**SQL:**

CREATE DATABASE BusTicketSystem;

USE BusTicketSystem;

CREATE TABLE Admins (

admin\_id INT PRIMARY KEY AUTO\_INCREMENT,

username VARCHAR(50) NOT NULL UNIQUE,

password VARCHAR(50) NOT NULL

);

CREATE TABLE Passengers (

passenger\_id INT PRIMARY KEY AUTO\_INCREMENT,

username VARCHAR(50) NOT NULL UNIQUE,

password VARCHAR(50) NOT NULL

);

CREATE TABLE Buses (

bus\_id INT PRIMARY KEY AUTO\_INCREMENT,

route VARCHAR(100) NOT NULL,

capacity INT NOT NULL,

departure\_time TIME NOT NULL

);

CREATE TABLE Tickets (

ticket\_id INT PRIMARY KEY AUTO\_INCREMENT,

bus\_id INT,

passenger\_id INT,

seat\_number INT,

FOREIGN KEY (bus\_id) REFERENCES Buses(bus\_id),

FOREIGN KEY (passenger\_id) REFERENCES Passengers(passenger\_id)

);

**JAVA:**

import java.sql.\*;

import java.util.Scanner;

public class BusTicketManagementSystem {

private static final String DB\_URL = "jdbc:mysql://localhost:3306/BusTicketSystem";

private static final String DB\_USER = "root";

private static final String DB\_PASSWORD = "harish2809"; // Replace with your DB password

private Connection connection;

private Scanner scanner;

public BusTicketManagementSystem() {

try {

connection = DriverManager.getConnection(DB\_URL, DB\_USER, DB\_PASSWORD);

scanner = new Scanner(System.in);

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

} catch (SQLException e) {

e.printStackTrace();

}

}

// Admin Login

private boolean adminLogin() {

System.out.print("Enter Admin Username: ");

String username = scanner.next();

System.out.print("Enter Admin Password: ");

String password = scanner.next();

try {

PreparedStatement stmt = connection.prepareStatement("SELECT \* FROM Admins WHERE username = ? AND password = ?");

stmt.setString(1, username);

stmt.setString(2, password);

ResultSet rs = stmt.executeQuery();

return rs.next();

} catch (SQLException e) {

e.printStackTrace();

}

return false;

}

// Passenger Login or Registration

private int passengerLoginOrRegister() {

System.out.print("Enter 1 to Register, 2 to Login: ");

int choice = scanner.nextInt();

if (choice == 1) {

System.out.print("Enter New Username: ");

String username = scanner.next();

System.out.print("Enter Password: ");

String password = scanner.next();

try {

PreparedStatement stmt = connection.prepareStatement("INSERT INTO Passengers(username, password) VALUES (?, ?)", Statement.RETURN\_GENERATED\_KEYS);

stmt.setString(1, username);

stmt.setString(2, password);

stmt.executeUpdate();

ResultSet keys = stmt.getGeneratedKeys();

if (keys.next()) {

System.out.println("Registration successful!");

return keys.getInt(1);

}

} catch (SQLException e) {

e.printStackTrace();

}

} else if (choice == 2) {

System.out.print("Enter Username: ");

String username = scanner.next();

System.out.print("Enter Password: ");

String password = scanner.next();

try {

PreparedStatement stmt = connection.prepareStatement("SELECT passenger\_id FROM Passengers WHERE username = ? AND password = ?");

stmt.setString(1, username);

stmt.setString(2, password);

ResultSet rs = stmt.executeQuery();

if (rs.next()) {

System.out.println("Login successful!");

return rs.getInt("passenger\_id");

}

} catch (SQLException e) {

e.printStackTrace();

}

}

System.out.println("Login/Registration failed.");

return -1;

}

// Admin Options

private void adminOptions() {

while (true) {

System.out.println("\nAdmin Options:\n1. Add Bus\n2. Delete Bus\n3. View All Buses\n4. Logout");

int choice = scanner.nextInt();

switch (choice) {

case 1:

addBus();

break;

case 2:

deleteBus();

break;

case 3:

viewAllBuses();

break;

case 4:

return;

default:

System.out.println("Invalid choice.");

}

}

}

// Passenger Options

private void passengerOptions(int passengerId) {

while (true) {

System.out.println("\nPassenger Options:\n1. View Available Buses\n2. Book Ticket\n3. Logout");

int choice = scanner.nextInt();

switch (choice) {

case 1:

viewAllBuses();

break;

case 2:

bookTicket(passengerId);

break;

case 3:

return;

default:

System.out.println("Invalid choice.");

}

}

}

// Add a new bus (Admin)

private void addBus() {

System.out.print("Enter Route: ");

String route = scanner.next();

System.out.print("Enter Capacity: ");

int capacity = scanner.nextInt();

System.out.print("Enter Departure Time (HH:MM:SS): ");

String time = scanner.next();

try {

PreparedStatement stmt = connection.prepareStatement("INSERT INTO Buses(route, capacity, departure\_time) VALUES (?, ?, ?)");

stmt.setString(1, route);

stmt.setInt(2, capacity);

stmt.setTime(3, Time.valueOf(time));

stmt.executeUpdate();

System.out.println("Bus added successfully.");

} catch (SQLException e) {

e.printStackTrace();

}

}

// Delete a bus (Admin)

private void deleteBus() {

System.out.print("Enter Bus ID to delete: ");

int busId = scanner.nextInt();

try {

PreparedStatement stmt = connection.prepareStatement("DELETE FROM Buses WHERE bus\_id = ?");

stmt.setInt(1, busId);

stmt.executeUpdate();

System.out.println("Bus deleted successfully.");

} catch (SQLException e) {

e.printStackTrace();

}

}

// View all buses

private void viewAllBuses() {

try {

Statement stmt = connection.createStatement();

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

System.out.println("Available Buses:");

while (rs.next()) {

System.out.println("Bus ID: " + rs.getInt("bus\_id") + ", Route: " + rs.getString("route") +

", Capacity: " + rs.getInt("capacity") + ", Departure: " + rs.getTime("departure\_time"));

}

} catch (SQLException e) {

e.printStackTrace();

}

}

// Book a ticket (Passenger)

private void bookTicket(int passengerId) {

System.out.print("Enter Bus ID to book: ");

int busId = scanner.nextInt();

System.out.print("Enter Seat Number: ");

int seatNumber = scanner.nextInt();

try {

PreparedStatement stmt = connection.prepareStatement("INSERT INTO Tickets(bus\_id, passenger\_id, seat\_number) VALUES (?, ?, ?)");

stmt.setInt(1, busId);

stmt.setInt(2, passengerId);

stmt.setInt(3, seatNumber);

stmt.executeUpdate();

System.out.println("Ticket booked successfully.");

} catch (SQLException e) {

e.printStackTrace();

}

}

// Main Menu

public void mainMenu() {

while (true) {

System.out.println("\nMain Menu:\n1. Admin Login\n2. Passenger Login/Register\n3. Exit");

int choice = scanner.nextInt();

switch (choice) {

case 1:

if (adminLogin()) {

adminOptions();

} else {

System.out.println("Invalid admin credentials.");

}

break;

case 2:

int passengerId = passengerLoginOrRegister();

if (passengerId != -1) {

passengerOptions(passengerId);

}

break;

case 3:

System.out.println("Exiting...");

return;

default:

System.out.println("Invalid choice.");

}

}

}

public static void main(String[] args) {

BusTicketManagementSystem system = new BusTicketManagementSystem();

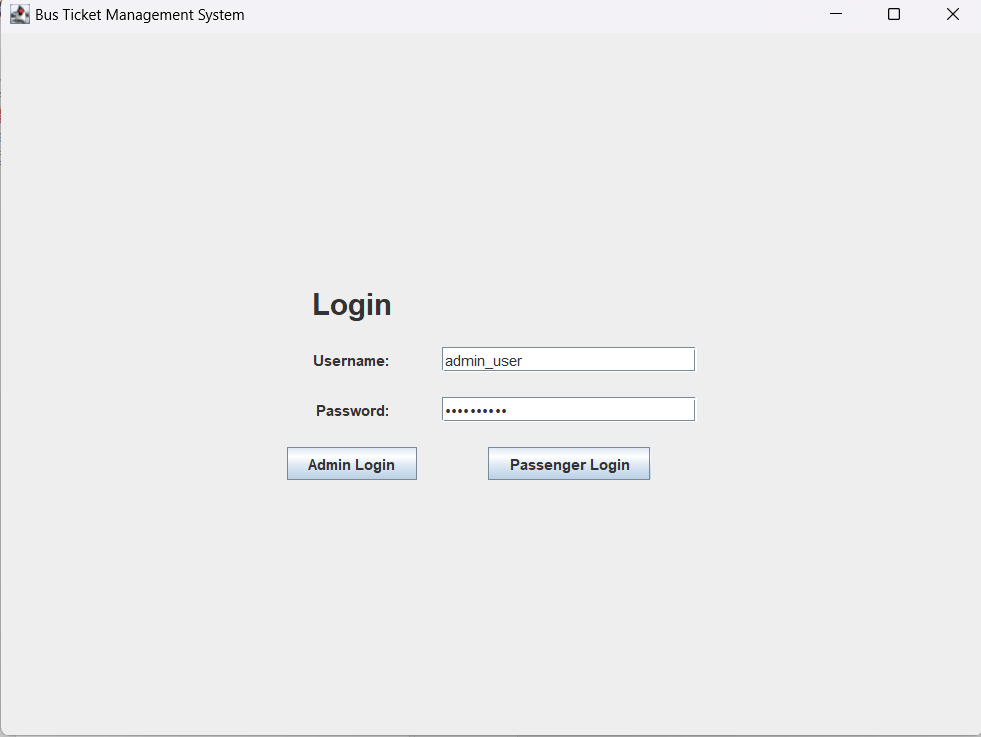
system.mainMenu();

}

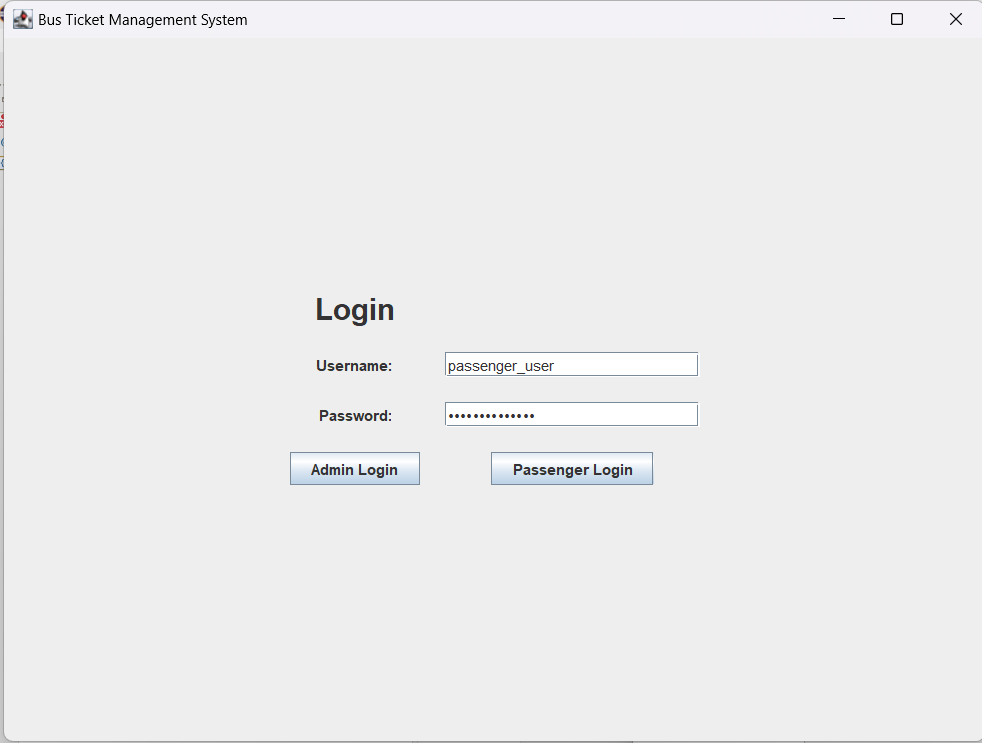
}

**OUTPUT:**

**Admin:**

****

**Passenger:**

****