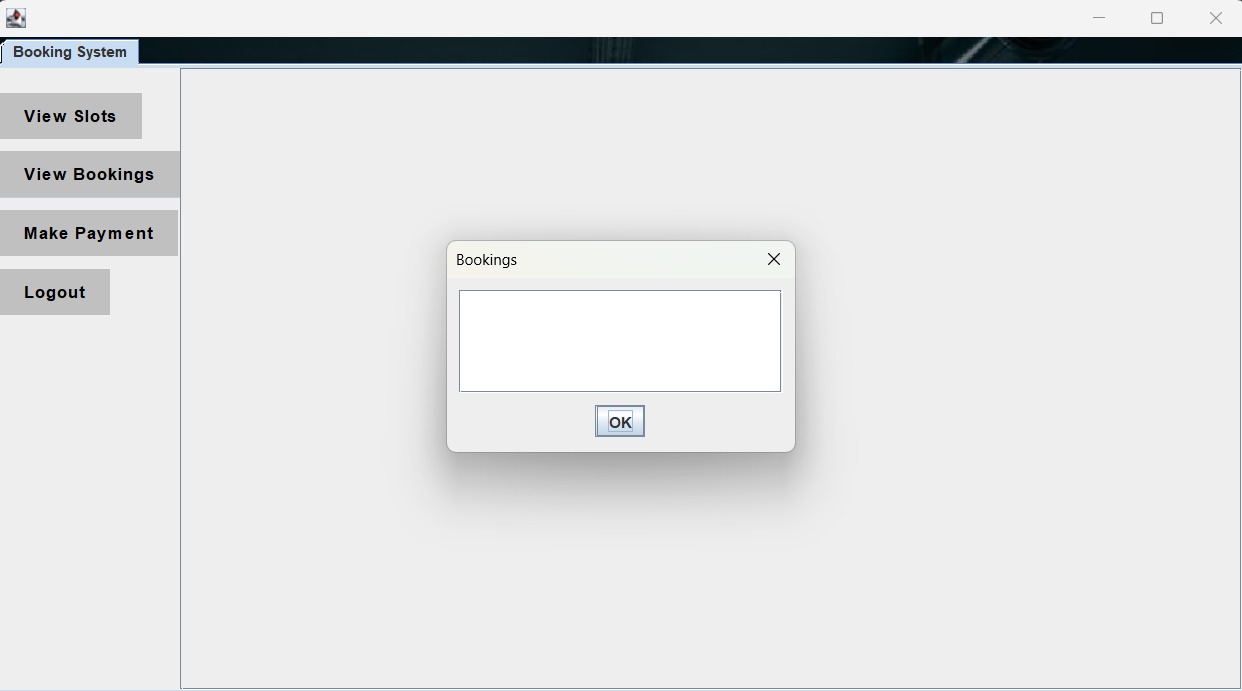
**e.printStackTrace();**

**}**

**}**

**}**

**OUTPUT:**

**Booking Page:**

**package project;**

**import javax.swing.\*;**

**import java.awt.\*;**

**import java.awt.event.ActionEvent;**

**import java.awt.event.ActionListener;**

**import java.sql.\*;**

**public class BookingPage {**

**private JFrame frame;**

**private JTextField slotIdField;**

**private JTextField vehicleTypeField;**

**private JTextField bookingDateTimeField;**

**private JTextField durationField;**

**private Connection connection;**

**private int getLoggedInCustomerId(String username, String password) throws SQLException {**

**int customerId = -1;**

**String query = "SELECT customer\_id FROM login WHERE username = ? AND password = ?";**

**try (PreparedStatement statement = connection.prepareStatement(query)) {**

**statement.setString(1, username);**

**statement.setString(2, password);**

**try (ResultSet resultSet = statement.executeQuery()) {**

**if (resultSet.next()) {**

**customerId = resultSet.getInt("customer\_id");**

**}**

**}**

**}**

**return customerId;**

**}**

**public BookingPage() {**

**initialize();**

**connectToDatabase();**

**}**

**private void initialize() {**

**frame = new JFrame();**

**frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);**

**frame.setMinimumSize(new Dimension(400, 300));**

**frame.setTitle("Booking Page");**

**JPanel panel = new JPanel();**

**panel.setLayout(new GridLayout(6, 2, 10, 10));**

**frame.getContentPane().add(panel, BorderLayout.CENTER);**

**JLabel slotIdLabel = new JLabel("Slot ID:");**

**slotIdField = new JTextField();**

**JLabel vehicleTypeLabel = new JLabel("Vehicle Type:");**

**vehicleTypeField = new JTextField();**

**JLabel bookingDateTimeLabel = new JLabel("Booking Date & Time:");**

**bookingDateTimeField = new JTextField();**

**JLabel durationLabel = new JLabel("Duration:");**

**durationField = new JTextField();**

**JButton bookButton = new JButton("Book");**

**bookButton.addActionListener(new ActionListener() {**

**public void actionPerformed(ActionEvent e) {**

**bookSlot();**

**}**

**});**

**JButton logoutButton = new JButton("Logout");**

**logoutButton.addActionListener(new ActionListener() {**

**public void actionPerformed(ActionEvent e) {**

**logout();**

**}**

**});**

**panel.add(slotIdLabel);**

**panel.add(slotIdField);**

**panel.add(vehicleTypeLabel);**

**panel.add(vehicleTypeField);**

**panel.add(bookingDateTimeLabel);**

**panel.add(bookingDateTimeField);**

**panel.add(durationLabel);**

**panel.add(durationField);**

**panel.add(logoutButton);**

**panel.add(bookButton);**

**frame.pack();**

**frame.setLocationRelativeTo(null);**

**frame.setVisible(true);**

**}**

**private void bookSlot() {**

**try {**

**String username = "";**

**String password = "";**

**int customerId = getLoggedInCustomerId(username, password);**

**int slotId = Integer.parseInt(slotIdField.getText());**

**String vehicleType = vehicleTypeField.getText();**

**String bookingDateTime = bookingDateTimeField.getText();**

**int duration = Integer.parseInt(durationField.getText());**

**PreparedStatement statement = connection.prepareStatement(**

**"INSERT INTO booked\_slots (customer\_id, slot\_number, booking\_time, duration, vehicle\_type) " +**

**"VALUES (?, ?, ?, ?, ?)");**

**statement.setInt(1, customerId);**

**statement.setInt(2, slotId);**

**statement.setString(3, bookingDateTime);**

**statement.setInt(4, duration);**

**statement.setString(5, vehicleType);**

**int rowsAffected = statement.executeUpdate();**

**if (rowsAffected > 0) {**

**JOptionPane.showMessageDialog(frame, "Slot booked successfully!");**

**} else {**

**JOptionPane.showMessageDialog(frame, "Failed to book the slot.", "Error", JOptionPane.ERROR\_MESSAGE);**

**}**

**} catch (NumberFormatException | SQLException ex) {**

**ex.printStackTrace();**

**JOptionPane.showMessageDialog(frame, "Invalid input or database error.", "Error", JOptionPane.ERROR\_MESSAGE);**

**}**

**}**

**private void connectToDatabase() {**

**String url = "jdbc:mysql://localhost:3307/parking\_management\_system";**

**String username = "root";**

**String password = "root";**

**try {**

**connection = DriverManager.getConnection(url, username, password);**

**System.out.println("Connected to the database.");**

**} catch (SQLException e) {**

**e.printStackTrace();**

**}**

**}**

**private void logout() {**

**frame.dispose();**

**new LogoutPage();**

**}**

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

**EventQueue.invokeLater(new Runnable() {**

**public void run() {**

**try {**

**BookingPage window = new BookingPage();**

**} catch (Exception e) {**

**e.printStackTrace();**

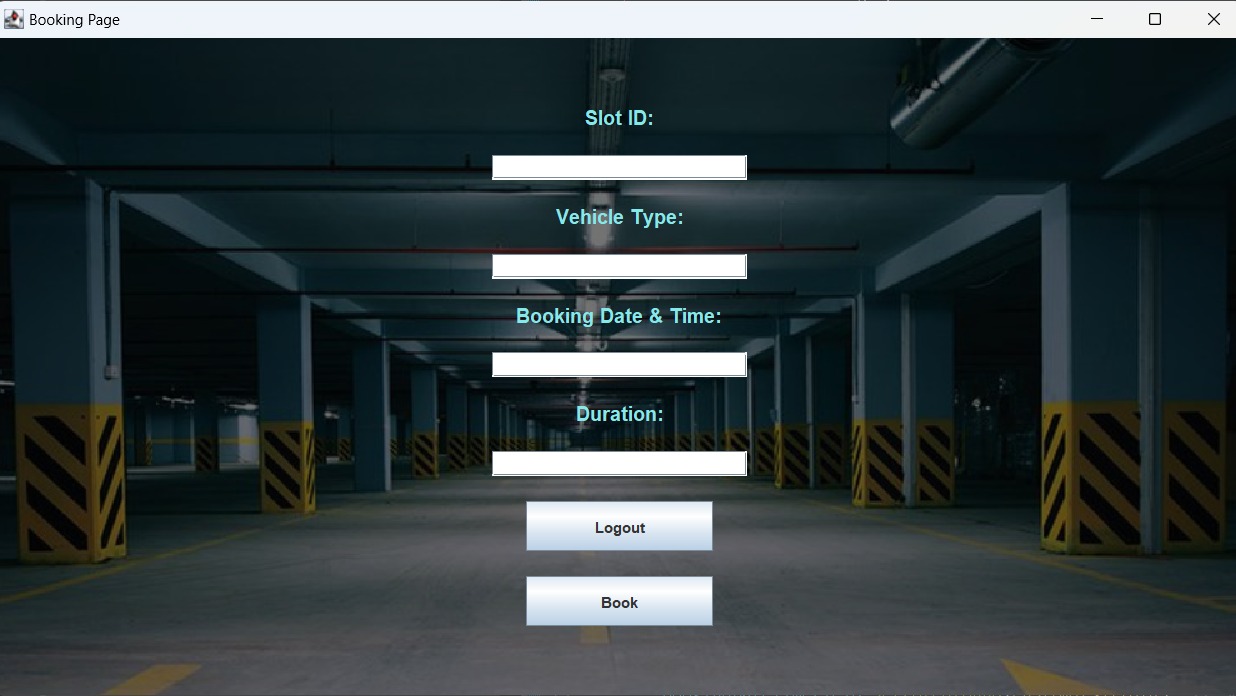
**}**

**}**

**});**

**}**

**}**

**OUTPUT:** 

**Admin Page:**

**package project;**

**import javax.swing.\*;**

**import java.awt.\*;**

**import java.awt.event.ActionEvent;**

**import java.awt.event.ActionListener;**

**import java.sql.\*;**

**public class AdminPage extends JFrame implements ActionListener {**

**private JButton viewBookedSlotsButton;**

**private JButton viewCustomerDetailsButton;**

**private JButton updateCostButton;**

**private JButton logoutButton;**

**public AdminPage() {**

**setTitle("Admin Page");**

**setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);**

**setSize(400, 200);**

**setLocationRelativeTo(null);**

**viewBookedSlotsButton = new JButton("View Booked Slots");**

**viewCustomerDetailsButton = new JButton("View Customer Details");**

**updateCostButton = new JButton("Update Cost");**

**logoutButton = new JButton("Logout");**

**viewBookedSlotsButton.addActionListener(this);**

**viewCustomerDetailsButton.addActionListener(this);**

**updateCostButton.addActionListener(this);**

**logoutButton.addActionListener(this);**

**Container contentPane = getContentPane();**

**contentPane.setLayout(new GridLayout(4, 1));**

**contentPane.add(viewBookedSlotsButton);**

**contentPane.add(viewCustomerDetailsButton);**

**contentPane.add(updateCostButton);**

**contentPane.add(logoutButton);**

**}**

**@Override**

**public void actionPerformed(ActionEvent e) {**

**if (e.getSource() == viewBookedSlotsButton) {**

**viewBookedSlots();**

**} else if (e.getSource() == viewCustomerDetailsButton) {**

**viewCustomerDetails();**

**} else if (e.getSource() == updateCostButton) {**

**updateCost();**

**} else if (e.getSource() == logoutButton) {**

**JOptionPane.showMessageDialog(this, "Logged out");**

**dispose();**

**}**

**}**

**private void viewBookedSlots() {**

**try {**

**Connection connection = DriverManager.getConnection("jdbc:mysql://localhost:3307/parking\_management\_system", "root", "root");**

**PreparedStatement statement = connection.prepareStatement("SELECT \* FROM booked\_slots");**

**ResultSet resultSet = statement.executeQuery();**

**StringBuilder slotsData = new StringBuilder();**

**while (resultSet.next()) {**

**slotsData.append("Slot ID: ").append(resultSet.getInt("slot\_id")).append(", Customer ID: ").append(resultSet.getInt("customer\_id")).append("\n");**

**}**

**JOptionPane.showMessageDialog(this, slotsData.toString(), "Booked Slots", JOptionPane.INFORMATION\_MESSAGE);**

**resultSet.close();**

**statement.close();**

**connection.close();**

**} catch (SQLException ex) {**

**ex.printStackTrace();**

**JOptionPane.showMessageDialog(this, "Error retrieving booked slots data", "Error", JOptionPane.ERROR\_MESSAGE);**

**}**

**}**

**private void viewCustomerDetails() {**

**try {**

**Connection connection = DriverManager.getConnection("jdbc:mysql://localhost:3307/parking\_management\_system", "root", "root");**

**PreparedStatement statement = connection.prepareStatement("SELECT \* FROM customer\_details");**

**ResultSet resultSet = statement.executeQuery();**

**StringBuilder customerData = new StringBuilder();**

**while (resultSet.next()) {**

**customerData.append("Customer ID: ").append(resultSet.getInt("customer\_id")).append(", Name: ").append(resultSet.getString("name")).append("\n");**

**}**

**JOptionPane.showMessageDialog(this, customerData.toString(), "Customer Details", JOptionPane.INFORMATION\_MESSAGE);**

**resultSet.close();**

**statement.close();**

**connection.close();**

**} catch (SQLException ex) {**

**ex.printStackTrace();**

**JOptionPane.showMessageDialog(this, "Error retrieving customer details data", "Error", JOptionPane.ERROR\_MESSAGE);**

**}**

**}**

**private void updateCost() {**

**try {**

**Connection connection = DriverManager.getConnection("jdbc:mysql://localhost:3307/parking\_management\_system", "root", "root");**

**PreparedStatement statement = connection.prepareStatement("UPDATE cost\_table SET cost = ? WHERE id = 1");**

**statement.setDouble(1, 15.00);**

**int rowsAffected = statement.executeUpdate();**

**if (rowsAffected > 0) {**

**JOptionPane.showMessageDialog(this, "Cost updated successfully");**

**} else {**

**JOptionPane.showMessageDialog(this, "Failed to update cost", "Error", JOptionPane.ERROR\_MESSAGE);**

**}**

**statement.close();**

**connection.close();**

**} catch (SQLException ex) {**

**ex.printStackTrace();**

**JOptionPane.showMessageDialog(this, "Error updating cost", "Error", JOptionPane.ERROR\_MESSAGE);**

**}**

**}**

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

**createDatabaseAndTables();**

**SwingUtilities.invokeLater(() -> {**

**AdminPage adminPage = new AdminPage();**

**adminPage.setVisible(true);**

**});**

**}**

**private static void createDatabaseAndTables() {**

**try {**

**Connection connection = DriverManager.getConnection("jdbc:mysql://localhost:3307/", "root", "root");**

**Statement statement = connection.createStatement();**

**statement.executeUpdate("CREATE DATABASE IF NOT EXISTS parking\_management\_system");**

**statement.close();**

**statement = connection.createStatement();**

**statement.execute("USE parking\_management\_system");**

**statement.close();**

**statement = connection.createStatement();**

**statement.executeUpdate("CREATE TABLE IF NOT EXISTS booked\_slots (" +**

**"slot\_id INT AUTO\_INCREMENT PRIMARY KEY," +**

**"customer\_id INT," +**

**"slot\_number INT," +**

**"booking\_time DATETIME," +**

**"duration INT," +**

**"cost DECIMAL(10, 2)" +**

**")");**

**statement.executeUpdate("CREATE TABLE IF NOT EXISTS customer\_details (" +**

**"customer\_id INT AUTO\_INCREMENT PRIMARY KEY," +**

**"name VARCHAR(100)," +**

**"email VARCHAR(100)," +**

**"phone\_number VARCHAR(20)" +**

**")");**

**statement.executeUpdate("CREATE TABLE IF NOT EXISTS cost\_table (" +**

**"id INT AUTO\_INCREMENT PRIMARY KEY," +**

**"cost DECIMAL(10, 2)" +**

**")");**

**statement.executeUpdate("INSERT INTO cost\_table (cost) VALUES (10.00)");**

**statement.close();**

**connection.close();**

**} catch (SQLException e) {**

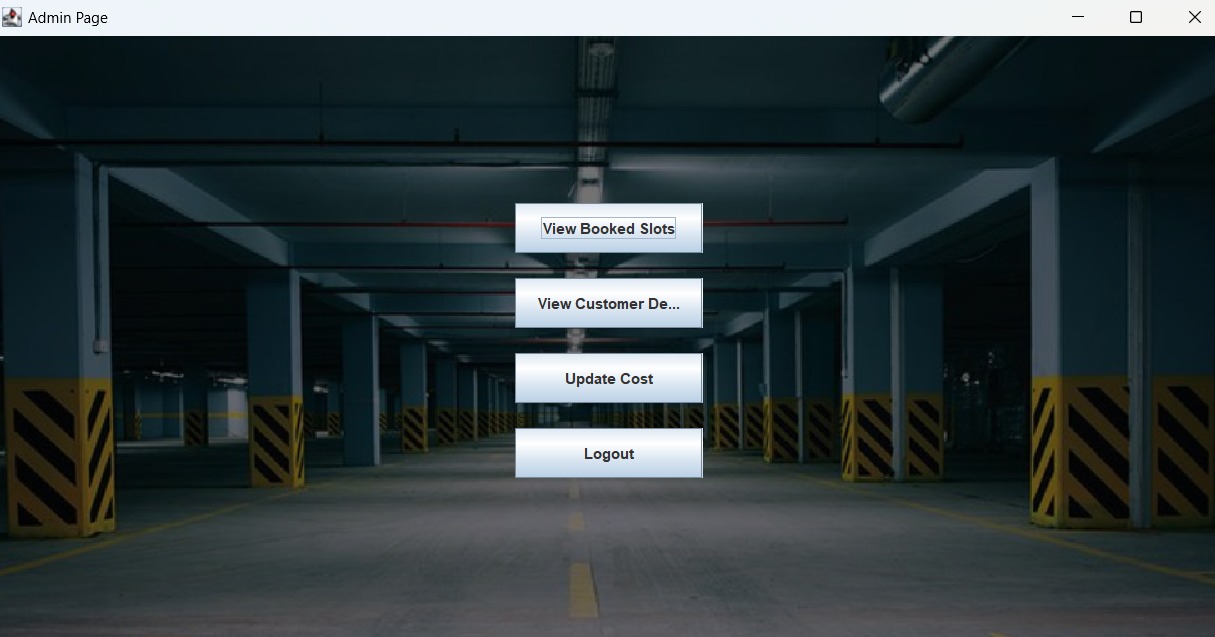
**e.printStackTrace();**

**}**

**}**

**}**

**OUTPUT:**



**Logout Page:**

**package project;**

**import javax.swing.\*;**

**import java.awt.\*;**

**import java.awt.event.ActionEvent;**

**import java.awt.event.ActionListener;**

**public class LogoutPage {**

**private JFrame frame;**

**public LogoutPage() {**

**initialize();**

**}**

**private void initialize() {**

**frame = new JFrame();**

**frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);**

**frame.setMinimumSize(new Dimension(300, 200));**

**frame.setTitle("Logout");**

**JPanel panel = new JPanel();**

**frame.getContentPane().add(panel, BorderLayout.CENTER);**

**JLabel label = new JLabel("You have been logged out.");**

**label.setHorizontalAlignment(SwingConstants.CENTER);**

**panel.add(label);**

**JButton loginButton = new JButton("Login");**

**loginButton.addActionListener(new ActionListener() {**

**public void actionPerformed(ActionEvent e) {**

**frame.dispose();**

**new LoginSignupPage(); /**

**}**

**});**

**panel.add(loginButton);**

**frame.pack();**

**frame.setLocationRelativeTo(null);**

**frame.setVisible(true);**

**}**

**}**

FINDINGS:

**Park-Zilla, a smart parking system, was developed to address the common issue of parking availability at popular destinations. The system allows users to book parking spots in advance, thereby eliminating the last-minute hassle of finding parking. During the course of the project, several key findings emerged:**

**1. Parking Congestion: The research revealed that many popular destinations experience parking congestion due to a lack of organized parking systems.**

**2. User Convenience: Users highly value the convenience of being able to book parking in advance, especially when planning impromptu outings.**

**3. Technology Integration: Integrating technology such as mobile apps and online platforms proved to be effective in streamlining the parking booking process.**

**4. Operational Efficiency: The implementation of ParkZilla led to improved operational efficiency at parking facilities, reducing congestion and enhancing user experience.**

**‘**

CONCLUSION:

**In summary, Park-Zilla offers a potential remedy for the widespread issue of parking availability. Convenience and peace of mind are provided by the system's ability to reserve parking spaces in advance, particularly during busy seasons or peak hours. Parking management and customer satisfaction may improve even more as a result of future system extensions and improvements.**