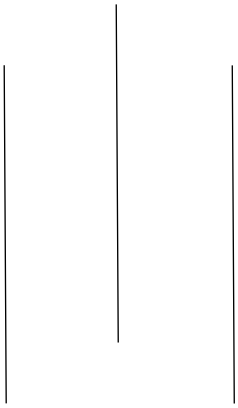
FEDERATION UNIVERSITY

ITECH7415 Masters Project



Development guide

"Electronic Health Records"

Table of Contents

Introduction 3
Purpose of the Document 3
Development Tools and Technology Stack 3
IDE: Visual Studio 2022 3
.NET Framework 8 3
SQL Express 4
SQL Server Management Studio (SSMS) 4
Refer to Setup Guide 4
Layers of the EMR Simulator Web App Project 5
Application Layer (Class Library) 5
Domain Layer (Class Library) 5
Infrastructure Layer (Class Library) 5
WebApp Layer 6
Connection String 6
Publishing the EmrSimulator website 7
Steps to publish from Visual Studio 7
Deployment to IIS 8
Stop the website from IIS 8
Start the website from IIS 10
Browse EmrSimulator application 10
Login to EmrSimulator 11

Introduction

Purpose of the Document

The purpose of this developer guide is to facilitate a smooth transition and onboarding process for new developers working on the EMR Simulator Web App. This guide aims to provide comprehensive documentation and best practices related to the project's clean architecture, which consists of four distinct layers: Application, Domain, Infrastructure, and WebApp.

Development Tools and Technology Stack

IDE: Visual Studio 2022

The EMR Simulator Web App has been developed using **Visual Studio Community 2022 (64-bit)**, a powerful Integrated Development Environment (IDE) that provides a wide range of features for building .NET applications. Visual Studio 2022 offers advanced debugging tools, IntelliSense for code completion, integrated version control, and a user-friendly interface that enhances developer productivity.

.NET Framework 8

The entire application is built using .NET Framework 8, which is designed to support modern development practices while ensuring compatibility with a broad range of libraries and tools. The choice of .NET Framework 8 allows for:

Rich Library Support: Access to a vast array of libraries that simplify tasks like data access, authentication, and web services, thus accelerating the development process.

Enhanced Performance: Optimisations in .NET Framework 8 provide better performance and resource management, making it suitable for building scalable applications.

SQL Express

SQL Server Express is a free, lightweight edition of Microsoft SQL Server, ideal for small to medium-sized applications. It provides core database features like data storage, querying, and management while using minimal resources, making it suitable for local or limited-scale applications. SQL Server Express supports essential tools like SQL Server Management Studio (SSMS) for easy database management and is compatible with applications developed in .NET, enabling secure, reliable data handling with moderate hardware requirements.

SQL Server Management Studio (SSMS)

SQL Server Management Studio (SSMS) is a free, comprehensive tool for managing SQL Server databases, including SQL Server Express. It provides an intuitive interface for tasks like creating databases, writing and executing queries, managing security, and performing backups. SSMS simplifies database administration with features for monitoring performance, troubleshooting issues, and optimising queries, making it essential for database management and maintenance in SQL Server environments.

Refer to Setup Guide

For more information on setting up IIS web server, SQL Server Express, SQL Management Studio and .Net Framework, please refer the setup guide provided.

Layers of the EMR Simulator Web App Project

Application Layer (Class Library)

Purpose: This layer contains the application logic and defines the use cases for the system. It coordinates between the user interface and the domain layer.

Responsibilities:

Interfacing with services and repositories to retrieve and manipulate data.

Benefits: Isolates the business logic from the presentation and data access layers, making the application easier to maintain and test.

Domain Layer (Class Library)

Purpose: This layer represents the business logic and domain entities. It encapsulates the rules and behaviours of the business.

Responsibilities:

Defining domain entities (e.g., Patient, Medication).

Implementing business rules and validation.

Interacting with the application layer to respond to commands or queries.

Benefits: Promotes a clear understanding of the business model, allowing developers to focus on business requirements and logic without worrying about technical details.

Infrastructure Layer (Class Library)

Purpose: This layer provides the technical capabilities needed to support the application, including data access, external services, and configuration.

Responsibilities:

Managing data persistence, connecting SQL database using (SQL Data Client or can introduce Entity Framework).

Can be enhanced for handling communication with external APIs or services.

Benefits: Abstracts technical implementation details, making it easier to swap out technologies (e.g., changing databases) without affecting the core application logic.

WebApp Layer

Purpose: The presentation layer of the application, responsible for user interaction and the user interface (UI).

Responsibilities:

Implementing the front-end (UI) using ASP.NET.

Managing user input and displaying data to users.

Interacting with the application layer to perform operations.

Benefits: Provides a clear separation between the UI and business logic, allowing for independent development and testing of the user interface.

Connection String

```
"ConnectionStrings": {
  "EmrSimulationConnection": "Server=.\\SQLEXPRESS;Database=EmrSimulator;User ID=sa;Password=Emrp@ssword123!;"
}
```

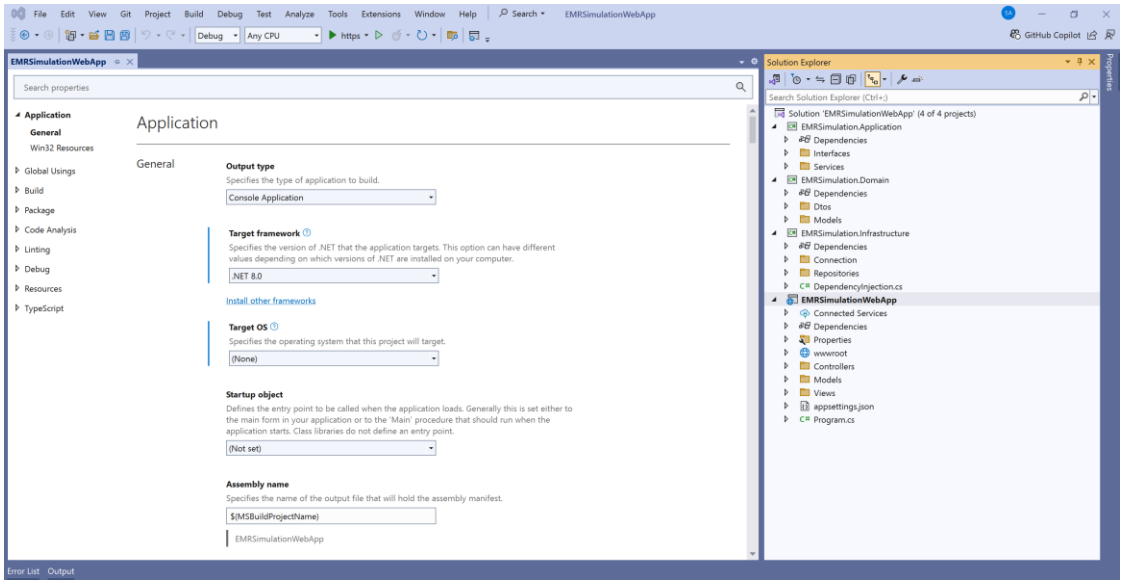
Each layer of the EMR Simulator Web App—the Application, Domain, Infrastructure, and WebApp layers has been developed using **.NET Framework 8**, promoting a cohesive architecture and allowing for smooth interactions between the layers.

By leveraging **Visual Studio 2022** and **.NET Framework 8**, developers can take advantage of the latest tools and features, ensuring that the EMR Simulator is not only functional but also maintainable and extensible for future development needs.

Note:

Application, Domain and Infrastructure layers are class library projects.

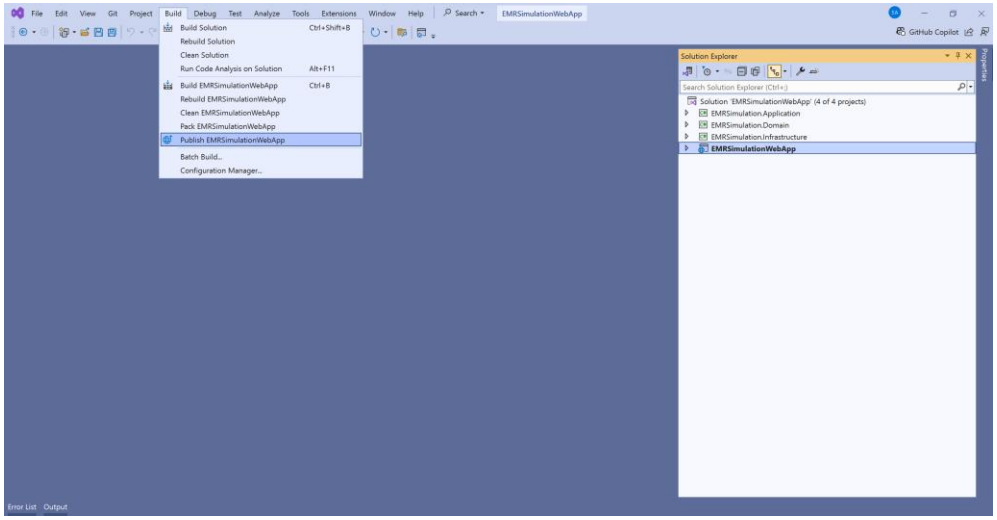
Class library can be easily referenced and reused across different projects or services, promoting modularity.

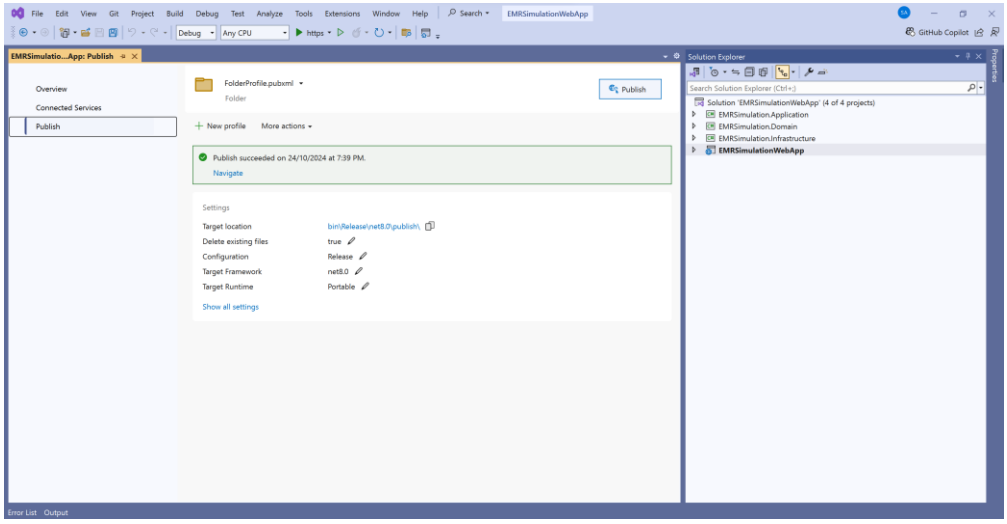


Publishing the EmrSimulator website

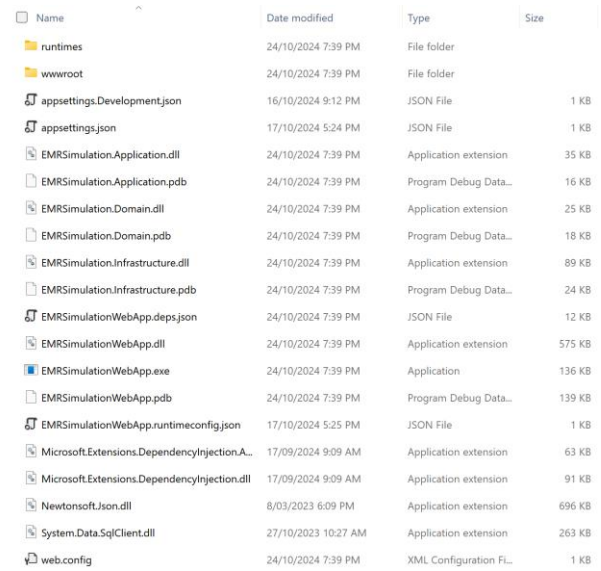
If some development work has been done or if it is just to create the publish files and to deploy to IIS.

Steps to publish from Visual Studio





Click on the **bin\Release\net8.0\publish**, will open the below folder

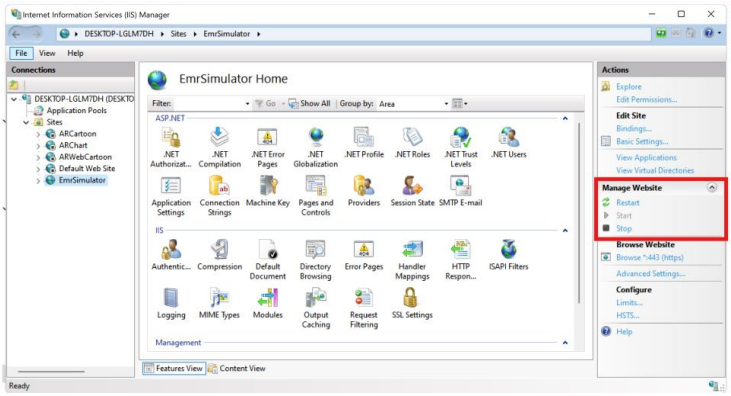


Name	Date modified	Type	Size
runtimes	24/10/2024 7:39 PM	File folder	
wwwroot	24/10/2024 7:39 PM	File folder	
appsettings.Development.json	16/10/2024 9:12 PM	JSON File	1 KB
appsettings.json	17/10/2024 5:24 PM	JSON File	1 KB
EMRSimulation.Application.dll	24/10/2024 7:39 PM	Application extension	35 KB
EMRSimulation.Application.pdb	24/10/2024 7:39 PM	Program Debug Data...	16 KB
EMRSimulation.Domain.dll	24/10/2024 7:39 PM	Application extension	25 KB
EMRSimulation.Domain.pdb	24/10/2024 7:39 PM	Program Debug Data...	18 KB
EMRSimulation.Infrastructure.dll	24/10/2024 7:39 PM	Application extension	89 KB
EMRSimulation.Infrastructure.pdb	24/10/2024 7:39 PM	Program Debug Data...	24 KB
EMRSimulationWebApp.deps.json	24/10/2024 7:39 PM	JSON File	12 KB
EMRSimulationWebApp.dll	24/10/2024 7:39 PM	Application extension	575 KB
EMRSimulationWebApp.exe	24/10/2024 7:39 PM	Application	136 KB
EMRSimulationWebApp.pdb	24/10/2024 7:39 PM	Program Debug Data...	139 KB
EMRSimulationWebApp.runtimeconfig.json	17/10/2024 5:25 PM	JSON File	1 KB
Microsoft.Extensions.DependencyInjection.A...	17/09/2024 9:09 AM	Application extension	63 KB
Microsoft.Extensions.DependencyInjection.dll	17/09/2024 9:09 AM	Application extension	91 KB
Newtonsoft.Json.dll	8/03/2023 6:09 PM	Application extension	696 KB
System.Data.SqlClient.dll	27/10/2023 10:27 AM	Application extension	263 KB
web.config	24/10/2024 7:39 PM	XML Configuration Fi...	1 KB

Deployment to IIS

Stop the website from IIS

Make sure to stop the website from IIS and then paste to the **C:\inetpub\wwwroot\EmrSimulator** folder.

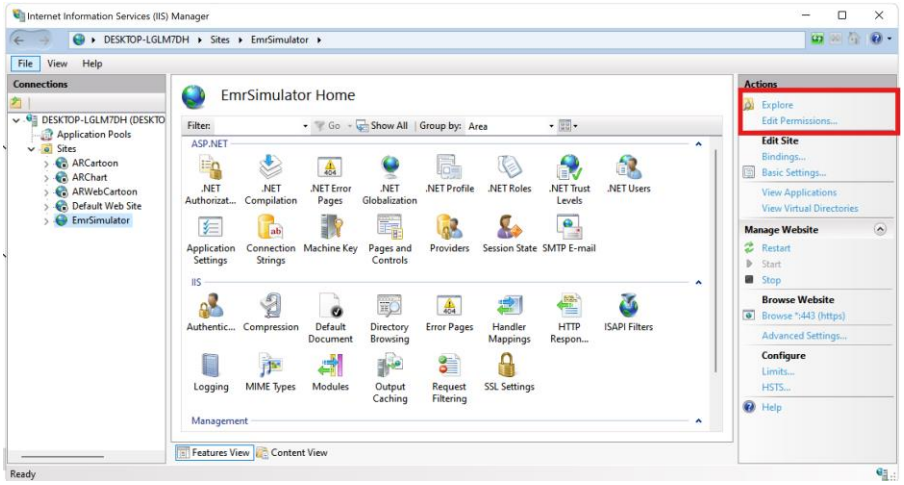


Copy the files in the folder and paste to the IIS **EmrSimulator** folder.

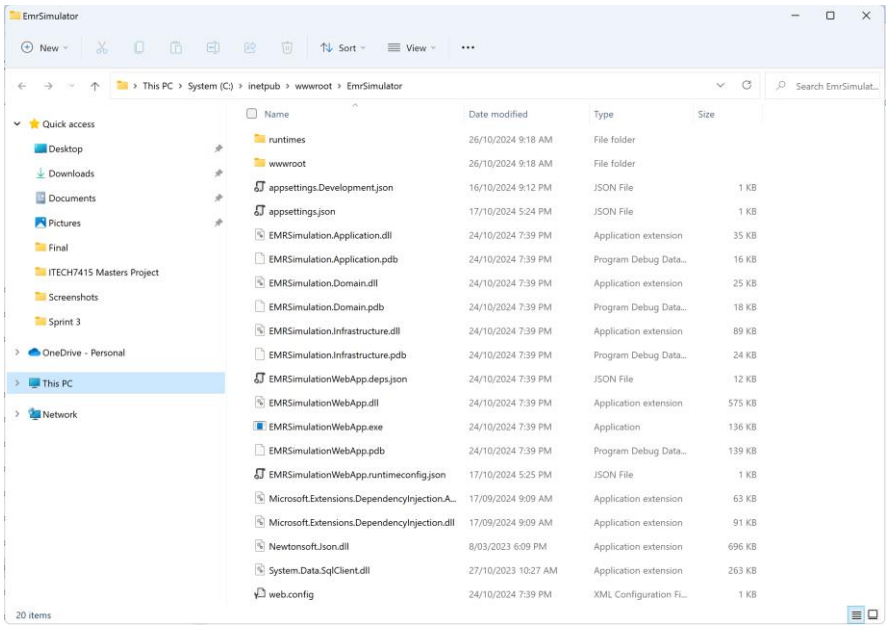
C:\inetpub\wwwroot\EmrSimulator

Other way to go to **C:\inetpub\wwwroot\EmrSimulator** folder

Open IIS, select the website named **EmrSimulator** and click **Explore**



EmrSimulator Folder



Copy all the files make sure the below **2** files have below connection string.

Make sure the user Id is **sa** and password is **Emrp@ssword123!**

As the above credentials were setup while installing the SQL Express.

The application uses these credentials to connect to EmrSimulator database.

appsettings.json

appsettings.Development.json

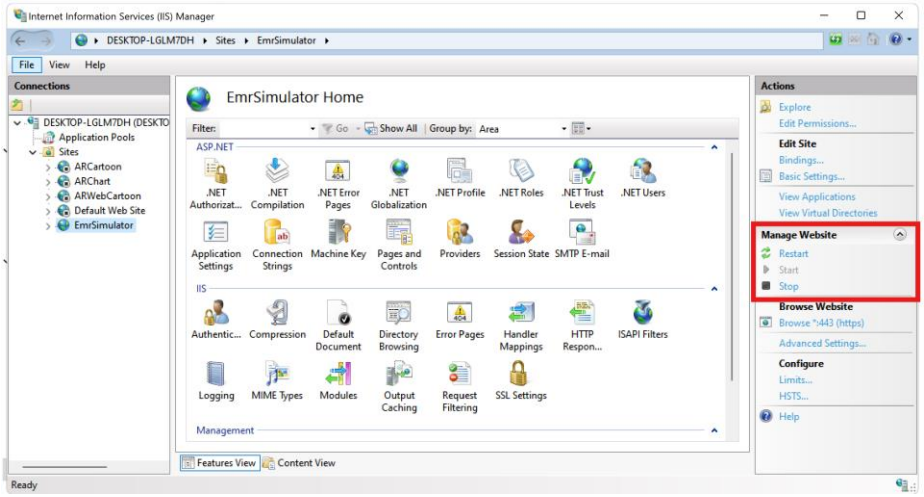
```
"ConnectionStrings": {
  "EmrSimulationConnection": "Server=.\\SQLEXPRESS;Database=EmrSimulator;User ID=sa;Password=Emrp@ssword123!;"
}
```

Note:

If the folder **C:\inetpub\wwwroot\EmrSimulator** already contains these 2 files and there has been no change in the connection string, then you can skip these 2 files and paste the rest files.

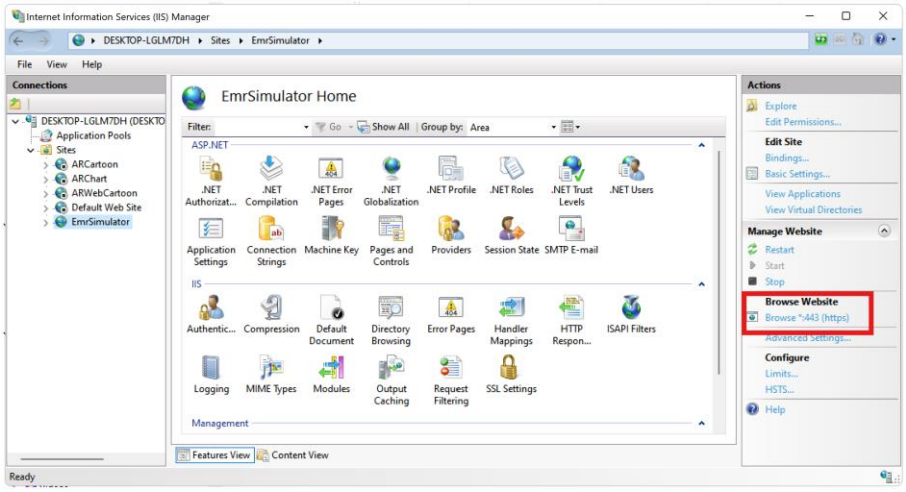
Start the website from IIS

After pasting the files, Start the website from IIS



Browse EmrSimulator application

Click on the link under **Browse Website** as highlighted



Login to EmrSimulator

Browse the application and try login as **Student** or **Supervisor** using login id and password provided.