NAME: SNEHAL SUR

PROJECT TITLE: BUS BOOKING SYSTEM

Abstract:

The project aims to develop a user-friendly and efficient online platform that allows users to search for available buses, check schedules, select seats, and make bookings seamlessly. Administrator, meanwhile, are equipped with tools to manage buses, routes, schedules, reservations and user accounts effortlessly. The Bus Booking System Project introduces a comprehensive solution for streamlining the process of bus ticket reservations, catering to the needs of both passengers and administrator.

Key Features:

* **Admin Module:** Grants administrators the ability to manage buses, routes, bus stops and schedule details.
* **User Management:** Supports user registration, login and view profile details.
* **Bus Search:** Enable users to find bus rides based on various criteria such as date, destination and departure location.
* **Booking Management:**Allows users to book, view and reservations.

Project Goals:

* To create a user-friendly and efficient online platform for users to book a Bus.
* Implement search functionality that allows users to find Bus rides based on various criteria such as source, destination and departure date.
* Allow users to create and manage their accounts where they can track their bookings.

Technologies Used:

* Web Application Framework: ASP.NET MVC 5 (.NET Framework 4.7.2)
* Database: SQL Server 2019
* Front-End: HTML, CSS, Bootstrap
* Back-End: C#

The choice of ASP.NET MVC allows for a separation of concerns, making the system more maintainable and testable. SQL Server provides a robust and scalable database management system capable of handling complex queries.

Database Design

Database Description:

The database for Bus Booking System consists of seven tables. They are:

1. Users

2. BusInfos

3. BusStops

4. Routes

5. Schedules

6. Bookings

7. Payments

Table Description:

* **Users Table:** Contains user information, such as users name, email address and password.
* **BusInfos Table**: Contains details about buses operating withing the system, such as bus name and total seats.
* **BusStops Table**: Stores information about bus stops such as bus stop name, city where the bus stop is located.
* **Routes Table**: Stores all the relevant route information, including route number, source and destination bus stop.
* **Schedules Table:**Contains scheduling information for routes such as scheduled date, departure and arrival time, price of the flight.
* **Bookings Table:** Records of user bookings, including details like booking date, number of passengers.
* **Payments Table:**Details of payments of bookings such as payment date, card type, card number, CVV, amount

Entity-Relationship (E.R.) Diagram:

A screenshot of a computer

Description automatically generated

Data-Flow Diagrams

DFD Level 0:

A diagram of a bus ticket system

Description automatically generated

DFD Level 1:

A diagram of a bus terminal

Description automatically generated

Application Design

Models:

Models represent the application data structures and are designed to mirror the entities in the database. They include:

1. **User Model**: Represents the system users, both customers and administrators.
2. **BusInfo Model**: Encapsulates the details about the buses.
3. **BusStop Model**: Contains information about the bus stops.
4. **Route Model**: Details about routes operated by the buses.
5. **Schedule Model**: Scheduling information for bus trips, including departure and arrival times.
6. **Booking Model**: Records of user booking on scheduled trips.
7. **Payment Model**: Details of payments made for bookings.

Controllers:

Controllers acts as the intermediary between the Model and View, processing all the business logic and user inputs.

* **Handle Requests**: Process user inputs from the view, such as search queries and booking requests.
* **Update Model**: Based on the admin’s actions, it updates the data in the model, like adding, modifying, deleting bus info, bus stop, route, and schedule information. Users can add a new booking and cancel booking.
* **Update View**: After processing the input and updating the model, it selects the appropriate view to display the response or result back to the user.

View:

The view component displays the application user interface and presenting data to the user. In the context, it would involve:

* **Search Results**: Showing available bus trips based on queries.
* **Booking Forms**: Interfaces for entering number of passengers and payment details.
* **User Dashboard**: User’s personalized pages to view their profiles and booking history.
* **Admin Dashboard**: Admin’s personalized pages to manage bus trip details.

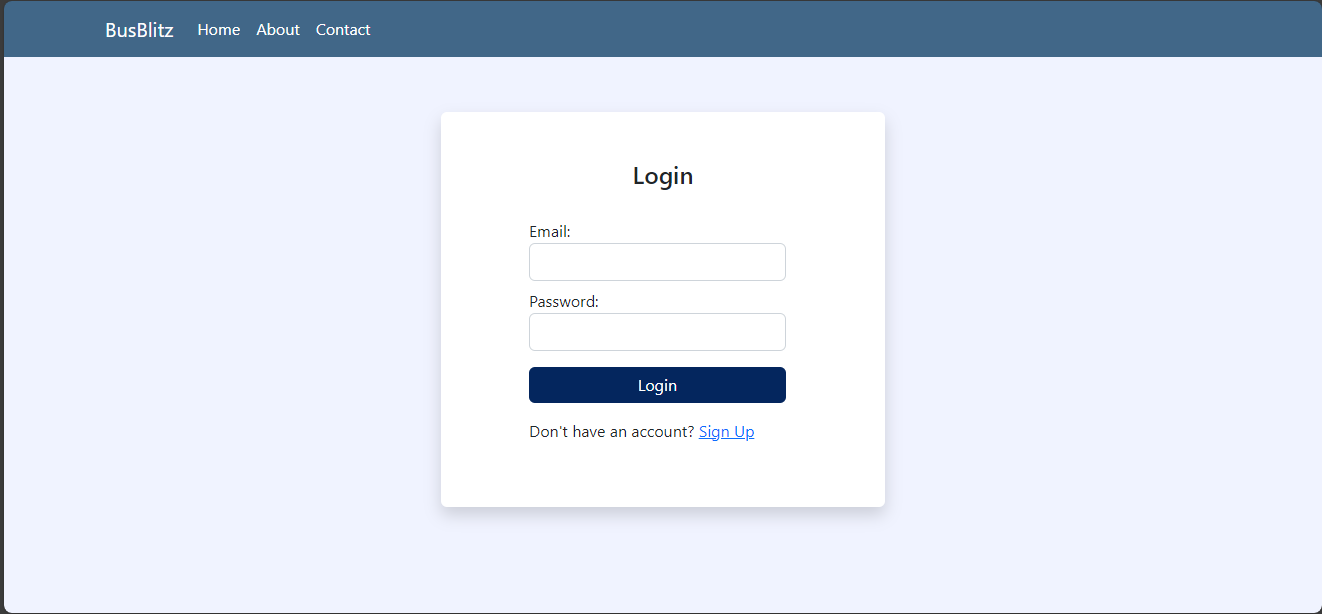
User Interface

Home Page:

A bus driving down the road

Description automatically generated

Login Page:



Admin Home Page:

A screenshot of a computer

Description automatically generated

User Home Page:

A screenshot of a computer

Description automatically generated

Bus Schedules (CRUD Operations):

A screenshot of a computer

Description automatically generated

Future Enhancements:

In future, we will use API to add payment gateways for real-time payments after booking. Some important credentials for users such as passwords will be encrypted before passing to database. It will enhance the security and prevent attackers from stealing user credentials. We will also want to add different roles which will have different functionalities aside from admin and user role which can perform different tasks. We will also make this web application as responsive as possible and add some more security features in future.