

---

## INSTRUCTIONS

**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 **Visual Studio 2019 or above, MSSQL Server 2008 or above** for your development tools.

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

1. Create Solution in Visual Studio named **PE\_PRN222\_SP25\_TrialTest\_StudentName**. Inside your Solution, the Project ASP.NET Core Razor Page must be named: **PharmaceuticalManagement\_StudentName**.
  2. To do your program, you must use **ASP.NET Core Web App (Razor Pages)**, apply **3-Layer architecture** (Presentation Layer, Business Logic Layer, Data Access Layer), there are at least 2 Projects for the Solution. *The database connection string must get from the appsettings.json file.*  
*In the case your code connects direct to the database from ASP.NET Core Web App (Razor Pages) or hard coded the connection string, you will get 0 point.*
  3. Create your MS SQL database named **Sp25PharmaceuticalDB** by running code in script **Sp25PharmaceuticalDB.sql**.
  4. Set the default user interface for your project as a **Login** page.
  5. *If there are syntax errors or compilation errors in your PE program, you will not pass the PE requirements, the mark will be 0.*
  6. *Your work will be considered invalid (0 point) if your code inserts stuff that is unrelated to the test.*
-

Create an application using **ASP.NET Core Razor Pages** with .NET Core (.NET 8/.NET 9), C#, and Entity Framework Core. A MS SQL Server database will be created to persist the data and it will be used for reading and managing data.

A part of **Sp25PharmaceuticalDB** database.

*Store account role: Administrator = 1; Manager = 2; Staff = 3; Member = 4*

1. (1.00 point) Authentication function. *Using EmailAddress and StoreAccountPassword for the authentication/authorization process.*

If the user with a *Manager Role or Staff Role* logs in successfully, save this information to a temporary parameter. In case the login is unsuccessful, display the message “*You do not have permission to do this function!*”.

- No permissions for the function: List all of items.
- All Create-Update-Delete and Search functions are required with *Manager role*,
- *Staff role* is required for Search actions.

2. (1.5 points) List all of items in **MedicineInformation** table (*each record includes information of MedicineInformation and ManufacturerName*). Paging is required for this function, 3 records/1page. (Note that, No permissions for this function).

3. (2.5 points) If a user with a *Manager role* logs in successfully, add a new item with the *requirements*:
- *The new item will display at the top of the list.*
  - All fields are required.
  - Value for *ActiveIngredients* is greater than 10 characters. Each word of the *ActiveIngredients* must begin with the capital letter or a number (0, 1, 2 →9). The value of *ActiveIngredients* is not allowed with special characters such as #, @, &, (, ).
  - The *ManufacturerID/ManufacturerName* will come from the ***Manufacturer*** table (you can use ComboBox UI control in this case)
4. (2.0 points) If a user with *Manger role* logs in successfully, update the selected existing item.
- With update function, the *same validation requirements with Add* function.
5. (1.50 points) If a user with *Manager role* logs in successfully, search all items with 3 conditions: *ActiveIngredients*, *ExpirationDate*, *WarningsAndPrecautions* (OR option for this case and using relative search). ***Search results need to be grouped by ActiveIngredients, ExpirationDate, WarningsAndPrecautions.***
6. (1.50 points) Using realtime communication with SignalR for delete function. If a user with *Manager role* logs in successfully, delete the selected item with the confirmation then update the list of items for all connected clients.