SARVESHWAR .P

BUSTICKET RESERVATION

JAVA CODE

import javax.swing.\*;

import javax.swing.table.DefaultTableModel;

import java.sql.\*;

public class Main {

static Connection conn;

public static void main(String[] args) {

connectToDatabase();

JFrame frame = new JFrame("Bus Reservation System");

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

JButton userButton = new JButton("User Panel");

adminButton.setBounds(100, 50, 200, 50);

userButton.setBounds(100, 120, 200, 50);

frame.add(adminButton);

frame.add(userButton);

frame.setSize(400, 300);

frame.setLayout(null);

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

frame.setVisible(true);

adminButton.addActionListener(e -> openAdminPanel());

userButton.addActionListener(e -> openUserPanel());

}

// Connect to MySQL Database

private static void connectToDatabase() {

try {

String url = "jdbc:mysql://localhost:3306/bus\_reservation";

String user = "root"; // Replace with your MySQL username

String password = "Shalini@2005"; // Replace with your MySQL password

conn = DriverManager.getConnection(url, user, password);

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

} catch (Exception e) {

e.printStackTrace();

JOptionPane.showMessageDialog(null, "Database connection failed!");

}

}

// Admin Panel

private static void openAdminPanel() {

JFrame adminFrame = new JFrame("Admin Panel");

JButton addBusButton = new JButton("Add Bus");

JButton viewReservationsButton = new JButton("View Reservations");

addBusButton.setBounds(100, 50, 200, 50);

viewReservationsButton.setBounds(100, 120, 200, 50);

adminFrame.add(addBusButton);

adminFrame.add(viewReservationsButton);

adminFrame.setSize(400, 300);

adminFrame.setLayout(null);

adminFrame.setVisible(true);

addBusButton.addActionListener(e -> addBus());

viewReservationsButton.addActionListener(e -> viewReservations());

}

// Add Bus Functionality

private static void addBus() {

JFrame frame = new JFrame("Add Bus");

JTextField routeField = new JTextField();

JTextField timingField = new JTextField();

JTextField fareField = new JTextField();

JTextField seatsField = new JTextField();

JButton addButton = new JButton("Add");

JLabel routeLabel = new JLabel("Route:");

JLabel timingLabel = new JLabel("Timing:");

JLabel fareLabel = new JLabel("Fare:");

JLabel seatsLabel = new JLabel("Seats:");

routeLabel.setBounds(50, 30, 100, 30);

routeField.setBounds(150, 30, 150, 30);

timingLabel.setBounds(50, 70, 100, 30);

timingField.setBounds(150, 70, 150, 30);

fareLabel.setBounds(50, 110, 100, 30);

fareField.setBounds(150, 110, 150, 30);

seatsLabel.setBounds(50, 150, 100, 30);

seatsField.setBounds(150, 150, 150, 30);

addButton.setBounds(150, 200, 100, 30);

frame.add(routeLabel);

frame.add(routeField);

frame.add(timingLabel);

frame.add(timingField);

frame.add(fareLabel);

frame.add(fareField);

frame.add(seatsLabel);

frame.add(seatsField);

frame.add(addButton);

frame.setSize(400, 300);

frame.setLayout(null);

frame.setVisible(true);

addButton.addActionListener(e -> {

try {

String route = routeField.getText();

String timing = timingField.getText();

double fare = Double.parseDouble(fareField.getText());

int seats = Integer.parseInt(seatsField.getText());

PreparedStatement stmt = conn.prepareStatement(

"INSERT INTO buses (route, timing, fare, seats) VALUES (?, ?, ?, ?)");

stmt.setString(1, route);

stmt.setString(2, timing);

stmt.setDouble(3, fare);

stmt.setInt(4, seats);

stmt.executeUpdate();

JOptionPane.showMessageDialog(frame, "Bus added successfully!");

frame.dispose();

} catch (Exception ex) {

ex.printStackTrace();

JOptionPane.showMessageDialog(frame, "Failed to add bus.");

}

});

}

// View Reservations

private static void viewReservations() {

JFrame frame = new JFrame("View Reservations");

JTable table = new JTable();

DefaultTableModel model = new DefaultTableModel();

model.addColumn("Reservation ID");

model.addColumn("User ID");

model.addColumn("Bus ID");

model.addColumn("Booking Date");

model.addColumn("Seats Booked");

try {

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("SELECT \* FROM reservations");

while (rs.next()) {

model.addRow(new Object[]{

rs.getInt("reservation\_id"),

rs.getInt("user\_id"),

rs.getInt("bus\_id"),

rs.getDate("booking\_date"),

rs.getInt("seats\_booked")

});

}

} catch (Exception e) {

e.printStackTrace();

}

table.setModel(model);

JScrollPane sp = new JScrollPane(table);

sp.setBounds(20, 20, 360, 200);

frame.add(sp);

frame.setSize(400, 300);

frame.setLayout(null);

frame.setVisible(true);

}

// User Panel

private static void openUserPanel() {

JFrame userFrame = new JFrame("User Panel");

JButton registerButton = new JButton("Register");

JButton loginButton = new JButton("Login");

JButton makeReservationButton = new JButton("Make Reservation");

registerButton.setBounds(100, 50, 200, 50);

loginButton.setBounds(100, 120, 200, 50);

makeReservationButton.setBounds(100, 190, 200, 50);

userFrame.add(registerButton);

userFrame.add(loginButton);

userFrame.add(makeReservationButton);

userFrame.setSize(400, 300);

userFrame.setLayout(null);

userFrame.setVisible(true);

registerButton.addActionListener(e -> registerUser());

loginButton.addActionListener(e -> loginUser());

makeReservationButton.addActionListener(e -> openReservationPanel(-1)); // Will open the reservation panel without logging in

}

// Register User

private static void registerUser() {

JFrame frame = new JFrame("Register User");

JTextField nameField = new JTextField();

JTextField emailField = new JTextField();

JPasswordField passwordField = new JPasswordField();

JButton registerButton = new JButton("Register");

JLabel nameLabel = new JLabel("Name:");

JLabel emailLabel = new JLabel("Email:");

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

nameLabel.setBounds(50, 30, 100, 30);

nameField.setBounds(150, 30, 150, 30);

emailLabel.setBounds(50, 70, 100, 30);

emailField.setBounds(150, 70, 150, 30);

passwordLabel.setBounds(50, 110, 100, 30);

passwordField.setBounds(150, 110, 150, 30);

registerButton.setBounds(150, 160, 100, 30);

frame.add(nameLabel);

frame.add(nameField);

frame.add(emailLabel);

frame.add(emailField);

frame.add(passwordLabel);

frame.add(passwordField);

frame.add(registerButton);

frame.setSize(400, 300);

frame.setLayout(null);

frame.setVisible(true);

registerButton.addActionListener(e -> {

try {

String name = nameField.getText();

String email = emailField.getText();

String password = new String(passwordField.getPassword());

PreparedStatement stmt = conn.prepareStatement(

"INSERT INTO users (name, email, password) VALUES (?, ?, ?)");

stmt.setString(1, name);

stmt.setString(2, email);

stmt.setString(3, password);

stmt.executeUpdate();

JOptionPane.showMessageDialog(frame, "Registration successful!");

frame.dispose();

} catch (Exception ex) {

ex.printStackTrace();

JOptionPane.showMessageDialog(frame, "Failed to register.");

}

});

}

// User Login

private static void loginUser() {

JFrame frame = new JFrame("User Login");

JTextField emailField = new JTextField();

JPasswordField passwordField = new JPasswordField();

JButton loginButton = new JButton("Login");

JLabel emailLabel = new JLabel("Email:");

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

emailLabel.setBounds(50, 50, 100, 30);

emailField.setBounds(150, 50, 150, 30);

passwordLabel.setBounds(50, 90, 100, 30);

passwordField.setBounds(150, 90, 150, 30);

loginButton.setBounds(150, 140, 100, 30);

frame.add(emailLabel);

frame.add(emailField);

frame.add(passwordLabel);

frame.add(passwordField);

frame.add(loginButton);

frame.setSize(400, 300);

frame.setLayout(null);

frame.setVisible(true);

loginButton.addActionListener(e -> {

try {

String email = emailField.getText();

String password = new String(passwordField.getPassword());

PreparedStatement stmt = conn.prepareStatement(

"SELECT \* FROM users WHERE email = ? AND password = ?");

stmt.setString(1, email);

stmt.setString(2, password);

ResultSet rs = stmt.executeQuery();

if (rs.next()) {

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

frame.dispose();

openReservationPanel(rs.getInt("user\_id"));

} else {

JOptionPane.showMessageDialog(frame, "Invalid credentials.");

}

} catch (Exception ex) {

ex.printStackTrace();

}

});

}

// Reservation Panel

private static void openReservationPanel(int userId) {

if (userId == -1) {

JOptionPane.showMessageDialog(null, "Please log in to make a reservation.");

return;

}

JFrame frame = new JFrame("Make Reservation");

JComboBox<String> busComboBox = new JComboBox<>();

JTextField seatsField = new JTextField();

JButton reserveButton = new JButton("Reserve");

JLabel busLabel = new JLabel("Select Bus:");

JLabel seatsLabel = new JLabel("Seats:");

busLabel.setBounds(50, 30, 100, 30);

busComboBox.setBounds(150, 30, 150, 30);

seatsLabel.setBounds(50, 70, 100, 30);

seatsField.setBounds(150, 70, 150, 30);

reserveButton.setBounds(150, 110, 100, 30);

frame.add(busLabel);

frame.add(busComboBox);

frame.add(seatsLabel);

frame.add(seatsField);

frame.add(reserveButton);

// Load bus list into combo box

try {

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("SELECT bus\_id, route FROM buses");

while (rs.next()) {

busComboBox.addItem(rs.getInt("bus\_id") + ": " + rs.getString("route"));

}

} catch (SQLException e) {

e.printStackTrace();

}

frame.setSize(400, 200);

frame.setLayout(null);

frame.setVisible(true);

reserveButton.addActionListener(e -> {

try {

int busId = Integer.parseInt(busComboBox.getSelectedItem().toString().split(":")[0]);

int seats = Integer.parseInt(seatsField.getText());

PreparedStatement stmt = conn.prepareStatement(

"INSERT INTO reservations (user\_id, bus\_id, seats\_booked, booking\_date) VALUES (?, ?, ?, NOW())");

stmt.setInt(1, userId);

stmt.setInt(2, busId);

stmt.setInt(3, seats);

stmt.executeUpdate();

JOptionPane.showMessageDialog(frame, "Reservation successful!");

frame.dispose();

} catch (Exception ex) {

ex.printStackTrace();

JOptionPane.showMessageDialog(frame, "Failed to reserve.");

}

});

}

}

JDBC CONNECTIVITY

-- Create the database

CREATE DATABASE bus\_reservation;

-- Use the database

USE bus\_reservation;

-- Create users table

CREATE TABLE users (

user\_id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(50),

email VARCHAR(50) UNIQUE,

password VARCHAR(50)

);

-- Create buses table

CREATE TABLE buses (

bus\_id INT AUTO\_INCREMENT PRIMARY KEY,

bus\_number VARCHAR(20),

route VARCHAR(100),

timing VARCHAR(50),

fare DOUBLE,

seats INT

);

-- Create reservations table

CREATE TABLE reservations (

reservation\_id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT,

bus\_id INT,

booking\_date DATE,

seats\_booked INT,

FOREIGN KEY (user\_id) REFERENCES users(user\_id),

FOREIGN KEY (bus\_id) REFERENCES buses(bus\_id)

);