សកលវិទ្យាល័យភូមិន្ទនីតិសាស្ត្រ និងវិទ្យាសាស្ត្រសេដ្ឋកិច្ច

ROYAL UNIVERSITY OF LAW AND ECONOMIC

កិច្ចការស្រាវជ្រាវស្ដីអំពីប្រធានបទ

BOOK ME BUS SYSTEM

បរិញ្ញាបត្រ ៖ ព័ត៌មានវិទ្យា

លោកគ្រូ ៖ មាស ភារម្យ

មុខវិជ្ជា ៖ DOT NET WEB PROGRAMMING

ថ្នាក់ ៖ IT3C01

រៀបរៀងដោយសាមណនិស្សិត និងនិស្សិត ក្រុមទី៤

១. សាមណនិស្សិត ៖ ហែ អេម ២. និស្សិត ៖ ទ្រី ប៊ុនហេង

៣. និស្សិត ៖ ម៉ន ធា ៤. និស្សិត ៖ អុល បញ្ញា

[Click To Our Github Repo](#)

ព.ស. ២៥៦៩ គ.ស. ២០២៤

Content

9. Summary

1.1. Overview

1.2. Future Directions

Introduction

1.1. Objective
1.2. Scope

1.1. Objective

The "Book Me Bus System" is a web-based application designed to streamline the bus ticket booking process. It aims to provide users with a simple and efficient platform for searching bus routes, selecting seats, and making reservations. The system is built to enhance user convenience and operational efficiency by automating manual processes and integrating modern technology.

2.1. Scope

The system encompasses several key functionalities:

- User Registration and Authentication: Secure user account creation and login.
- Route Management: Allows users to search for and view available bus routes.
- Seat Selection: Enables users to select available seats for their chosen routes.
- Booking Management: Users can view, modify, and cancel their bookings.
- Payment Processing: Integrates with payment gateways to handle financial transactions securely.

Background

2.1. Context

2.2. Existing Solutions

2.1.   Context

Traditional bus ticket booking methods often involve manual processes, which can be time-consuming and error-prone. These systems may lack real-time information, making it difficult for users to book tickets efficiently. The "Book Me Bus System" addresses these issues by offering a fully automated and accessible solution that simplifies the booking process.

2.2.   Existing Solutions

Current market solutions include online booking platforms and mobile apps. However, they may:

- Lack User-Friendly Interfaces: Some systems can be difficult to navigate.
- Have Limited Management Tools: Administrative tasks might not be integrated effectively.
- Miss Advanced Features: Features such as real-time tracking and dynamic pricing are often absent.

The "Book Me Bus System" seeks to provide a comprehensive solution with an emphasis on user experience and administrative efficiency.

Challenges and Opportunities

3.1. Challenges

3.2. Opportunities

3.1. Challenges

1. Scalability: As the number of users and bookings grows, the system must efficiently handle increased load without performance degradation.

2. User Experience: Designing an interface that is intuitive and accessible to a diverse user base.

3. Security: Protecting sensitive user information and payment details from potential threats.

3.2. Opportunities

1. Integration: Potential to integrate with other travel services, such as hotel reservations or ride-sharing options.

2. Advanced Features: Adding functionalities like real-time bus tracking, dynamic pricing based on demand, and personalized offers.

3. Data Analytics: Utilizing booking data to analyze trends, optimize routes, and tailor marketing strategies.

Functionalities and Features

4.1. Core Features

4.2. Additional Features

4.1. Core Functionalities

1. User Registration and Authentication:
   - Registration Form: Collects user details including name, email, and password.
   - Login: Allows users to access their accounts securely.
   - Password Recovery: Provides a way for users to reset their passwords if forgotten.

2. Route Search and Management:
   - Search Functionality: Users can search for routes by entering details such as origin, destination, and date.
   - Route Details: Displays information about available routes, including bus numbers, departure, and arrival times.

3. Seat Selection:
   - Interactive Map: Users can view an interactive seat map that shows available and booked seats.
   - Seat Reservation: Allows users to select their preferred seats from the available options.

4. Booking Management:
   - View Bookings: Users can view their booking history and details.
   - Modify Bookings: Provides options to change or cancel existing bookings.

5. Payment Processing:
   - Payment Gateway Integration: Facilitates secure transactions through integration with payment processors.
   - Transaction Confirmation: Sends immediate confirmation of successful payments and booking.

4.2. Additional Features

1. Email Confirmation: Automated email notifications for registration, booking confirmations, and other relevant updates.

2. User Dashboard: A personalized dashboard where users can manage their bookings, update personal details, and view their booking history.

3. Admin Panel: A backend interface for administrators to manage routes, seats, users, and view system analytics.
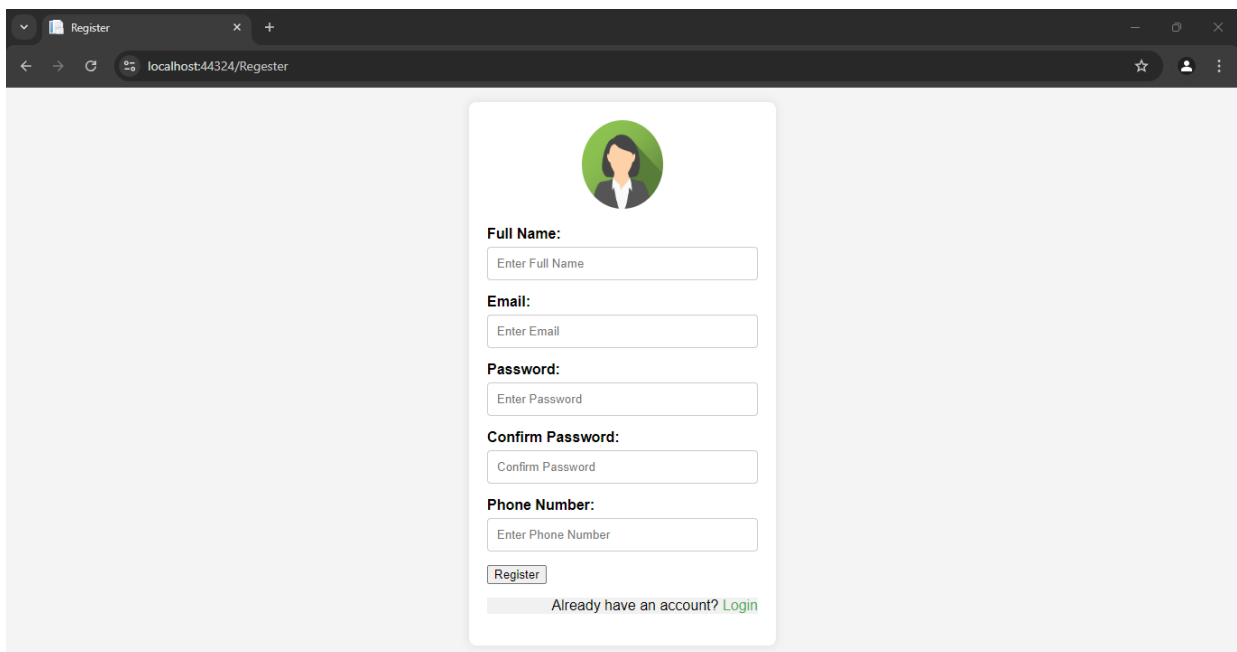
UX/UI

**5.1. Design Principles**

5.2. User Flow

5.1. Design Principles

1. Simplicity: The interface should be straightforward and easy to navigate, minimizing user effort and confusion.

2. Consistency: Design elements such as colors, fonts, and navigation should be consistent across the application to provide a unified experience.

3. Accessibility: The system should accommodate users with disabilities by following accessibility guidelines, such as providing keyboard navigation and screen reader support.

5.2. User Flow

1. Registration/Login:
   - Users enter their details to create an account or log in.
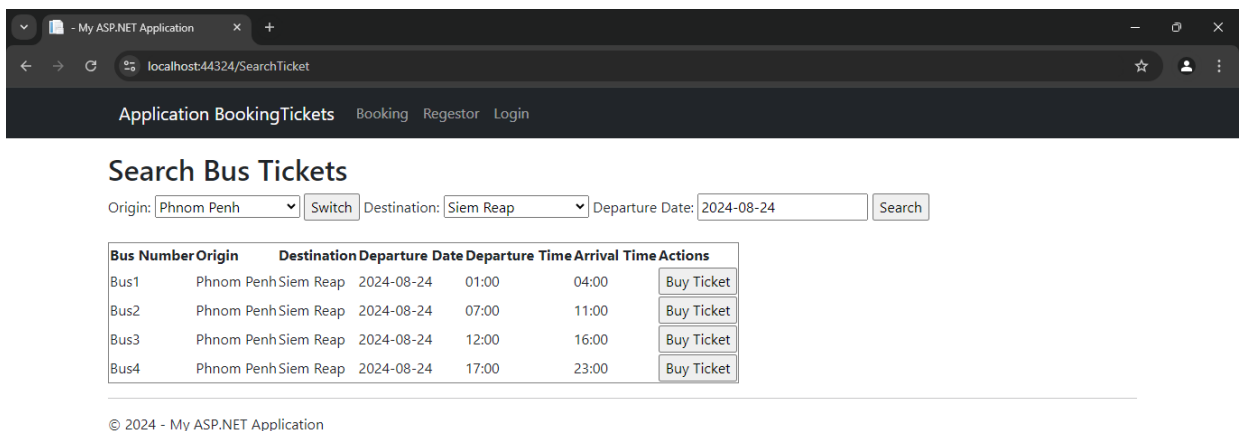   - Passwords are stored securely, and user sessions are managed effectively.
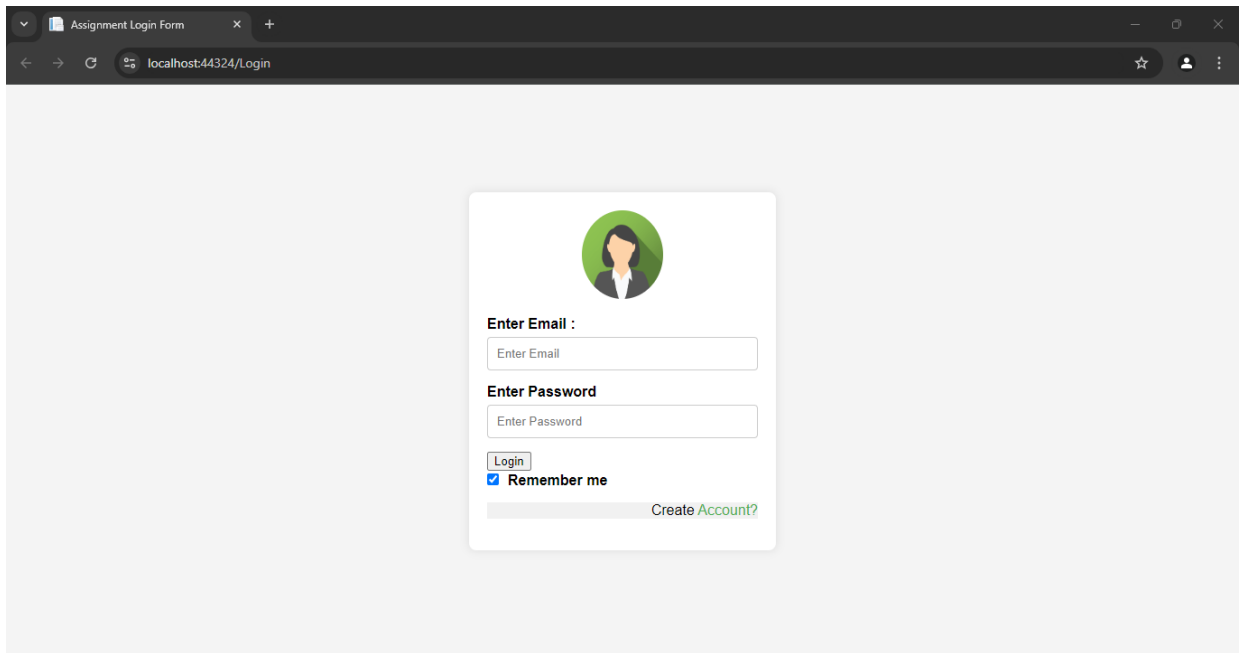
2. Route Search or search Bus Tickets:

   - o Users input search criteria such as origin, destination, and travel date.

   - o The system displays matching routes with detailed information.

3. Seat Selection:

   - o Users view an interactive map to select available seats.

   - o Users can choose their preferred seats and proceed to booking.

4. Payment:

- o Users enter payment details and confirm the booking.
- o The system processes the payment securely and provides confirmation.

Wireframes and Mockups

Provide visual representations of the following key pages:

- Homepage: The main landing page with search functionality and navigation.
- Route Search Page: Displays search results with route details.
- Seat Selection Page: Interactive map showing seat availability.
- User Dashboard: Interface for managing user profile and bookings.

Include annotations and explanations for each wireframe or mockup to clarify design decisions.

System Design

6.1. Architecture

6.1. Architecture Overview

1. Front-End: Built using ASP.NET to create a responsive and interactive user interface.
2. Back-End: Developed with C# to handle server-side logic and business processes.
3. Database: SQL Server is used to manage and store data related to users, bookings, routes, and transactions.

Components

1. Front-End:
   o HTML/CSS: For structuring and styling web pages.
   o JavaScript: For client-side interactivity and asynchronous data fetching.
2. Back-End:
   o ASP.NET WebForm: For managing HTTP requests and responses.
   o C#: Used for implementing business logic and handling data operations.
3. Database:
   o SQL Server: For storing user, booking, and route data.
   o Entity Framework: ORM tool for simplifying database interactions.

Diagram

Include a system architecture diagram that shows:

- User Interaction Flow: How users interact with the front-end.
- Data Flow: How data moves between the front-end, back-end, and database.

Back-End Functionalities

7.1. Key Functions

7.2. Sample Code

7.1. Key Functions

1. User Management:
   - ○ Registration: Code to handle user sign-up, including email validation and password hashing.
   - ○ Login: Authentication mechanism to verify user credentials.
   - ○ Profile Management: Code to update user information and handle password changes.

2. Route Management:
   - ○ CRUD Operations: Functions to create, read, update, and delete routes.
   - ○ Search Functionality: Logic to filter routes based on user input.

3. Booking System:
   - ○ Seat Availability: Code to manage seat availability and booking status.
   - ○ Booking Logic: Functions to handle booking creation, modifications, and cancellations.

4. Payment Processing:
   - ○ Integration: Code snippets for integrating with payment gateways like Stripe or PayPal.
   - ○ Transaction Management: Handling payment processing and updating booking status.

7.2. Sample Code

Provide examples of:

- User Authentication:

```
csharp
Copy code
// Example of user login method in C#
```

```csharp
public bool Login(string username, string password)
{
    var user = _context.Users.SingleOrDefault(u => u.Username == username);
    if (user != null && BCrypt.Net.BCrypt.Verify(password, user.PasswordHash))
    {
        // Generate authentication token
        return true;
    }
    return false;
}
```

- Booking Management:

  csharp
  Copy code
  ```csharp
  // Example of booking creation method in C#
  public Booking CreateBooking(int userId, int routeId, int seatId)
  {
      var booking = new Booking
      {
          UserId = userId,
          RouteId = routeId,
          SeatId = seatId,
          BookingDate = DateTime.Now
      };
      _context.Bookings.Add(booking);
      _context.SaveChanges();
      return booking;
  }
  ```

- Payment Integration:

  csharp
  Copy code
  ```csharp
  // Example of payment processing method in C#
  public async Task<bool> ProcessPayment(PaymentDetails details)
  {
      var client = new HttpClient();
      var response = await client.PostAsync("https://paymentgateway.com/api/pay", new
  StringContent(JsonConvert.SerializeObject(details), Encoding.UTF8,
  "application/json"));
      return response.IsSuccessStatusCode;
  }
  ```

Technologies

8.1. Front-End

**8.2. Back-End**

8.3. Additional Tools

8.1. Front-End

- ASP.NET: A framework for building dynamic web applications using C# and .NET. It supports server-side rendering and has tools for building scalable and secure web applications.

8.2. Back-End

- C#: A versatile programming language used for server-side application logic. It is integrated with .NET for building robust and scalable applications.
- SQL Server: A relational database management system used to store and manage data. It supports complex queries and transactions.
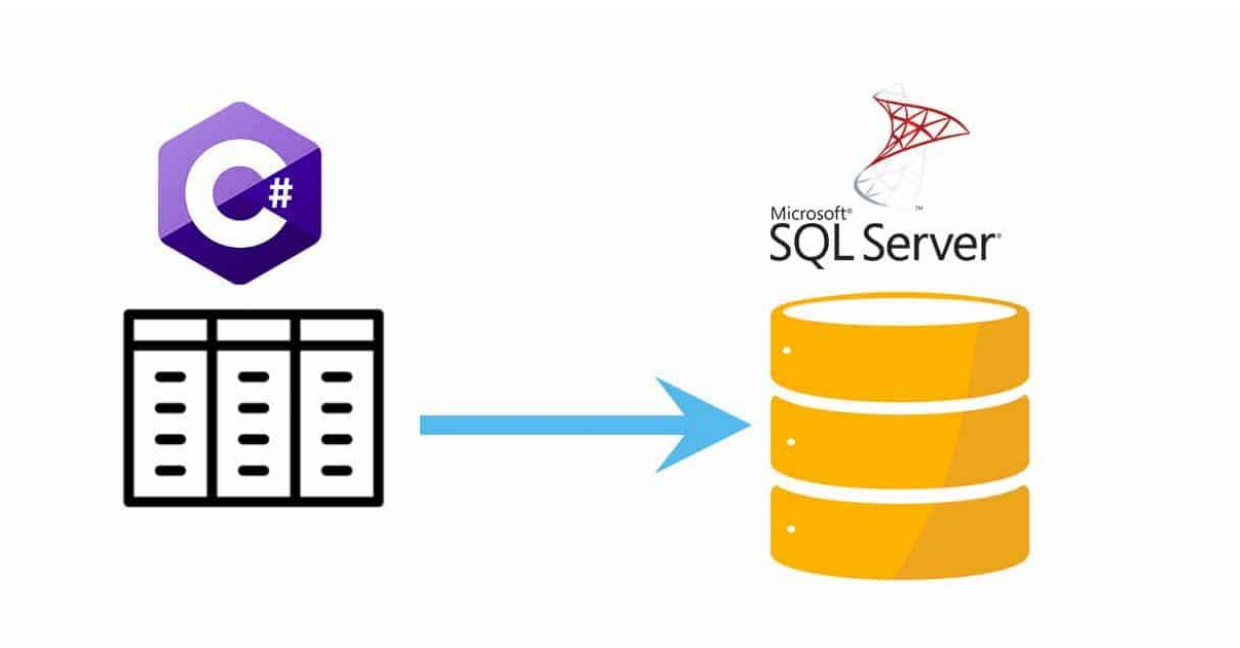
8.3. Additional Tools

- Entity Framework: An ORM (Object-Relational Mapper) that simplifies database operations by mapping database tables to .NET objects.
- ASP.NET Identity: A framework for handling user authentication and authorization in ASP.NET applications.

Development Tools

- Visual Studio: An integrated development environment (IDE) used for coding, debugging, and testing the application.
- SQL Server Management Studio (SSMS): A tool for managing SQL Server databases, including running queries and managing schema.

Summary

9.1. Overview

9.2. Future Directions

9.1. Project Overview

The "Book Me Bus System" provides an integrated solution for bus ticket booking, combining a user-friendly front-end with a powerful back-end. It

facilitates seamless user registration, route searching, seat selection, and payment processing, all within a secure and efficient framework.

9.2. Future Directions

Potential future improvements include:

- Mobile App Integration: Developing mobile applications to expand user access.
- Additional Payment Options: Offering a broader range of payment methods.
- Advanced Analytics: Using data insights to enhance service offerings and marketing strategies.

Appendices

1. Database Schema:

o **_SQL Scripts:_**

```sql
sql
Copy code
-- Create the database
CREATE DATABASE BookingTicketDB;
USE BookingTicketDB;

-- Create the Passengers table
CREATE TABLE Passengers (
    pasId INT IDENTITY(1,1) PRIMARY KEY,
    pasName NVARCHAR(100) NOT NULL,
    pasGender NVARCHAR(10) NOT NULL,
    pasPhone NVARCHAR(15) NOT NULL,
    pasEmail NVARCHAR(100) NOT NULL,
    pasConfirmEmail NVARCHAR(100) NOT NULL,
    pasNationality NVARCHAR(50) NOT NULL,
    pasSeat NVARCHAR(10) NOT NULL,
    CONSTRAINT chk_email CHECK (pasEmail =
pasConfirmEmail)
);

-- Create the tblRoutes table
CREATE TABLE tblRoutes (
    RouteID INT IDENTITY(1,1) PRIMARY KEY,
    BusNumber NVARCHAR(50),
    Origin NVARCHAR(50),
    Destination NVARCHAR(50),
    DepartureDate DATE,
    DepartureTime TIME,
    ArrivalTime TIME
);

-- Create the tblSeats table
CREATE TABLE tblSeats (
    SeatId INT IDENTITY(1,1) PRIMARY KEY,
    RouteId INT NOT NULL,
    SeatNumber NVARCHAR(10) NOT NULL,
    IsBooked BIT NOT NULL DEFAULT 0,
    CONSTRAINT FK_Route FOREIGN KEY (RouteId)
REFERENCES tblRoutes(RouteID)
);
```

```sql
-- Create the tblBookings table
CREATE TABLE tblBookings (
    BookingId INT IDENTITY(1,1) PRIMARY KEY,
    pasId INT NOT NULL,
    RouteId INT NOT NULL,
    SeatId INT NOT NULL,
    BookingDate DATETIME DEFAULT GETDATE(),
    FOREIGN KEY (pasId) REFERENCES Passengers(pasId),
    FOREIGN KEY (RouteId) REFERENCES tblRoutes(RouteID),
    FOREIGN KEY (SeatId) REFERENCES tblSeats(SeatId)
);

-- Create the tblUsers table
CREATE TABLE tblUsers (
    UserId INT PRIMARY KEY IDENTITY(1,1),
    Username NVARCHAR(50) NOT NULL UNIQUE,
    Password NVARCHAR(100) NOT NULL,
    Email NVARCHAR(100) NOT NULL,
    RegistrationDate DATETIME NOT NULL DEFAULT
GETDATE()
);

-- Create the TicketSales table
CREATE TABLE TicketSales (
    SaleID INT PRIMARY KEY IDENTITY(1,1),
    BookingID INT NOT NULL,
    SaleDateTime DATETIME DEFAULT GETDATE(),
    Amount DECIMAL(10, 2) NOT NULL,
    PaymentMethod NVARCHAR(50),
    FOREIGN KEY (BookingID) REFERENCES
tblBookings(BookingId)
);

-- Create the tblCheckout table
CREATE TABLE tblCheckout (
    Id INT PRIMARY KEY IDENTITY(1,1),
    BookingId INT,
    FullName NVARCHAR(100),
    Email NVARCHAR(100),
    Phone NVARCHAR(15),
    Address NVARCHAR(255),
    City NVARCHAR(100),
    State NVARCHAR(100),
```

ZipCode NVARCHAR(10),
CreditCardNumber NVARCHAR(20),
ExpirationDate DATE,
CVV NVARCHAR(4),
FOREIGN KEY (BookingId) REFERENCES
tblBookings(BookingId)
);

2. Code Samples:

o **User Authentication:**

csharp

Copy code

```csharp
// Example of user login method in C#
public bool Login(string username, string password)
{
    var user = _context.Users.SingleOrDefault(u => u.Username ==
username);
    if (user != null && BCrypt.Net.BCrypt.Verify(password,
user.PasswordHash))
    {
        // Generate authentication token
        return true;
    }
    return false;
}
```

o **Booking Management:**

csharp
Copy code
```csharp
// Example of booking creation method in C#
public Booking CreateBooking(int userId, int routeId, int seatId)
{
    var booking = new Booking
    {
        UserId = userId,
        RouteId = routeId,
        SeatId = seatId,
        BookingDate = DateTime.Now
    };
    _context.Bookings.Add(booking);
```

```
    _context.SaveChanges();
    return booking;
}
```

- o **Payment Integration:**

```csharp
csharp
Copy code
// Example of payment processing method in C#
public async Task<bool> ProcessPayment(PaymentDetails details)
{
    var client = new HttpClient();
    var response = await
client.PostAsync("https://paymentgateway.com/api/pay", new
StringContent(JsonConvert.SerializeObject(details), Encoding.UTF8,
"application/json"));
    return response.IsSuccessStatusCode;
}
```

3. Wireframes and Mockups:

- o Homepage: Visual layout of the landing page with search bar and navigation.

- o Route Search Page: Design showing search results and route details.

- o Seat Selection Page: Layout of the seat map and selection options.

- o User Dashboard: Design of the dashboard for viewing and managing bookings.

online reference:

1. https://www.w3schools.com

2. https://www.w3resource.com

3. https://learn.microsoft.com/en-us/aspnet/web-forms/

4. https://www.c-sharpcorner.com/article/create-a-sql-database-in-asp-net-web-form-using-visual-studio-2015-update-3/