



**IUBAT- International University of Business Agriculture and
Technology**

4 Embankment Drive Road, Sector 10, Uttara Model Town, Dhaka 1230

Project Report

on

Doctor Appointment System

Submitted by

Section: A

Fall-2022

Course Name: Visual Programming Lab

| Name | ID |
|-------------------------------|-----------------|
| Samiha Maisha Jeba | 20203017 |
| Mitu Ara Narsin | 20203019 |
| Kazi Johir Raihan Suny | 20203022 |
| Tanvir Ahmed Khan | 20203036 |

Submitted to

Rashedul Islam Rana

Assistant Professor

Date of Submission

24/12/2022

Section A: In this section, you are required to explain about the project given to you. Suppose if your project is about, Airplane Scheduling System then you have to explain about what the airplane scheduling system is, the expected features from this system and so on.

A doctor appointment booking system is an online system that allows patients to easily book their appointment at a particular clinic related to their health issue. Such as Dental, Injury, Metal health, Muscle strain, Obesity, Arthritis, Asthma, etc. issues.

In this report, we will learn about the basics of building an ASP.NET Core Razor Pages web app for doctor appointment systems. Here we can insert, delete doctors details and patient details. We can choose doctor id and patient id from appointment model. It will automatically add all the details of that selected patient and doctor. How we have created this project are described below-

Section B: In this section, you need to explain minimum three related models with model data (attributes/fields) of the system, such as for airplane scheduling, the models may be, Airplane Info, Passenger Info and Ticket Info, Schedule Info, Schedule Delay Info, Cabin crew and Pilot Info etc.

Doctor

- I. ID
- II. Name
- III. Consultant

Patient

- I. ID
- II. PatientName
- III. PatientPhone
- IV. PatientAddress
- V. PatientAge

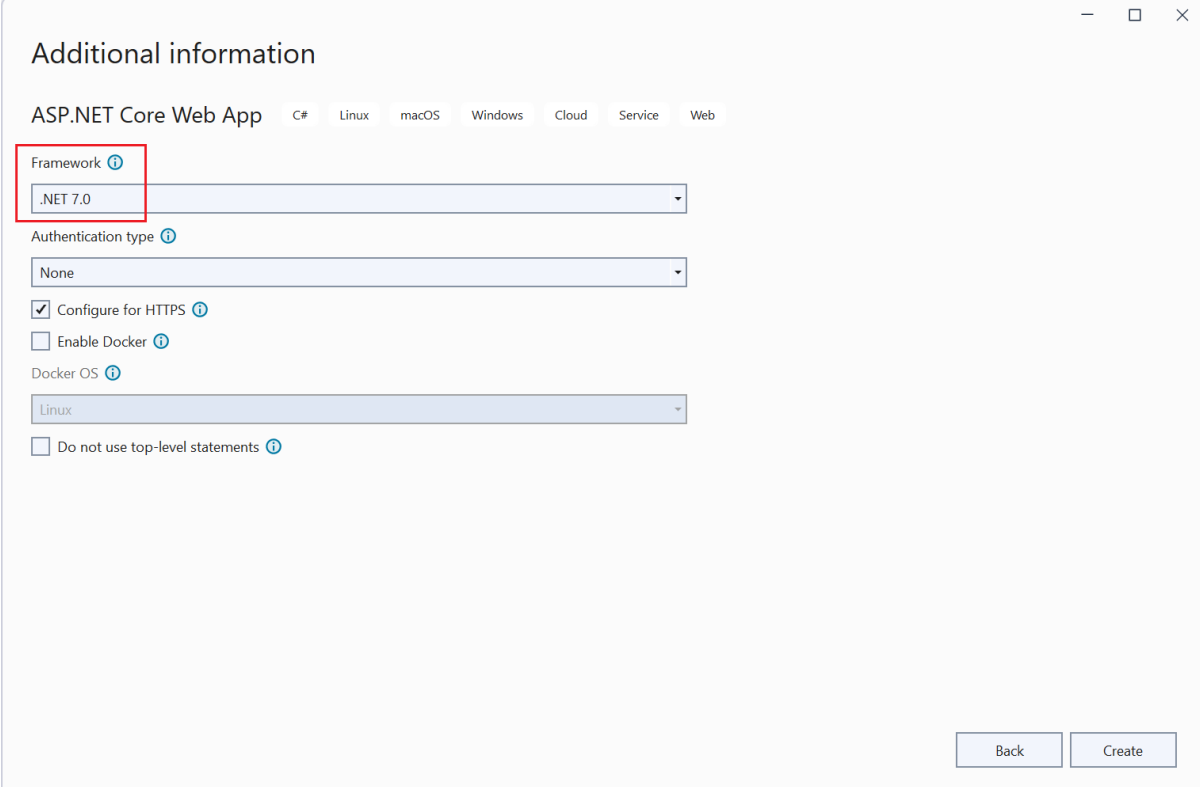
Appointment Info

- I. ID
- II. PatientName
- III. PatientPhone
- IV. PatientAddress
- V. PatientAge
- VI. ID
- VII. Name
- VIII. Consultant

Section C: Just like the Razor Page Step by Step Tutorial of Microsoft, you also need to mention and describe each step with proper screenshots so that anyone can copy your steps to recreate the project by their own.

Create a Razor Pages web app:

- Start Visual Studio and select Create a new project.
- In the Create a new project dialog, select ASP.NET Core Web App > Next.
- In the Configure your new project dialog, enter RazorPagesMovie for Project name. It's important to name the project **DoctorAppointmentSystem**, including matching the capitalization, so the namespaces will match when you copy and paste example code.
- Select Next.
- In the Additional information dialog:
 - ☐ Select .NET 7.0.
 - ☐ Verify: Do not use top-level statements unchecked.
 - Select Create.

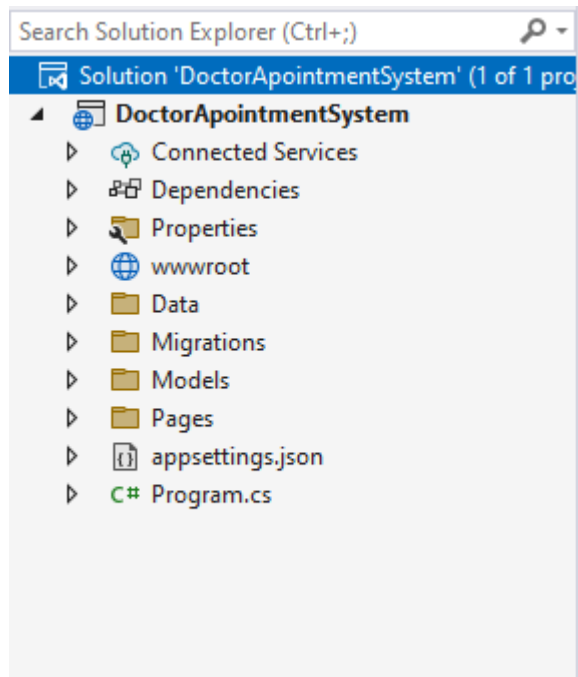


The screenshot shows the 'Additional information' dialog box in Visual Studio. The title bar says 'Additional information'. Below the title bar, there are tabs for 'ASP.NET Core Web App', 'C#', 'Linux', 'macOS', 'Windows', 'Cloud', 'Service', and 'Web'. The 'ASP.NET Core Web App' tab is selected. Under this tab, there are several settings:

- Framework:** A dropdown menu showing '.NET 7.0'. This section is highlighted with a red rectangle.
- Authentication type:** A dropdown menu showing 'None'.
- Configure for HTTPS:** A checkbox that is checked.
- Enable Docker:** A checkbox that is unchecked.
- Docker OS:** A dropdown menu showing 'Linux'.
- Do not use top-level statements:** A checkbox that is unchecked.

At the bottom right of the dialog, there are two buttons: 'Back' and 'Create'.

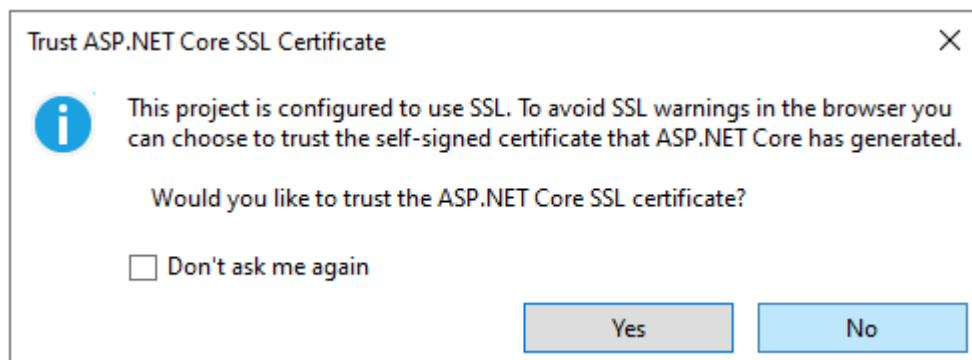
The following starter project is created:



Run the app:

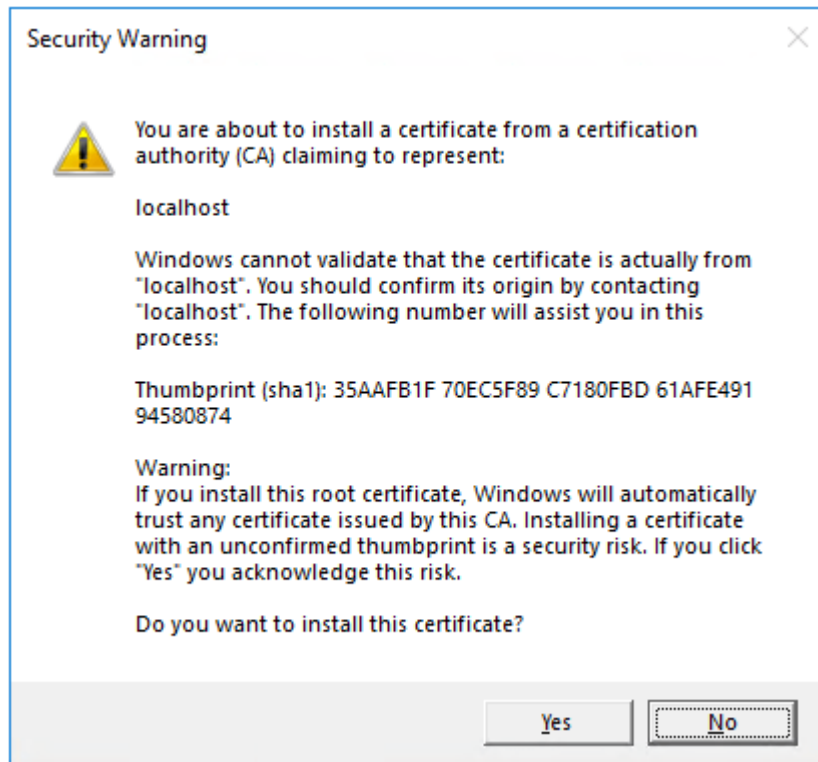
Select **DoctorAppointmentSystem** in Solution Explorer, and then press Ctrl+F5 to run without the debugger.

Visual Studio displays the following dialog when a project is not yet configured to use SSL:



Select Yes if you trust the IIS Express SSL certificate.

The following dialog is displayed:



Select Yes if you agree to trust the development certificate.

For information on trusting the Firefox browser, see Firefox SEC_ERROR_INADEQUATE_KEY_USAGE certificate error.

Visual Studio:

- Runs the app, which launches the Kestrel server.
- Launches the default browser at `https://localhost:<port>`, which displays the app's UI. `<port>` is the random port that is assigned when the app was created.

Examine the project files:

The following sections contain an overview of the main project folders and files that you'll work with in later tutorials.

Pages folder

Contains Razor pages and supporting files. Each Razor page is a pair of files:

- A `.cshtml` file that has HTML markup with C# code using Razor syntax.
- A `.cshtml.cs` file that has C# code that handles page events.

Supporting files have names that begin with an underscore. For example, the `_Layout.cshtml` file configures UI elements common to all pages. `_Layout.cshtml` sets up the navigation menu at the top of the page and the copyright notice at the bottom of the page. For more information, see [Layout in ASP.NET Core](#).

wwwroot folder

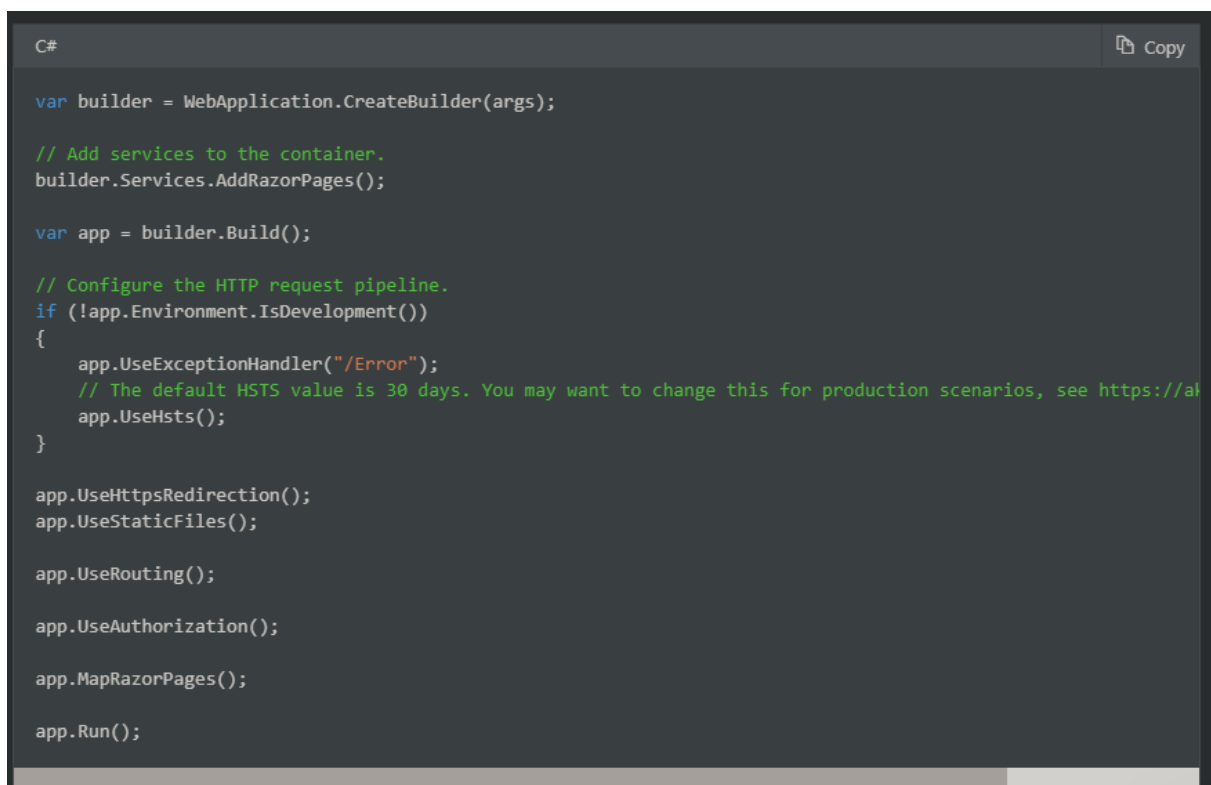
Contains static assets, like HTML files, JavaScript files, and CSS files. For more information, see [Static files in ASP.NET Core](#).

appsettings.json

Contains configuration data, like connection strings. For more information, see [Configuration in ASP.NET Core](#).

Program.cs

Contains the following code:

A screenshot of a code editor window with a dark theme. The title bar shows 'C#' and a 'Copy' button. The code is for Program.cs and includes comments in green and code in blue and white. The code creates a WebApplicationBuilder, adds Razor Pages, builds the app, configures the HTTP pipeline (including exception handling and HSTS), and runs the application.

```
C# Copy

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.
builder.Services.AddRazorPages();

var app = builder.Build();

// Configure the HTTP request pipeline.
if (!app.Environment.IsDevelopment())
{
    app.UseExceptionHandler("/Error");
    // The default HSTS value is 30 days. You may want to change this for production scenarios, see https://aka.ms/aspnetcore-hsts
    app.UseHsts();
}

app.UseHttpsRedirection();
app.UseStaticFiles();

app.UseRouting();

app.UseAuthorization();

app.MapRazorPages();

app.Run();
```

The following lines of code in this file create a `WebApplicationBuilder` with preconfigured defaults, add Razor Pages support to the Dependency Injection (DI) container, and builds the app:

```

C# Copy
var builder = WebApplication.CreateBuilder(args);

// Add services to the container.
builder.Services.AddRazorPages();

var app = builder.Build();

```

The developer exception page is enabled by default and provides helpful information on exceptions. Production apps should not be run in development mode because the developer exception page can leak sensitive information.

The following code sets the exception endpoint to /Error and enables [HTTP Strict Transport Security Protocol \(HSTS\)](#) when the app is not running in development mode:

```

C# Copy
// Configure the HTTP request pipeline.
if (!app.Environment.IsDevelopment())
{
    app.UseExceptionHandler("/Error");
    // The default HSTS value is 30 days. You may want to change this for production scenarios, see https://aka.ms/aspnetcore-hsts
    app.UseHsts();
}

```

For example, the preceding code runs when the app is in production or test mode.

The following code enables various Middleware:

- `app.UseHttpsRedirection();` : Redirects HTTP requests to HTTPS.
- `app.UseStaticFiles();` : Enables static files, such as HTML, CSS, images, and JavaScript to be served. For more information, see [Static files in ASP.NET Core](#).
- `app.UseRouting();` : Adds route matching to the middleware pipeline. For more information, see [Routing in ASP.NET Core](#)
- `app.MapRazorPages();` : Configures endpoint routing for Razor Pages.
- `app.UseAuthorization();` : Authorizes a user to access secure resources. This app doesn't use authorization, therefore this line could be removed.
- `app.Run();` : Runs the app.

Add three model to a Razor Pages app in ASP.NET Core

Add first data model (Doctor):

- In Solution Explorer, right-click the **DoctorAppointmentSystem** project > Add > New Folder. Name the folder **Models**.

- Right-click the Models folder. Select Add > Class. Name the class **Doctor**
- Add the following properties to the **Doctor** class:

```

1  using System.ComponentModel.DataAnnotations;
2  namespace DoctorAppointmentSystem.Models
3  {
4      7 references
4      public class Doctor
5      {
6          [Display(Name = "ID")]
7          14 references
7          public int DoctorId { get; set; }
8          12 references
8          public string? Name { get; set; }
9          12 references
9          public string? Consultant { get; set; }
10     }
11 }
12

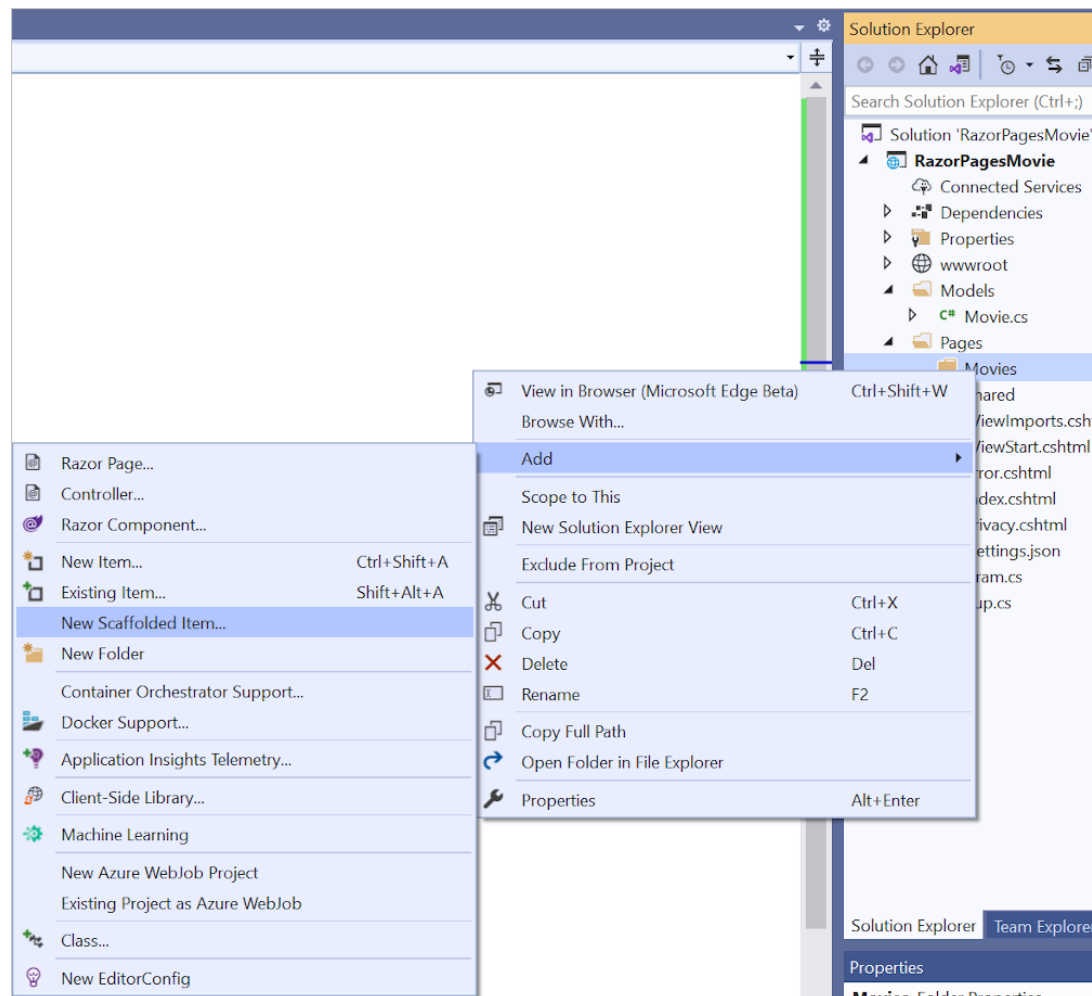
```

The Doctor class contains:

- The ID field is required by the database for the primary key.
 - The question mark after string indicates that the property is nullable.
- Build the project to verify there are no compilation errors.

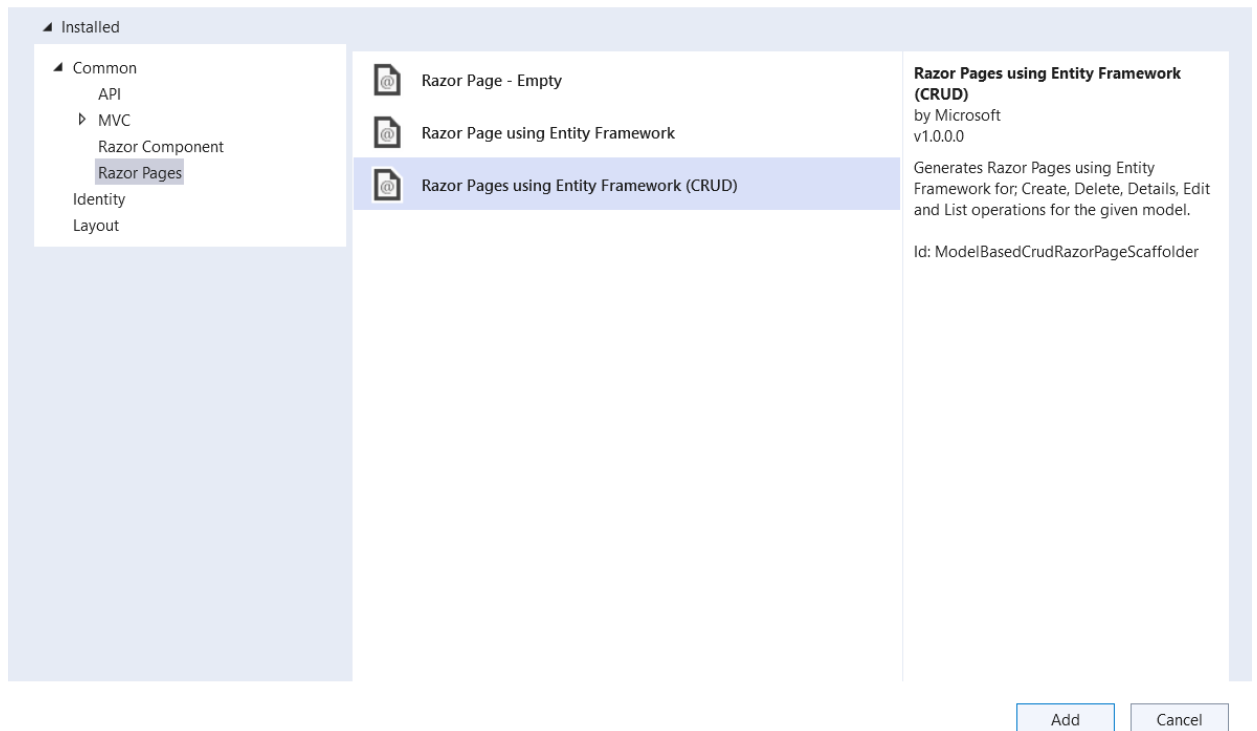
Scaffold the Doctor model:

- Create the Pages/Doctors folder:
 - Right-click on the Pages folder > Add > New Folder.
 - Name the folder Doctors.
- Right-click on the Pages/Doctors folder > Add > New Scaffolded Item.



- In the Add New Scaffold dialog, select Razor Pages using Entity Framework (CRUD) > Add.

Add New Scaffolded Item



- Complete the Add Razor Pages using Entity Framework (CRUD) dialog:
 - ☐ In the Model class drop down, select Doctors(DoctorAppointmentSystem.Models).
 - ☐ In the Data context class row, select the + (plus) sign.
 - i) In the Add Data Context dialog, the class name DoctorAppointmentSystem.Data.DoctorAppointmentSystemContext is generated.
 - ☐ Select Add.

×

Add Razor Pages using Entity Framework (CRUD)

Generates Razor Pages using Entity Framework for; Create, Delete, Details, Edit and List operations for the selected model.

Model class

Doctor (DoctorAppointmentSystem.Models) ▾

Data context class

DoctorAppointmentSystem.Data.DoctorAppointmentSystemContext ▾

+

Options

☐ Create as a partial view

☒ Reference script libraries

☒ Use a layout page

...

(Leave empty if it is set in a Razor _viewstart file)

Add

Cancel

Add second data model (Patient):

- In Solution Explorer, right-click the Models folder. Select Add > Class. Name the class Patient
- Add the following properties to the Patient class:

```

1  using System.ComponentModel.DataAnnotations;
2
3  namespace DoctorAppointmentSystem.Models
4  {
5      7 references
6      public class Patient
7      {
8          14 references
9          public int PatientId { get; set; }
10
11          [Display(Name = "Patient Name")]
12          12 references
13          public string PatientName { get; set; }
14
15          [Display(Name = "Patient Phone")]
16          12 references
17          public string PatientPhone { get; set; } = string.Empty;
18
19          [Display(Name = "Patient Address")]
20          12 references
21          public string PatientAddress { get; set; } = string.Empty;
22
23          [Display(Name = "Patient Age")]
24          12 references
25          public string PatientAge { get; set; } = string.Empty;
26      }
27  }

```

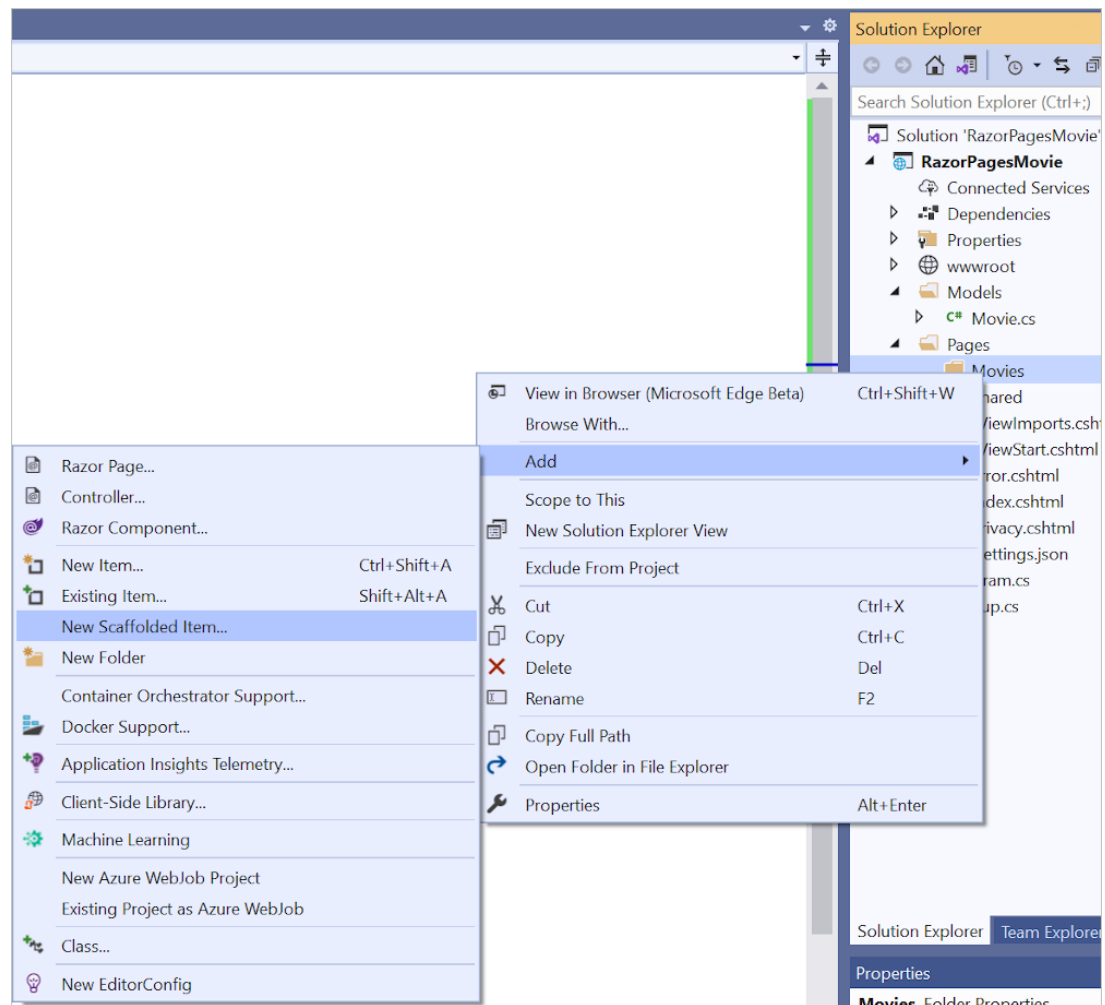
The Patient class contains:

- The ID field is required by the database for the primary key.

Build the project to verify there are no compilation errors.

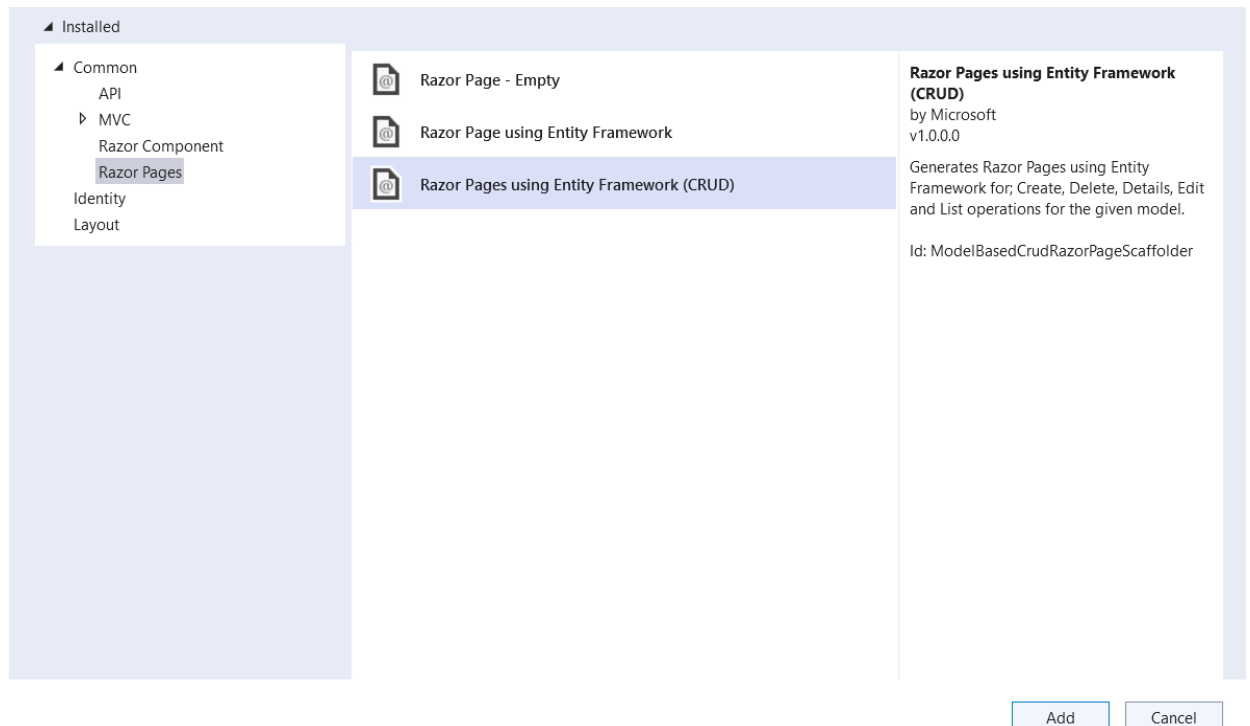
Scaffold the Patient model:

- Create the Pages/Patients folder:
 - Right-click on the Pages folder > Add > New Folder.
 - Name the folder Patients.
- Right-click on the Pages/Patients folder > Add > New Scaffolded Item.



- In the Add New Scaffold dialog, select Razor Pages using Entity Framework (CRUD) > Add.

Add New Scaffolded Item



- Complete the Add Razor Pages using Entity Framework (CRUD) dialog:
 - ☐ In the Model class drop down, select Patients(DoctorAppointmentSystem.Models).
 - ☐ In the Data context class row, select the + (plus) sign.
 - i) In the Add Data Context dialog, the class name DoctorAppointmentSystem.Data.DoctorAppointmentSystemContext is generated.
 - ☐ Select Add.

×

Add Razor Pages using Entity Framework (CRUD)

Generates Razor Pages using Entity Framework for; Create, Delete, Details, Edit and List operations for the selected model.

Model class

Patient (DoctorAppointmentSystem.Models)

Data context class

DoctorAppointmentSystem.Data.DoctorAppointmentSystemContext

+

Options

☐ Create as a partial view

☒ Reference script libraries

☒ Use a layout page

...

(Leave empty if it is set in a Razor _viewstart file)

Add

Cancel

Add third data model (Appointment):

- In Solution Explorer, Right-click the Models folder. Select Add > Class. Name the class **Appointment**
- Add the following properties to the **Appointment** class:

```

1  using System.ComponentModel.DataAnnotations;
2  using System.ComponentModel.DataAnnotations.Schema;
3  using DoctorAppointmentSystem.Models;
4  namespace DoctorAppointmentSystem.Models
5  {
6      6 references
7      public class Appointment
8      {
9          [Key]
10         [Required]
11         11 references
12         public int Id { get; set; }
13
14         [Display(Name = "Doctor ID")]
15         6 references
16         public int DoctorId { get; set; }
17
18         [Display(Name = "Patient ID")]
19         6 references
20         public int PatientId { get; set; }
21
22         [ForeignKey(nameof(PatientId))]
23
24         7 references
25         public Patient PatientData { get; set; }
26
27         [ForeignKey(nameof(DoctorId))]
28         7 references
29         public Doctor DoctorData { get; set; }
30     }
31 }

```

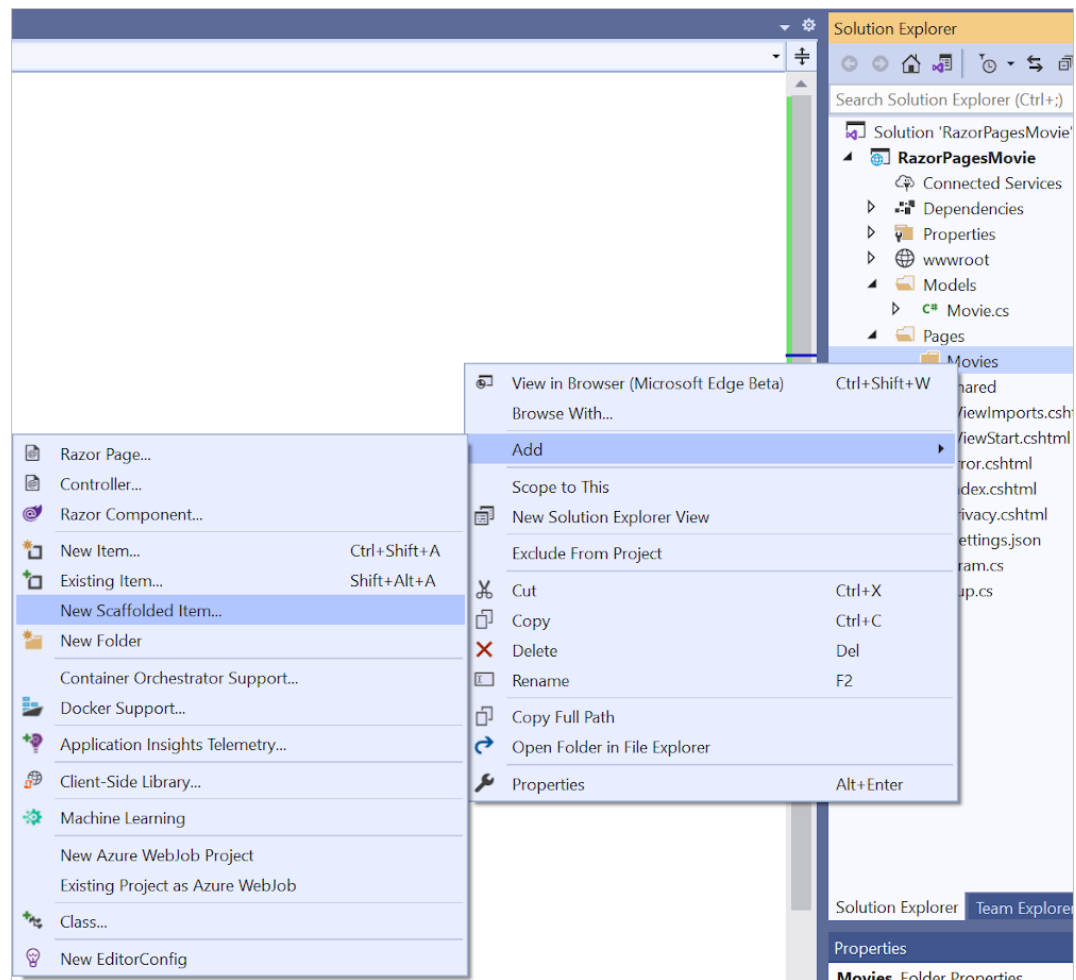
The Appointment class contains:

- The ID field is required by the database for the primary key.
- The PatientID field is foreign key.
- The question mark after string indicates that the property is nullable. For more information, see Nullable reference types.

Build the project to verify there are no compilation errors.

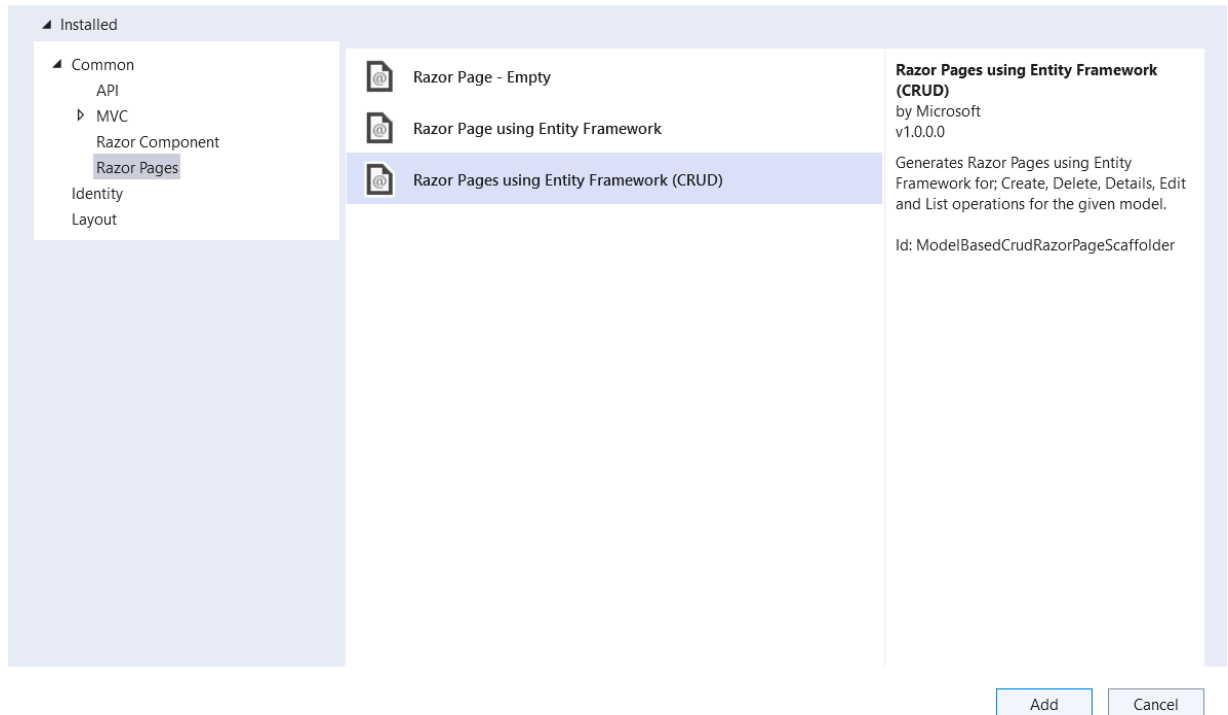
Scaffold the Appointment model:

- Create the Pages/Appointments folder:
 - Right-click on the Pages folder > Add > New Folder.
 - Name the folder Appointments.
- Right-click on the Pages/Appointments folder > Add > New Scaffolded Item.



- In the Add New Scaffold dialog, select Razor Pages using Entity Framework (CRUD) > Add.

Add New Scaffolded Item



- Complete the Add Razor Pages using Entity Framework (CRUD) dialog:
 - ☐ In the Model class drop down, select `Appointments(DoctorAppointmentSystem.Models)`.
 - ☐ In the Data context class row, select the + (plus) sign.
 - i) In the Add Data Context dialog, the class name `DoctorAppointmentSystem.Data.DoctorAppointmentSystemContext` is generated.
 - ☐ Select Add.

Add Razor Pages using Entity Framework (CRUD)

Generates Razor Pages using Entity Framework for; Create, Delete, Details, Edit and List operations for the selected model.

Model class: Appointment (DoctorAppointmentSystem.Models)

Data context class: DoctorAppointmentSystem.Data.DoctorAppointmentSystemContext

Options

- ☐ Create as a partial view
- ☒ Reference script libraries
- ☒ Use a layout page

(Leave empty if it is set in a Razor _viewstart file)

Add Cancel

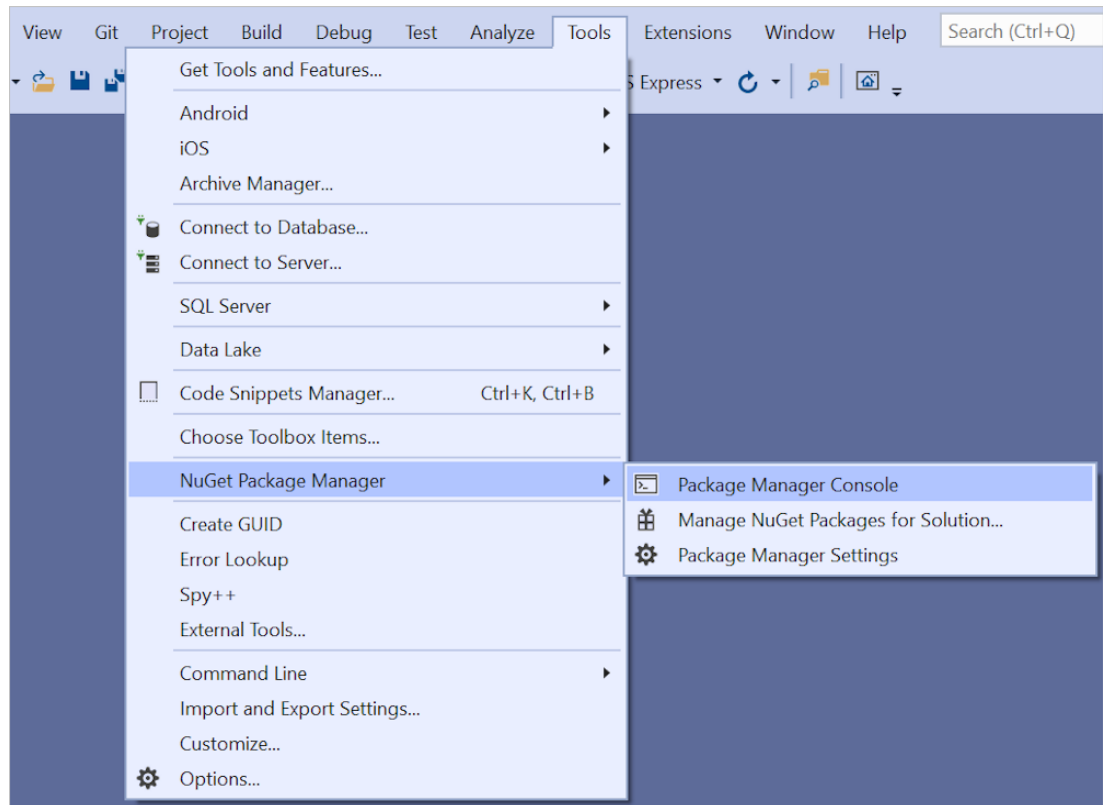
Create the initial database schema using EF's migration feature

The migrations feature in Entity Framework Core provides a way to:

1. Create the initial database schema.
2. Incrementally update the database schema to keep it in sync with the app's data model. Existing data in the database is preserved.

In this section, the Package Manager Console (PMC) window is used to:

1. Add an initial migration.
2. Update the database with the initial migration.



In the PMC, enter the following commands:

```
PowerShell
Add-Migration InitialCreate
Update-Database
```

Test the app

1. Run the app and append /Movies to the URL in the browser (http://localhost:port/doctors).

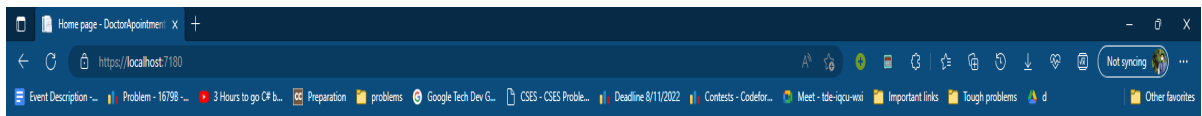
If you receive the following error:

SqlException: Cannot open database "DoctorAppointmentSystemContext-GUID" requested by the login. The login failed.Login failed for user 'User-name'.

You missed the migration step.

2. Test the Create New link.
3. Test the Edit, Details, and Delete links.

Section D: Screenshots for each functionality of your webpage and SQL Server Database such as, Insert, Edit, Details and Delete.



Doctor Appointment System [Doctors](#) [Patients](#) [Appointment](#) [Privacy](#) [Contact Us](#)

Welcome to the Doctors Appointment System



© 2022 - DoctorAppointmentSystem - [Privacy](#)

Doctors List

[Add new Doctor](#)

| ID | Name | Consultant | | | |
|----|-----------------------------|--------------------|---------------------------|-------------------------|------------------------|
| 1 | Dr. Tanvir Ahmed Khan | Orthopedic | Customize | Details | Delete |
| 2 | Dr. Kazi Jahir Raihan Sunny | Darmatologist | Customize | Details | Delete |
| 3 | Dr. Samiha Maisha Jeba | Cardiologist | Customize | Details | Delete |
| 4 | Dr. Mitu Ara Nasrin | Cardiology Surgeon | Customize | Details | Delete |
| 5 | Dr. Rashedul Islam | Dramatologist | Customize | Details | Delete |

Create

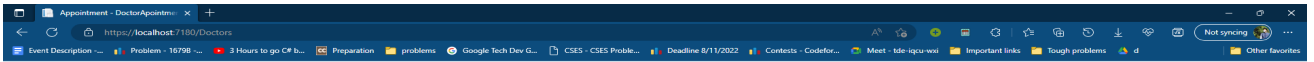
Doctor

Name

Consultant

[Create](#)

[Back to List](#)



Doctors List

[Add new Doctor](#)

| ID | Name | Consultant | | | |
|----|----------------------------|--------------------|---------------------------|-------------------------|------------------------|
| 1 | Dr. Tanvir Ahmed Khan | Orthopedic | Customize | Details | Delete |
| 2 | Dr. Kazi Jahir Raihan Suny | Darmatologist | Customize | Details | Delete |
| 3 | Dr. Samiha Maisha Jeba | Cardiologist | Customize | Details | Delete |
| 4 | Dr. Mitu Ara Nasrin | Cardiology Surgeon | Customize | Details | Delete |
| 5 | Dr. Rashedul Islam | Dramatologist | Customize | Details | Delete |
| 6 | Dr MD Faysal Islam | Urologist | Customize | Details | Delete |

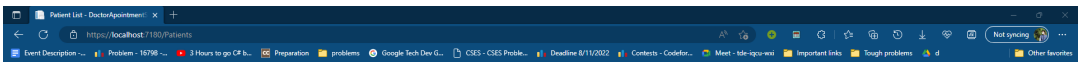


Details

Patient

Patient Name Kalam
Patient Phone 0198525461
Patient Address Mirpur DOH
Patient Age 29

[Edit](#) | [Back to List](#)



Patient List

[New Patient](#)

| Patientid | Patient Name | Patient Phone | Patient Address | Patient Age | | | |
|-----------|------------------|---------------|-----------------|-------------|---------------------------|-------------------------|------------------------|
| 2 | Mitu Ara Nasreen | 01956235656 | Mirpur Dhaka | 29 | Customize | Details | Delete |
| 3 | Oyshi Haque | 01532564798 | Uttara | 50 | Customize | Details | Delete |
| 4 | Kalam | 0198525461 | Mirpur DOH | 29 | Customize | Details | Delete |

Patient List - DoctorApoinmentSystem

https://localhost:7100/Patients

Event Description ~ Problem - 16798 ~ 3 Hours to go CP b... Preparation problems Google Tech Dev G... CSES - CSES Proble... Deadline 8/11/2022 Contests - Codefor... Meet - tde-iqu-wsi Important links Tough problems d Other favorites

Doctor Apoinment SystemDoctorsPatientsAppointmentPrivacyContact Us

Patient List

New Patient

| PatientId | Patient Name | Patient Phone | Patient Address | Patient Age | |
|-----------|------------------|---------------|-----------------|-------------|-----------------------------------|
| 2 | Mitu Ara Nasreen | 01956235656 | Mirpur Dhaka | 29 | <div>CustomizeDetailsDelete</div> |
| 3 | Oyshee | 01532564798 | Uttara | 50 | <div>CustomizeDetailsDelete</div> |
| 4 | Kalam | 0198525461 | Mirupur DOH | 29 | <div>CustomizeDetailsDelete</div> |

© 2022 - DoctorApoinmentSystem - Privacy

Appointment - DoctorApoinmentSystem

https://localhost:7100/Appointments

Event Description ~ Problem - 16798 ~ 3 Hours to go CP b... Preparation problems Google Tech Dev G... CSES - CSES Proble... Deadline 8/11/2022 Contests - Codefor... Meet - tde-iqu-wsi Important links Tough problems d Other favorites

Doctor Apoinment SystemDoctorsPatientsAppointmentPrivacyContact Us

Appointment List

New Appointment

| Patient ID | Patient Name | Phone | Address | Age | >>> | Doctor ID | Doctor | Consultant | |
|--------------|------------------|-------------|--------------|-----|-----|--------------|----------------------------|---------------|-----------------------------------|
| <div>2</div> | Mitu Ara Nasreen | 01956235656 | Mirpur Dhaka | 29 | >>> | <div>2</div> | Dr. Kazi Jahir Raihan Suny | Darmatologist | <div>CustomizeDetailsCancel</div> |
| <div>4</div> | Kalam | 0198525461 | Mirupur DOH | 29 | >>> | <div>3</div> | Dr. Samiha Maisha Jeba | Cardiologist | <div>CustomizeDetailsCancel</div> |

© 2022 - DoctorApoinmentSystem - Privacy

Create Appointment

Doctor ID
6

Patient ID
4

Create

[Back to List](#)

Delete

Are you sure you want to delete this?

Doctor

Name Dr. Tanvir Ahmed Khan
Consultant Orthopedic

[Delete](#) | [Back to List](#)

Appointment List

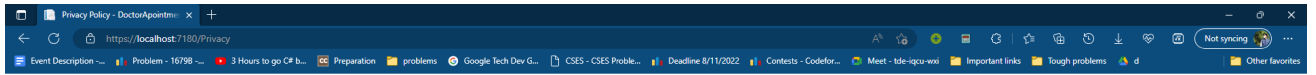
[New Appointment](#)

| Patient ID | Patient Name | Phone | Address | Age | >>> | Doctor ID | Doctor | Consultant | | | |
|--------------------------------|------------------|-------------|--------------|-----|-----|--------------------------------|----------------------------|---------------|---------------------------|-------------------------|------------------------|
| <input type="text" value="2"/> | Mitu Ara Nasreen | 01956235656 | Mirpur Dhaka | 29 | >>> | <input type="text" value="2"/> | Dr. Kazi Jahir Raihan Suny | Dermatologist | Customize | Details | Cancel |
| <input type="text" value="4"/> | Kalam | 0198525461 | Mirupur DOH | 29 | >>> | <input type="text" value="3"/> | Dr. Samiha Maisha Jeba | Cardiologist | Customize | Details | Cancel |
| <input type="text" value="4"/> | Kalam | 0198525461 | Mirupur DOH | 29 | >>> | <input type="text" value="6"/> | Dr MD Faysal Islam | Urologist | Customize | Details | Cancel |

Doctors List

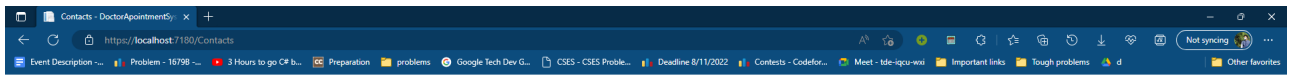
[Add new Doctor](#)

| ID | Name | Consultant | | | |
|----|----------------------------|--------------------|---------------------------|-------------------------|------------------------|
| 2 | Dr. Kazi Jahir Raihan Suny | Darmatologist | Customize | Details | Delete |
| 3 | Dr. Samiha Maisha Jeba | Cardiologist | Customize | Details | Delete |
| 4 | Dr. Mitu Ara Nasrin | Cardiology Surgeon | Customize | Details | Delete |
| 5 | Dr. Rashedul Islam | Dramatologist | Customize | Details | Delete |
| 6 | Dr MD Faysal Islam | Urologist | Customize | Details | Delete |



Privacy Policy

Use this page to detail your site's privacy policy.



Contacts

Phone: +8801XXXXXXXXXXXX

Email: Example@email.com