Assignment 02

Building a Sales Management Application with Windows Forms

Introduction

Imagine you're an employee of a product retailer named **FStore**. Your manager has asked you to develop a Windows Forms application for member management, product management, and order management. The application has a default account whose email is "**admin@fstore.com**" and password is "**admin@@**" that stored in the **appsettings.json**.

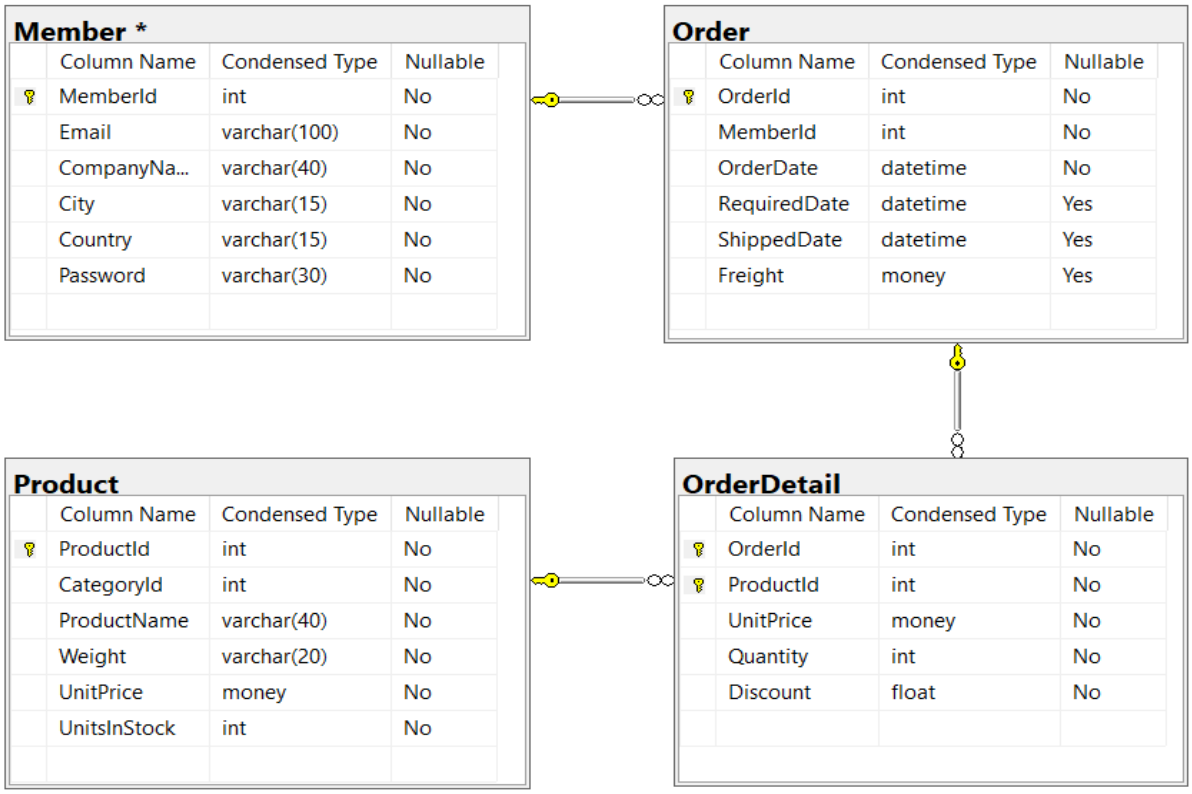
The application has to support adding, viewing, modifying, and removing products—a standardized usage action verbs better known as Create, Read, Update, Delete (CRUD) and Search. This assignment explores creating an application using Windows Forms with .NET Core, C#, and ADO.NET / Entity Framework. The MS SQL Server database will be created to persist the data and it will be used for reading and managing data.

Assignment Objectives

In this assignment, you will:

- Use the Visual Studio.NET to create Windows Forms and Class Library (.dll) projects.
- Develop MDI (Multiple Document Interface) application using WinForms.
- Perform CRUD actions using ADO.NET and Entity Framework Core
- Use LINQ to query and sort data
- Apply passing data in WinForms application
- Apply 3-layers architecture to develop an application
- Apply MPV (Model-Presenter-View) pattern in WinForms application
- Apply Repository pattern and Singleton pattern in a project
- Add CRUD and searching actions to WinForms application.
- Apply to validate data type for all fields
- Run the project and test the WinForms actions.

Database Design



Main Functions

- Member management, Product management, and Order management: Read, Create, Update and Delete actions. Creating and Updating actions must be performed by popup dialog
- Search Product by ID , ProductName (by keyword of ProductName), UnitPrice, and UnitInStock
- Create a report statistics sales by the period from StartDate to EndDate, and sort sales in descending order
- Member authentication by Email and Password. If the user is "**Admin**" then allows to perform all actions, otherwise, the normal user is allowed to view/create/update the profile and view their orders history.

Guidelines

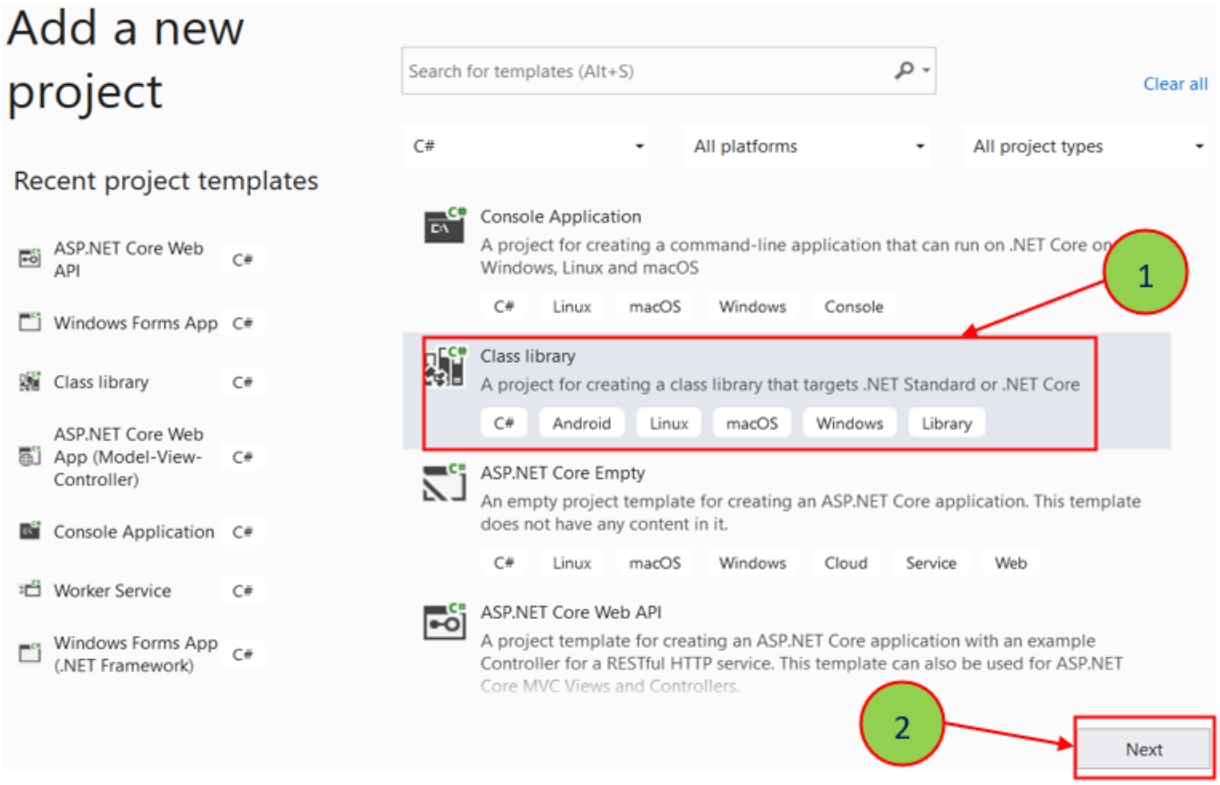
Activity 01: Build a solution [01 mark]

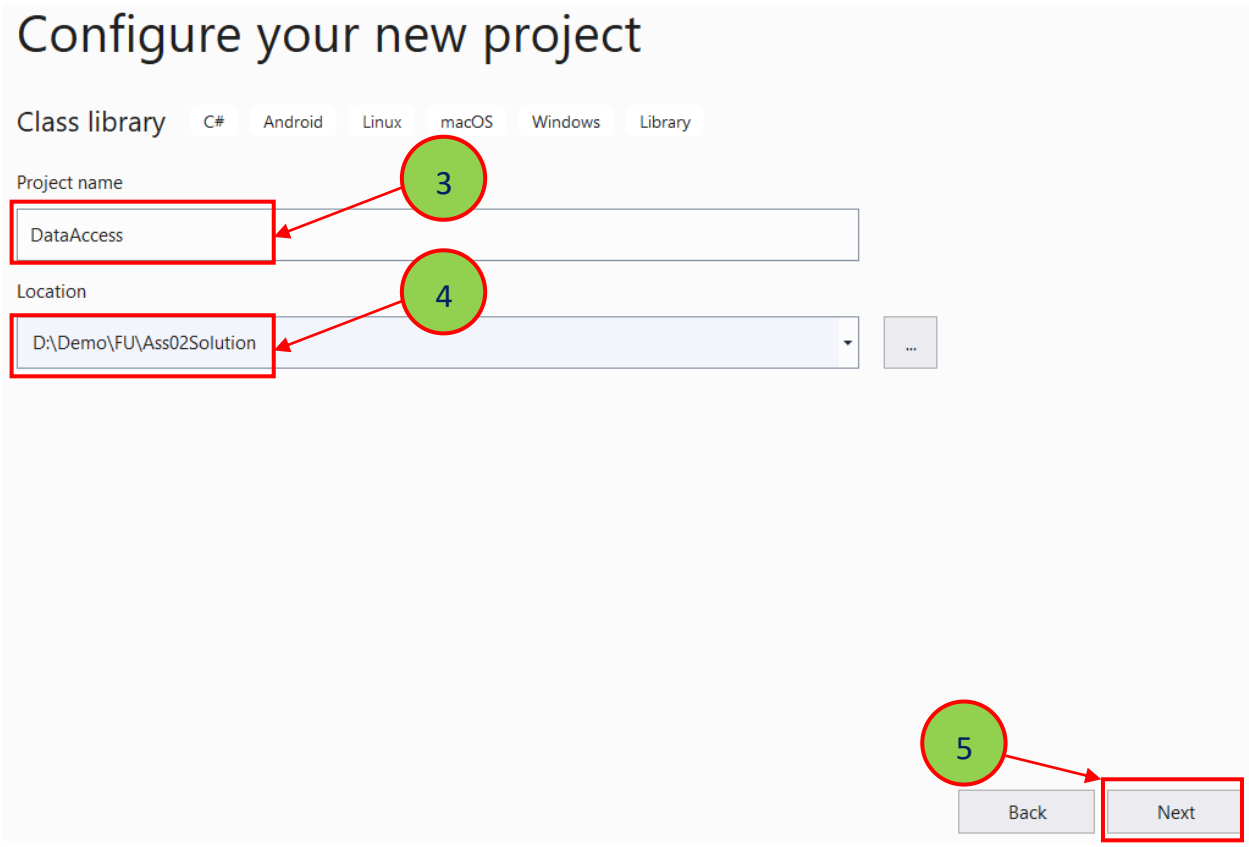
Create a Blank Solution named **Ass02Solution** that includes Class Library Project: **DataAccess, BusinessObject,** and a Windows Forms project named **SalesWinApp**

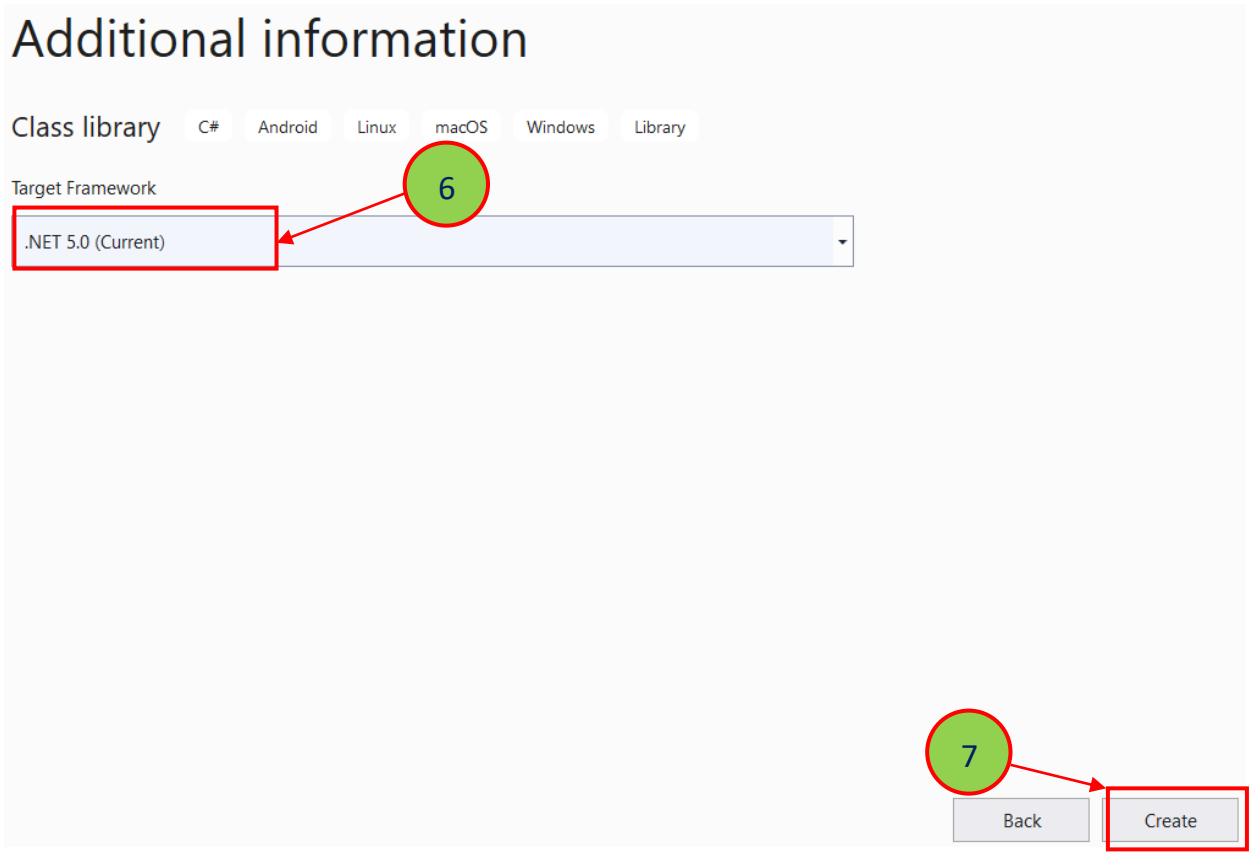
Step 01. Open the Visual Studio .NET application and create a Blank solution named **Asm02Solution**

Step 02. Create a Class Library project named **DataAccess**

From the File menu | Add | New Project, on the Add New Project dialog, select "Class Library" and performs steps as follows:


Add a new project
Search for templates (Alt+S)
Clear all
C#
All platforms
All project types
Recent project templates
ASP.NET Core Web API C#
Windows Forms App C#
Class library C#
ASP.NET Core Web App (Model-View-Controller) C#
Console Application C#
Worker Service C#
Windows Forms App (.NET Framework) C#
Console Application
A project for creating a command-line application that can run on .NET Core on Windows, Linux and macOS
C# Linux macOS Windows Console
Class library
A project for creating a class library that targets .NET Standard or .NET Core
C# Android Linux macOS Windows Library
ASP.NET Core Empty
An empty project template for creating an ASP.NET Core application. This template does not have any content in it.
C# Linux macOS Windows Cloud Service Web
ASP.NET Core Web API
A project template for creating an ASP.NET Core application with an example Controller for a RESTful HTTP service. This template can also be used for ASP.NET Core MVC Views and Controllers.
1
2
Next

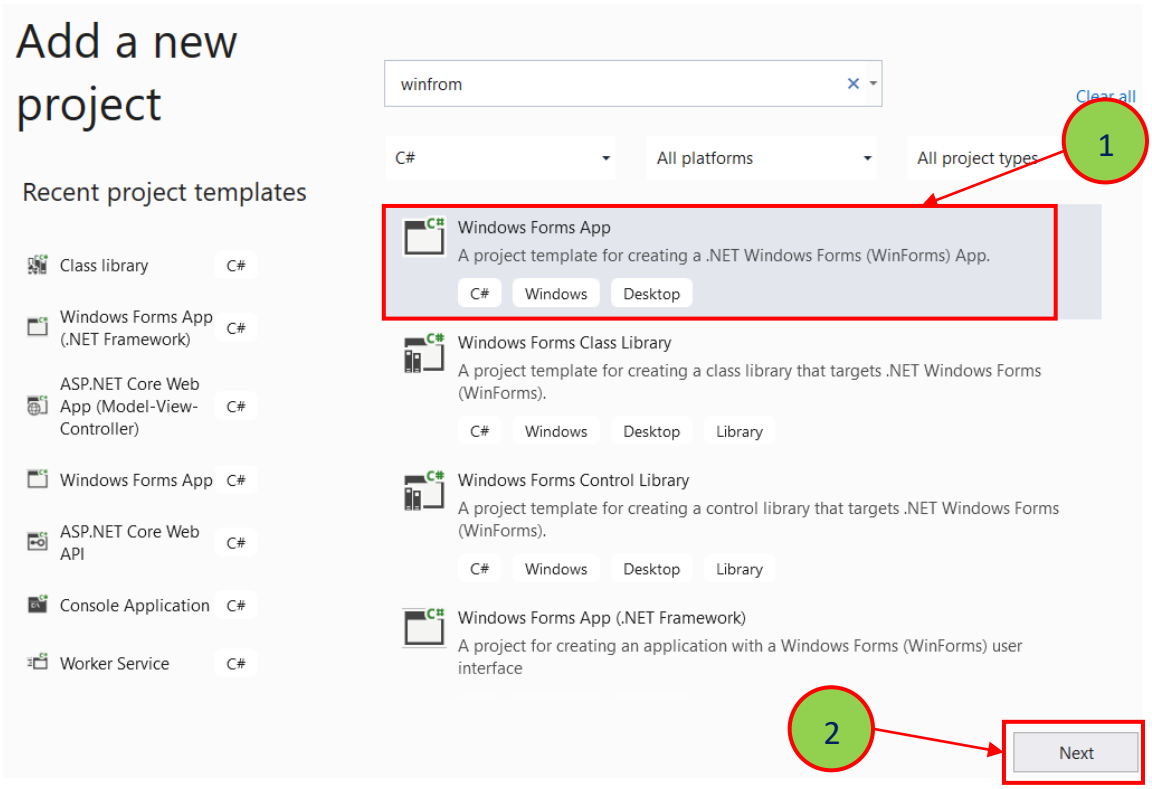

Configure your new project
Class library C# Android Linux macOS Windows Library
Project name
DataAccess
3
Location
D:\Demo\FU\Ass02Solution
4
...
5
Back
Next

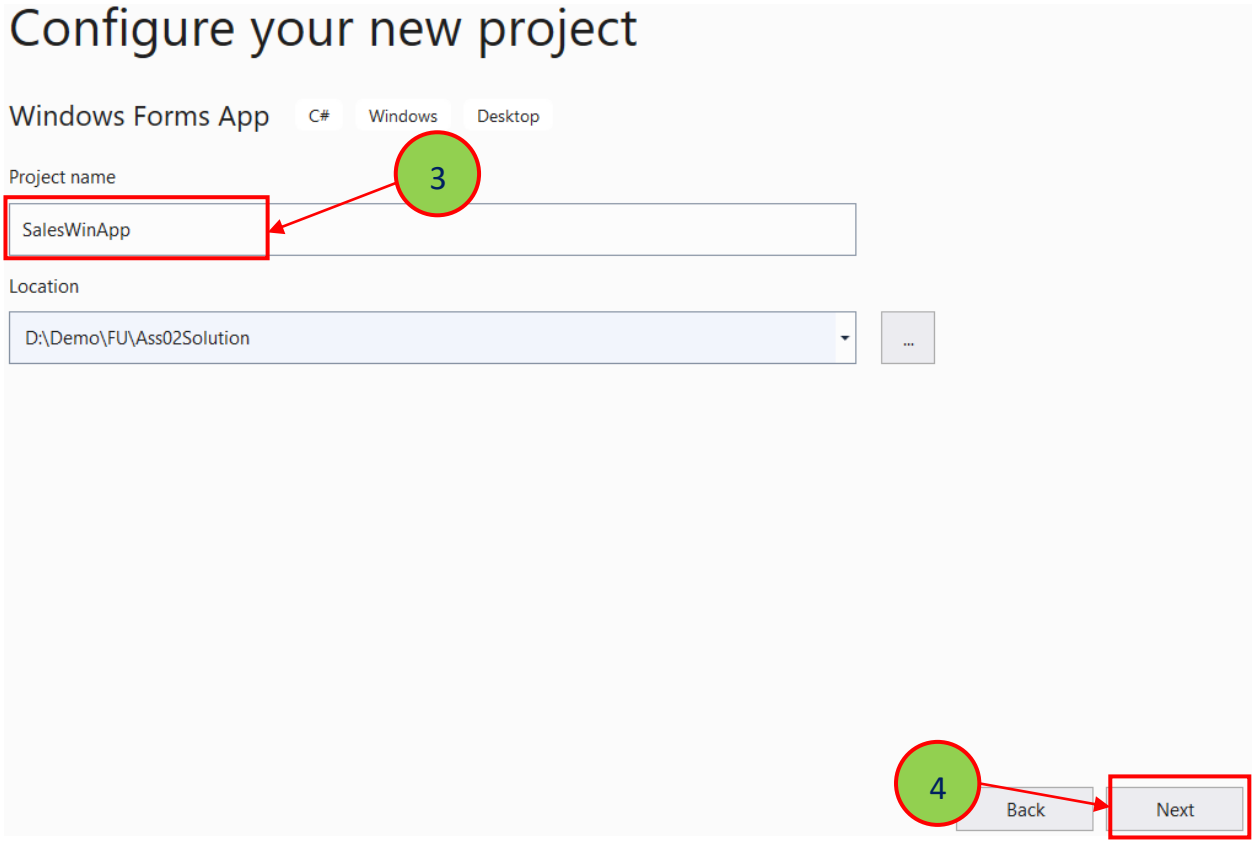


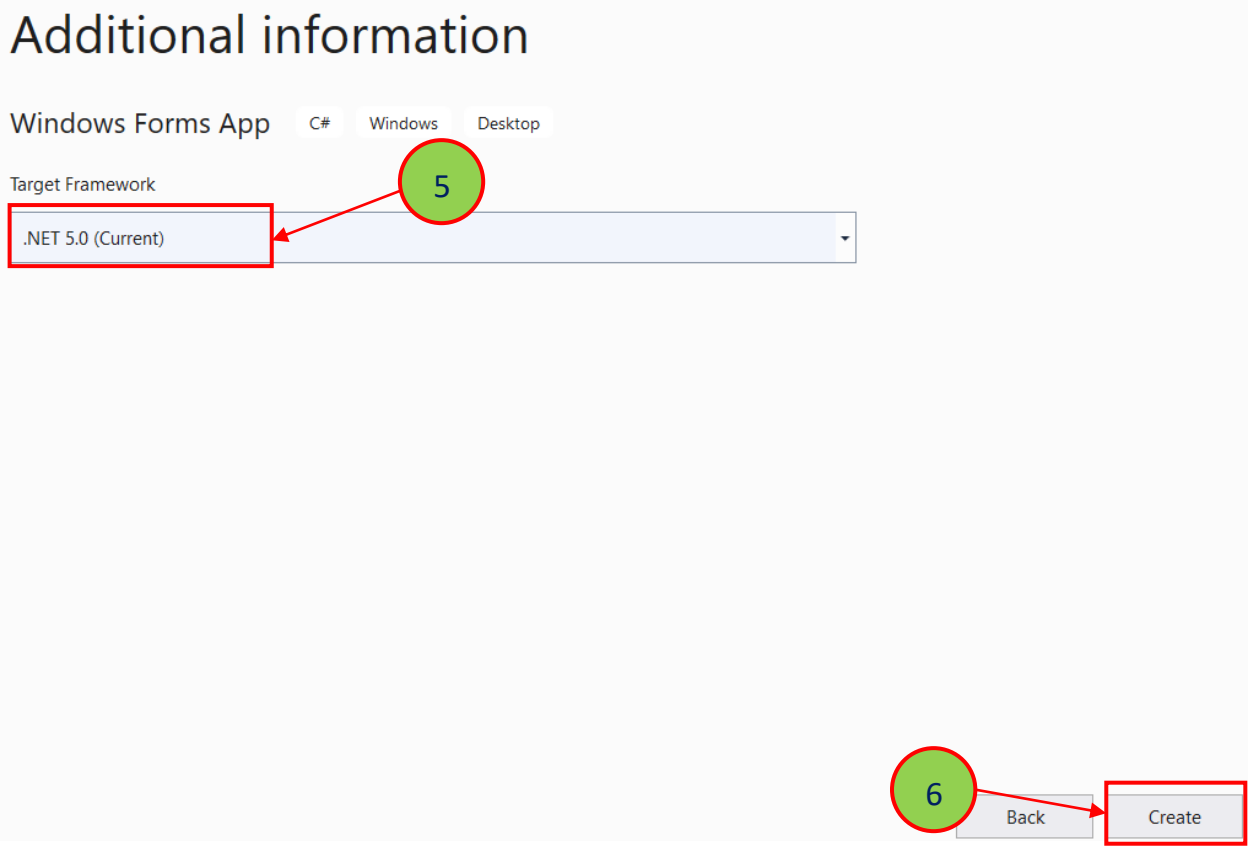
Step 03. Repeat **Step 02** to create a **BusinessObject** project.

Step 04. Create a Windows Forms project named **SalesWinApp**

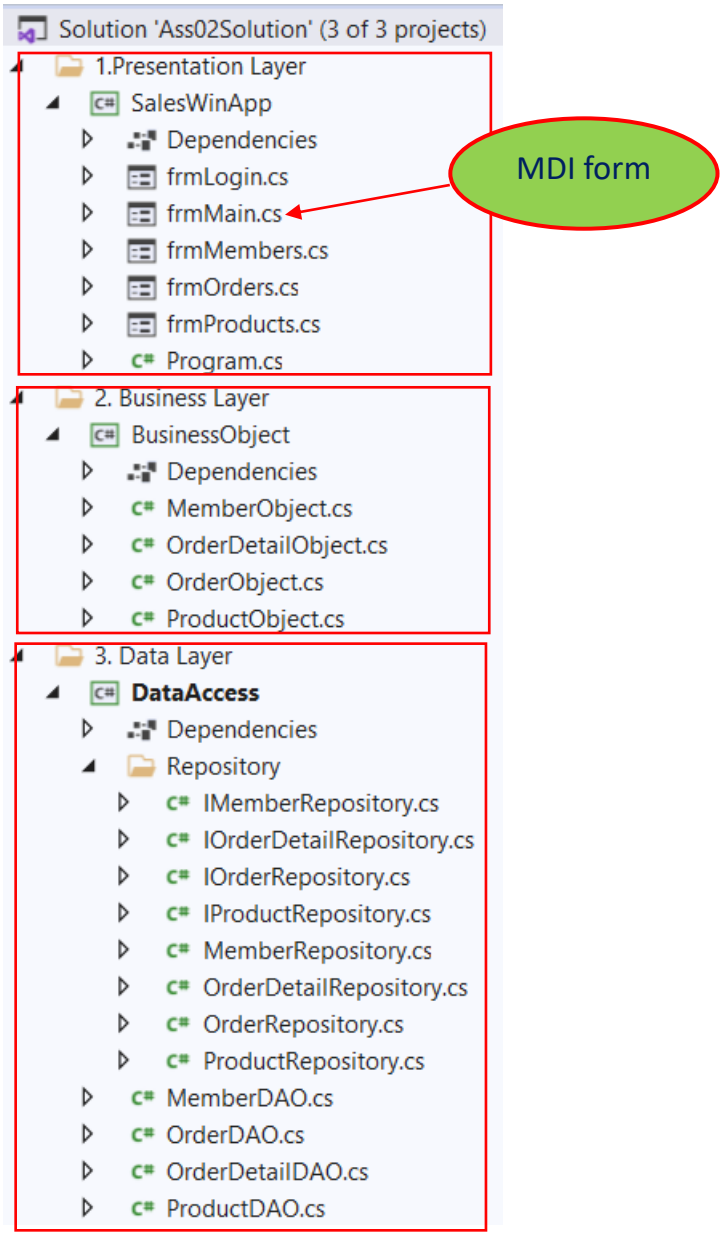
- From the File menu | Add | New Project, on the Add New Project dialog, select "Windows Forms App" and performs steps as follows:


Add a new project
winfrom
Clear all
C#
All platforms
All project types
Recent project templates
Class library C#
Windows Forms App (.NET Framework) C#
ASP.NET Core Web App (Model-View-Controller) C#
Windows Forms App C#
ASP.NET Core Web API C#
Console Application C#
Worker Service C#
Windows Forms App
A project template for creating a .NET Windows Forms (WinForms) App.
C# Windows Desktop
Windows Forms Class Library
A project template for creating a class library that targets .NET Windows Forms (WinForms).
C# Windows Desktop Library
Windows Forms Control Library
A project template for creating a control library that targets .NET Windows Forms (WinForms).
C# Windows Desktop Library
Windows Forms App (.NET Framework)
A project for creating an application with a Windows Forms (WinForms) user interface
1
2
Next


Configure your new project
Windows Forms App C# Windows Desktop
Project name
SalesWinApp
3
Location
D:\Demo\FU\Ass02Solution
...
4
Back
Next



Step 05. Create folders and add classes to the projects as follows:



Activity 02: Develop BusinessObject project [02 marks]

Step 01. Write codes to create classes and definition all data members

Step 02. Write codes to perform business rules for data members

Activity 03: Develop DataAccess project [03 marks]

Step 01. Add the project reference to the **BusinessObject** project

Step 02. Write codes for **MemberDAO.cs, IMemberRepository.cs** and **MemberRepository.cs**

Step 03. Write codes for **ProductDAO.cs, IProductRepository.cs** and **ProductRepository.cs**

Step 04. Write codes for **OrderDAO.cs, IOrderRepository.cs** and **OrderRepository.cs**

Step 05. Write codes for **OrderDetailDAO.cs, IOrderDetailRepository.cs** and **OrderDetailRepository.cs**

Hints*: *If using Entity Framework, you can install the AutoMapper package from Nuget to map Entity with Busines Object.

Activity 04: Develop SalesWinApp project [03 marks]

Step 01. Add the project reference to **BusinessObject** and **DataAccess** projects

Step 02. Design UI forms and write codes to perform functions

Hints: *Use MenuStrip, ToolStrip, StatusStrip on the MDI form (frmMain form)*

Activity 05: Run the WinForms project and test all actions [01 mark]