PE_PRN231_FALL22_01
FALL 22
Subject: PRN231
Duration: 90 minutes

INSTRUCTIONS

Please read the instructions carefully before doing the questions.

- 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. *Please check before you submit your work, no syntax error(s) in the submitted PE.*
2. *You are not allow to connect direct to database from ASP.NET Core Web API, every database connection must be used with Repository and Data Access Objects. The database connection string must get from appsettings.json file.*
3. Create Solution/Project in Visual Studio named **PRN231PE_FA22_StudentCode**.
4. You must create MS SQL database with the name **PRN231PE_FA22_StudentCode.**
5. Set the default user interface for your Client Project as **Login** page.

Implement a part of RealEstate Management System. The situation is described as a company type has many property profile and a property profile belongs to only one company.

A part of *RealEstate Management* database

HRStaff(EmailAddress, Fullname, Password, Role)

Role: Administrator = 1; Manager = 2; Staff = 3

Company(CompanyID, CompanyName, Address, TelephoneNumber, ContactName)

All fields are required.

Renting(CompanyID, PropertyProfileID, StartDate, EndDate)

All fields are required.

PropertyProfile(PropertyProfileID, Name, ShortDescription, Price, Location, PropertyDescription, Status)

All fields are required. Status = Active/Inactive

Part 01 (1.5 marks). Using Forward Engineering approach (Code First) create related classes then migrate to database named **PRN231PE_FA22_StudentCode** (MS SQL Server).

Part 01 (3.5 marks). Create an API using **ASP.NET Core Web API** with ADO.NET (or 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.

Using OData to implement the API

- Implement CRUD actions with **PropertyProfile** table.
- Get all renting information of **CompanyName**.
- Search all property profiles with 2 conditions: Name or Location (using relative search)

Part 02 (5.0 marks). Client application

Choose one of these options for client application

- *ASP.NET Core Web App with MVC*
- *ASP.NET Core Razor Pages*
- *Windows Forms*
- *Windows Presentation Foundation*

1. Authentication function

   If user with an *administrator role* logs in successfully, save this information to a temporary parameter. All CRUD actions are required with authentication. In the case login unsuccessfully, display *"You are not allowed to access this function!"*
2. Check if login successfully, search all properties with 2 conditions: Name or Location (OR option for this case and paging is required for this function).
3. Check if login successfully, delete the selected item with the confirmation then update the list of properties.
4. Check if login successfully, add new item with the r*equirements:*
   - All fields are required.
   - Value for Location includes a-z, A-Z, /, space and digit 0-9.
   - Value for Name is greater than 8 characters. Each word of the Name must begin with the capital letter.
5. Check if login successfully, list all renting information of **CompanyName** (paging is required for this function).