PE\_PRN211\_SP24\_212204

**Please read the instructions carefully before doing the questions.**

* You are **NOT allowed** to use any other materials. You are **NOT allowed** to use any device to share data with others.
* You must use IDE as **Visual Studio 2019 or later, MSSQL Server 2016 or** **later database** for your development tools.

**IMPORTANT – Before you start doing your solution, MUST do the following steps:**

1. To do your program, you must use **Windows Forms**, apply 3-Layer architecture. *Note that* *you are not allowed to connect directly to database from Windows Forms, every database connection must be used with Repository and Data Access Objects. The database connection string must get from appsettings.json file.* ***In the case your program connects directly to database from WindowsForms or you hardcode connection string, you will get 0 mark.***
2. ***If there are syntax errors or compilation errors in your PE program, you will not pass the PE requirements, the mark will be 0.***
3. Create Solution in Visual Studio 2019/2022 named **PE\_PRN211\_SP24\_212204\_StudentName**. Inside the Solution, Project Windows Forms named: **ConvenienceStore\_StudentName.**
4. Create your MS SQL database named **ConvenienceStoreDb** by running code in script **ConvenienceStoreDb.sql.**
5. Set the default user interface for your project as **Login** form.

**REFERENCES *(this session just for reference, student can use the other approach to do Practical Exam)***

***Working with DB connection string from JSON file****.*

* 1. In the Presentation layer (WindowsForms Project), you create *appsettings.json*and add ConnectionString same as the bellow to config the connection string to SQL Server Database.

*{*

*"ConnectionStrings": {*

*"DefaultConnectionStringDB": "server =(local); database=***ConvenienceStoreDb***;uid=****sa****;pwd=****1234567890****; TrustServerCertificate=True"*

*}*

*}*

You can change **server**, **uid** and **pwd** to suitable with your local machine.

* 1. Set property "Copy to output Directory" of *appsettings.json* file to "Copy if newer"
  2. Using Manage Nuget packages to install packages the following in Windows Form Project

*Microsoft.Extensions.Configuration -Version 5.0.0*

*Microsoft.Extensions.Configuration.Json -Version 5.0.0*

* 1. Using *ConfigurationBuilder* to init Configuration object for reading *appsettings.json* file same as this code:

*private string GetConnectionString()*

*{*

*IConfiguration config = new ConfigurationBuilder()*

*.SetBasePath(Directory.GetCurrentDirectory())*

*.AddJsonFile("appsettings.json",true,true)*

*.Build();*

*var strConn = config["ConnectionStrings:DefaultConnectionStringDB"];*

*return strConn;*

*}*

* 1. After that, durring development, student can bypass the ConnectionString (which read from *appsettings.json*) to Data access layer by constructor or others

*public partial class* ***ConvenienceStoreDbContext****: DbContext*

*{*

*public ConvenienceStoreDbContext (string connectionString)*

*{*

*this.Database.SetConnectionString(connectionString);*

*}*

*}*

***Package using for .NET:***

|  |  |  |
| --- | --- | --- |
|  | *Microsoft.EntityFrameworkCore.SqlServer version* | *Microsoft.Extensions.Configuration, Microsoft.Extensions.Configuration.Json version* |
| *.NET 5* | *5.0.17* | *5.0.0* |
| *.NET 6* | *6.0.27* | *6.0.1/6.0.0* |
| *.NET 7* | *7.0.16* | *7.0.0* |
| *.NET 8* | *8.0.2* | *8.0.0* |

*- Install package using Tools → NuGet Package Manager → Package Manager Console*

Install-Package Microsoft.EntityFrameworkCore.SqlServer -Version 5.0.17

Install-Package Microsoft.EntityFrameworkCore.Tools -Version 5.0.17

Install-Package Microsoft.EntityFrameworkCore.Design -Version 5.0.17

Install-Package Microsoft.Data.SqlClient -Version 3.0.1

Install-Package Microsoft.Extensions.Configuration -Version 5.0.0

Install-Package Microsoft.Extensions.Configuration.Json -Version 5.0.0

- *Install package using CLI or Power Shell*

dotnet add package Microsoft.EntityFrameworkCore.SqlServer --version 5.0.17

dotnet add package Microsoft.EntityFrameworkCore.Design --version 5.0.17

dotnet add package Microsoft.EntityFrameworkCore.Tools --version 5.0.17

dotnet add package Microsoft.Data.SqlClient --version 3.0.1

dotnet add package Microsoft.Extensions.Configuration --version 5.0.0

dotnet add package Microsoft.Extensions.Configuration.Json --version 5.0.0

Entity Framework Core

*- Install dotnet-ef for CLI*

dotnet tool install --global dotnet-ef --version 5.0.11

*- Use Entity Framework Core to generate Object Model from existing database – CLI*

dotnet ef dbcontext scaffold "Server=(local);uid=**sa**;pwd=**1234567890**;database=**ConvenienceStoreDb**;TrustServerCertificate=True" Microsoft.EntityFrameworkCore.SqlServer --output-dir Models

*- Generate database from domain classes – CLI.*

dotnet ef migrations add "InitialDB"

dotnet ef database update

Entity Framework Core

*- Use Entity Framework Core to generate Object Model from existing database – Package Manager Console*

Scaffold-DbContext "Server=(local);uid=**sa**;pwd=**1234567890**;database=**ConvenienceStoreDb**;TrustServerCertificate=True;" Microsoft.EntityFrameworkCore.SqlServer -OutputDir Models

*- Generate database from domain classes – Package Manager Console*

Add-Migration "InitialDB"

Update-Database -verbose