

# **МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ**

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. ТАРАСА ШЕВЧЕНКА**

**ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**КАФЕДРА МЕРЕЖЕВИХ ТА ІНТЕРНЕТ ТЕХНОЛОГІЙ**

## **СУЧАСНІ ІНТЕРНЕТ ТЕХНОЛОГІЇ**

**ЛАБОРАТОРНЕ ЗАНЯТТЯ № 1 : ”СТРУКТУРА ПРОЕКТУ ASP.NET CORE  
ВЕБ-ЗАСТОСУНКУ. РОЗДІЛЕННЯ МОНОЛІТУ НА МОДУЛІ (ШАРИ) “**

**Виконав студент групи : МІТ-41**

**Кухарчук Богдан Петрович**

## Хід виконання роботи:

### 1.1. Початок роботи з ASP.NET Core.

У середовищі розробки Microsoft Visual Studio було здійснено створення нового проекту . Для цього було використано шаблон "ASP.NET Core Web App (Model-View-Controller)".

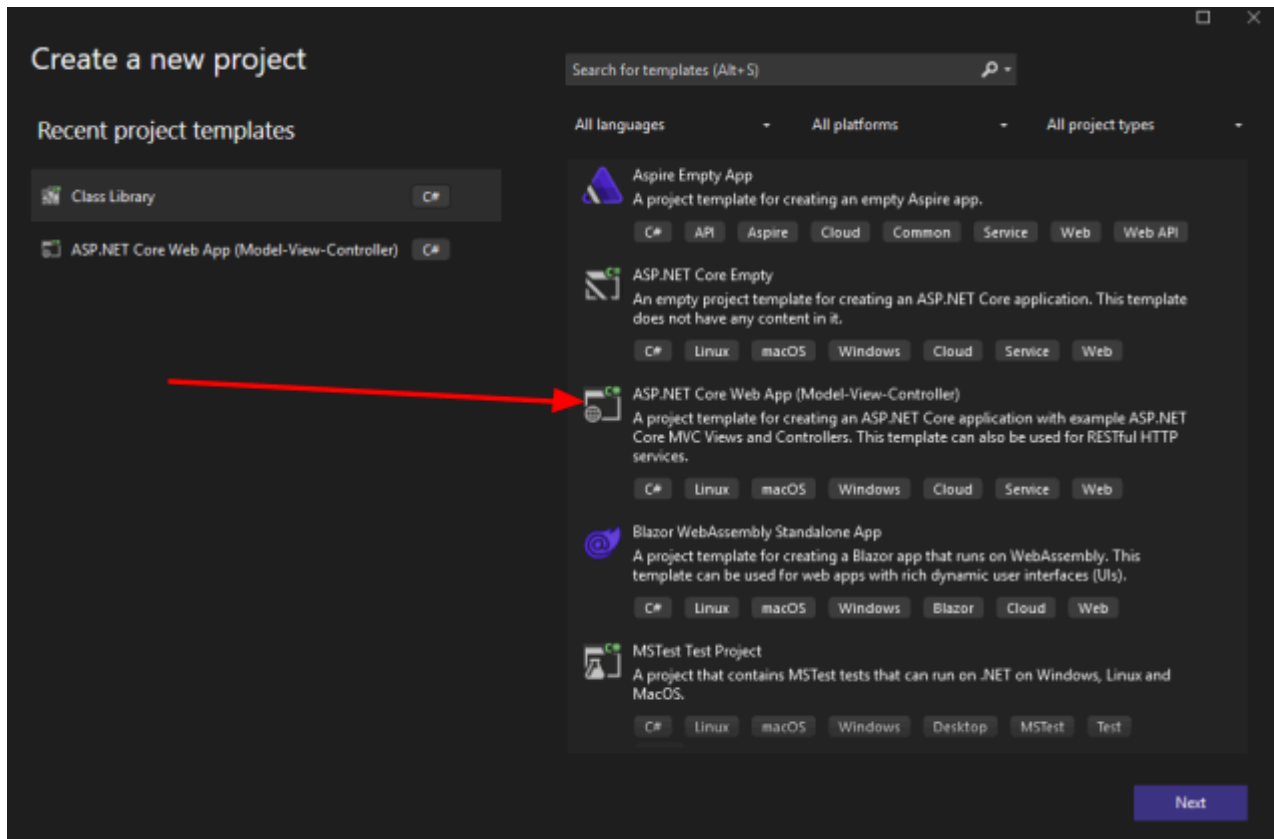


Рис. 1.1 Вибір середовища з яким будемо працювати

Наступним кроком потрібно було обрати тип аутентифікації Individual Accounts. Для цього після того як ми обрали середовище, з'явиться інше вікно, де ми зможемо обрати те що нам потрібне.

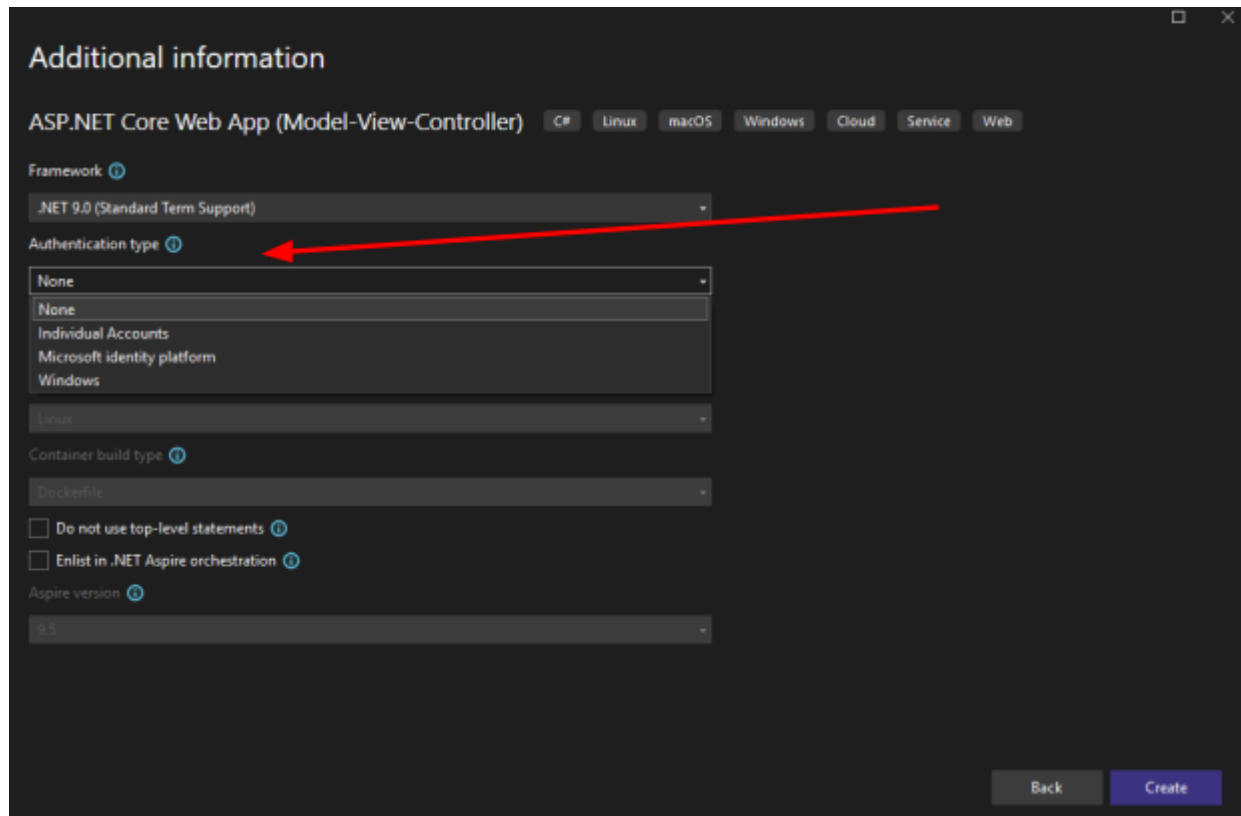


Рис. 1.2 Вибір типу аутентифікації

## 1.2. Розділення моноліту на модулі (шари)

Для реалізації цього завдання (розділення застосунку на модулі) було створено новий проєкт під назвою "WebApplicationData".

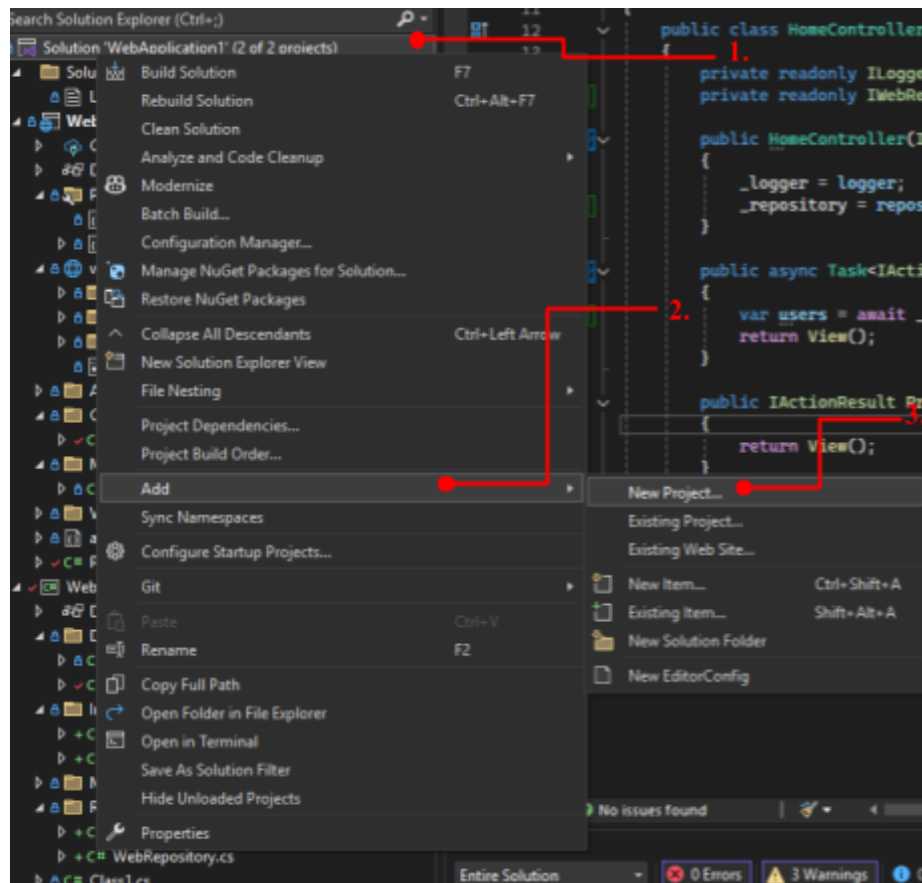


Рис. 2.1 Додавання нового проекту

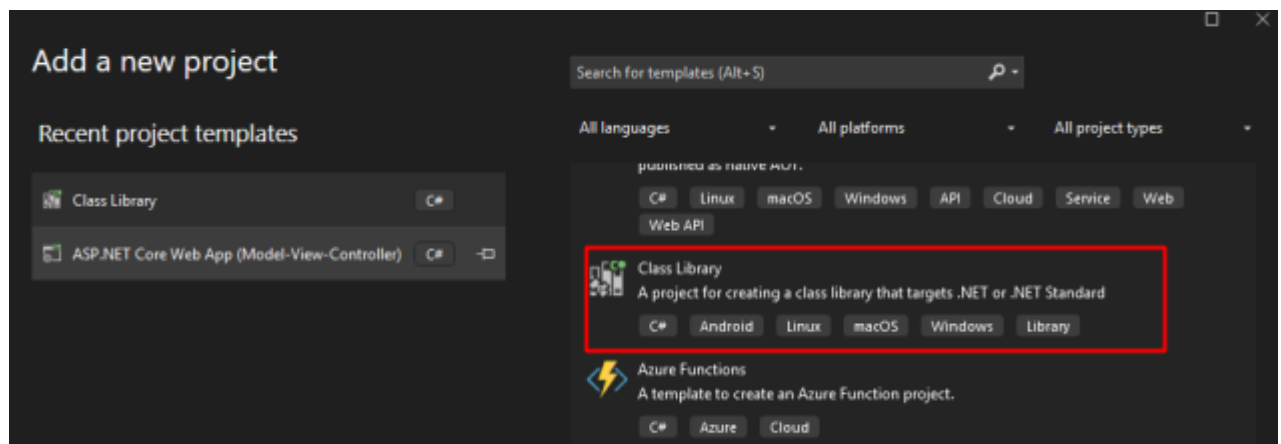


Рис. 2.2 Додавання нового модуля у вигляді бібліотеки класів

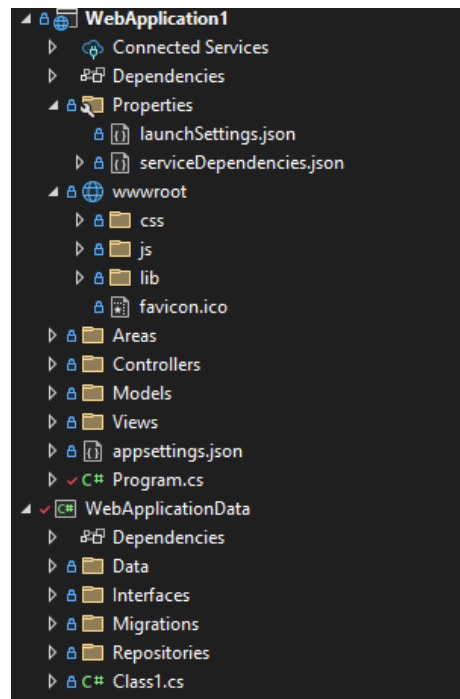


Рис. 2.3 - Розділений застосунок на модулі

Після розділення застосунку було помічено появу нового модуля **WebApplicationData** на панелі проекту (Рис. 2.3). У цей модуль необхідно було перенести компонент **ApplicationDbContext** з основного модуля **WebApplication1**. З цією метою у новому модулі була створена окрема папка **Data**, у якій створено файл **WebApplicationDbContext**, після чого весь відповідний вміст було перенесено до нього.

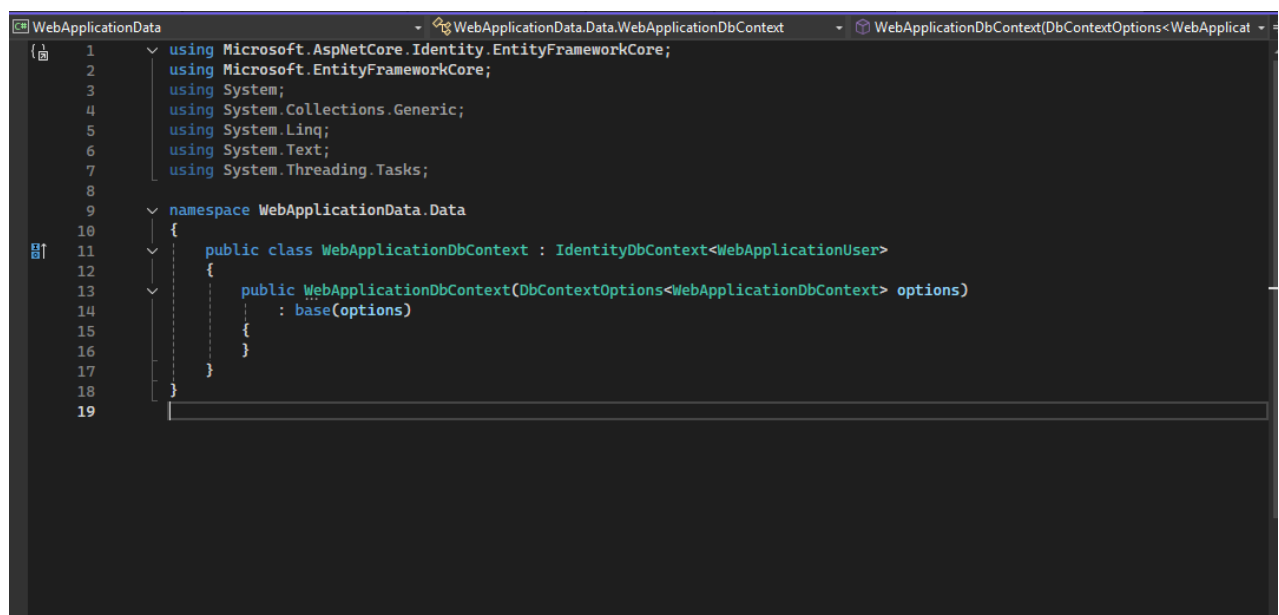


Рис. 2.4 Демонстрація вмісту класу WebApplicationDbContext

### 1.3 Розширення класу автентифікованого користувача IdentityUser

Також варто зазначити, що одним із завдань лабораторної роботи було розширення класу автентифікованого користувача **IdentityUser** шляхом додавання щонайменше двох нових властивостей — **FirstName** та **LastName**. Вказане завдання було успішно реалізовано та інтегровано до проєкту.

```
1  using Microsoft.AspNetCore.Identity;
2  using System;
3  using System.Collections.Generic;
4  using System.Linq;
5  using System.Text;
6  using System.Threading.Tasks;
7
8  namespace WebApplicationData.Data
9  {
10     public class ApplicationUser: IdentityUser
11     {
12         public string FirstName { get; set; }
13         public string LastName { get; set; }
14     }
15 }
16
17
```

Рис. 3.1 Демонстрація класу IdentityUser

Дуже важливим етапом під час виконання роботи є додавання необхідних бібліотек до проєкту. Для цього було відкрито менеджер пакетів **NuGet**, за допомогою якого здійснено пошук та встановлення бібліотек, необхідних для подальшої роботи застосунку.

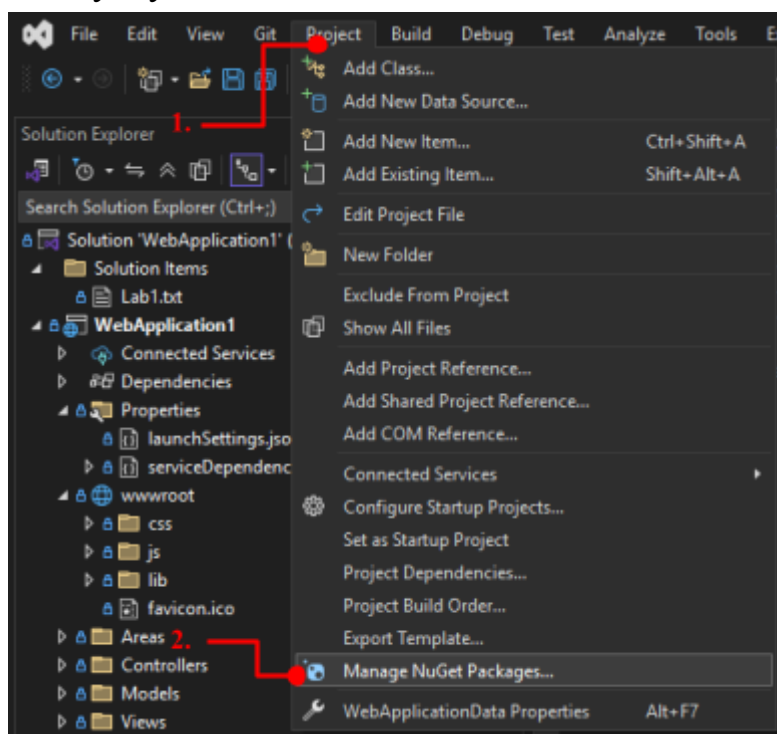


Рис. 3.2 Покрокове відкриття менеджера пакетів NuGet у середовищі розробки Visual Studio

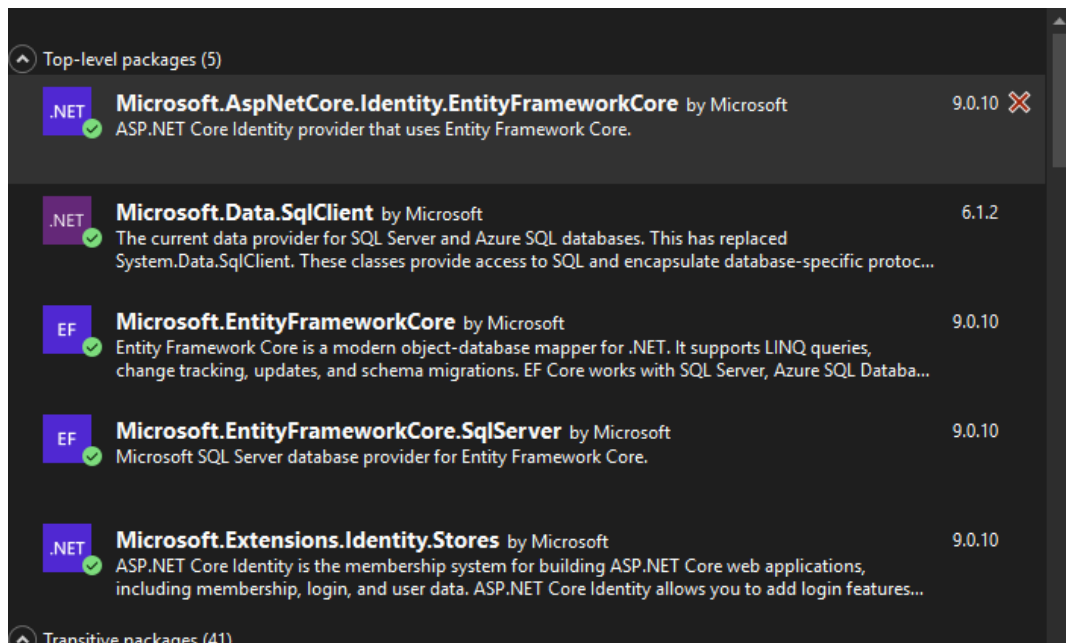


Рис. 3.3 Встановлення необхідних бібліотек

Наступним кроком є встановлення посилання на новостворений проєкт, що необхідно для забезпечення взаємодії між основним застосунком та додатковими модулями. Процес додавання посилання продемонстровано на Рис. 3.4.

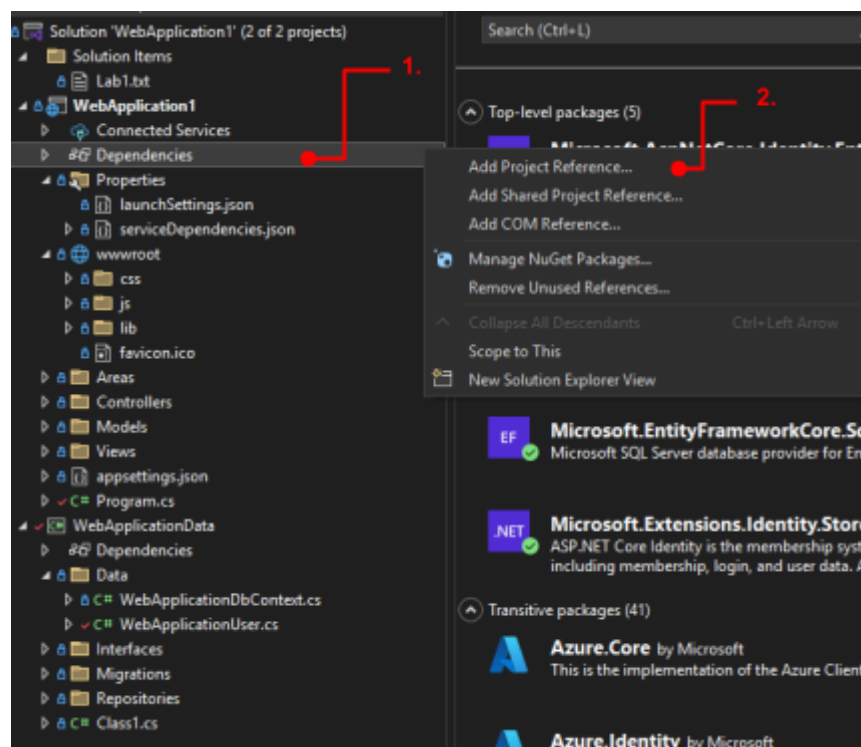


Рис. 3.4 Додавання посилання на новостворений проєкт

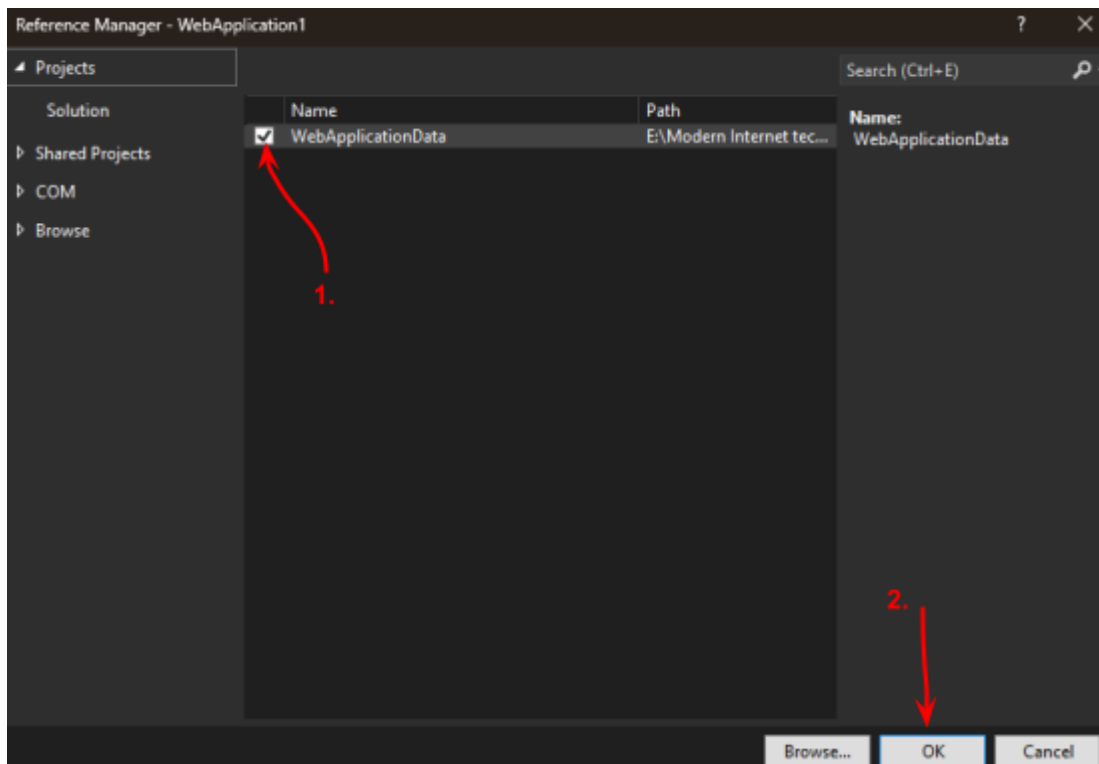


Рис. 3.5 Вибір опції для додавання посилання на проєкт

#### 1.4 Виконання міграцій, перевірка та оновлення структури таблиць

На цьому етапі було створено міграцію та виконано оновлення бази даних. Після цього здійснено перевірку таблиць, щоб упевнитися, що міграція та оновлення бази даних пройшли успішно і структура таблиць відповідає очікуваній моделі даних проєкту.



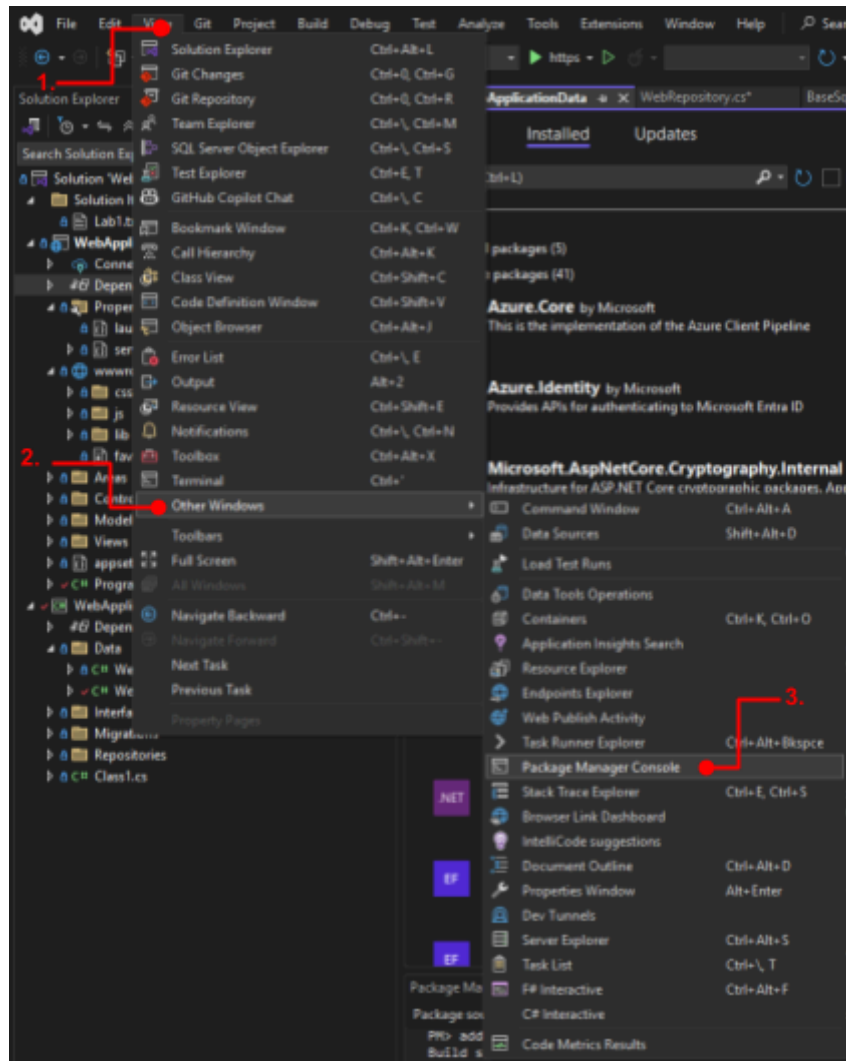


Рис.4.1 Розгашування Package Manager Console

```
Package Manager Console
Package source: All
Default project: WebApplicationData

PM> add-migration testmigr 2.
Build started...
Build succeeded.
To undo this action, use Remove-Migration.
PM> update-database 3.
Build started...
Build succeeded.
Microsoft.EntityFrameworkCore.Database.Command[20181]
  Executed DbCommand (13ms) [Parameters=[], CommandType='Text', CommandTimeout='30']
  SELECT 1
Microsoft.EntityFrameworkCore.Migrations[20411]
  Acquiring an exclusive lock for migration application. See https://aka.ms/efcore-docs-migrations-lock for more information if this takes too long.
  Acquiring an exclusive lock for migration application. See https://aka.ms/efcore-docs-migrations-lock for more information if this takes too long.
Microsoft.EntityFrameworkCore.Database.Command[20181]
  Executed DbCommand (16ms) [Parameters=[], CommandType='Text', CommandTimeout='30']
  DECLARE @result int;
  EXEC @result = sp_getapplock @resource = '___EFMigrationsLock', @LockOwner = 'Session', @LockMode = 'Exclusive';
  SELECT @result
Microsoft.EntityFrameworkCore.Database.Command[20181]
  Executed DbCommand (2ms) [Parameters=[], CommandType='Text', CommandTimeout='30']
  IF OBJECT_ID(N'___EFMigrationsHistory') IS NULL
  BEGIN
    CREATE TABLE [___EFMigrationsHistory] (
      [MigrationId] nvarchar(150) NOT NULL,
      [ProductVersion] nvarchar(32) NOT NULL,
      CONSTRAINT [PK___EFMigrationsHistory] PRIMARY KEY ([MigrationId])
    );
  END;
Microsoft.EntityFrameworkCore.Database.Command[20181]
  Executed DbCommand (8ms) [Parameters=[], CommandType='Text', CommandTimeout='30']
  SELECT 1
Microsoft.EntityFrameworkCore.Database.Command[20181]
  Executed DbCommand (8ms) [Parameters=[], CommandType='Text', CommandTimeout='30']
  SELECT OBJECT_ID(N'___EFMigrationsHistory');
Microsoft.EntityFrameworkCore.Database.Command[20181]
  Executed DbCommand (1ms) [Parameters=[], CommandType='Text', CommandTimeout='30']
  SELECT [MigrationId], [ProductVersion]
  FROM [___EFMigrationsHistory]
  ORDER BY [MigrationId];
Microsoft.EntityFrameworkCore.Migrations[20402]
  Applying migration '20251016091835_123321'.
  Applying migration '20251016091835_123321'.
Microsoft.EntityFrameworkCore.Database.Command[20181]
  Executed DbCommand (3ms) [Parameters=[], CommandType='Text', CommandTimeout='30']
  INSERT INTO [___EFMigrationsHistory] ([MigrationId], [ProductVersion])
  VALUES (N'20251016091835_123321', N'9.0.0');
  Applying migration '20251016093645_lastmigr'.
Microsoft.EntityFrameworkCore.Migrations[20402]
  Applying migration '20251016093645_lastmigr'.
Microsoft.EntityFrameworkCore.Database.Command[20181]
  Executed DbCommand (19ms) [Parameters=[], CommandType='Text', CommandTimeout='30']
  ALTER TABLE [AspNetUsers] ADD [FirstName] nvarchar(max) NOT NULL DEFAULT N'';
Microsoft.EntityFrameworkCore.Database.Command[20181]
```

Рис 4.2 Вибір нового проекту за замовчуванням та введення команд для успішного виконання міграції

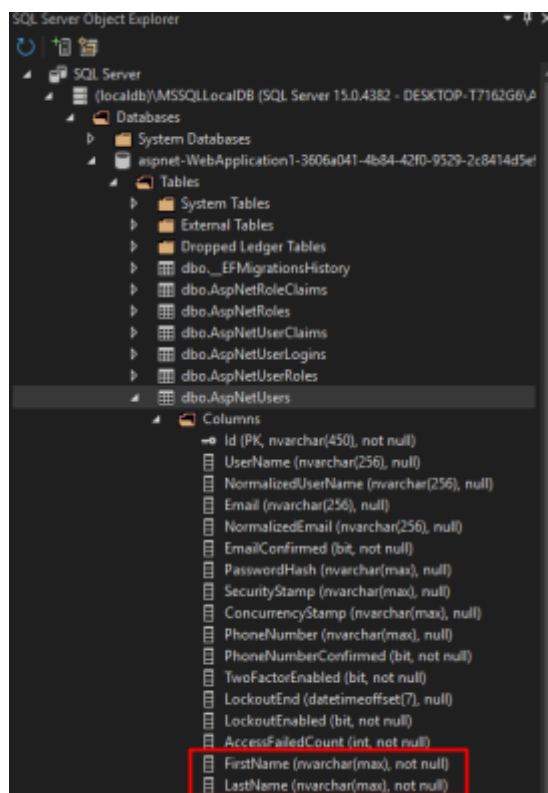


Рис 4.3 Підтвердження оновлення таблиці після виконання міграції бази даних

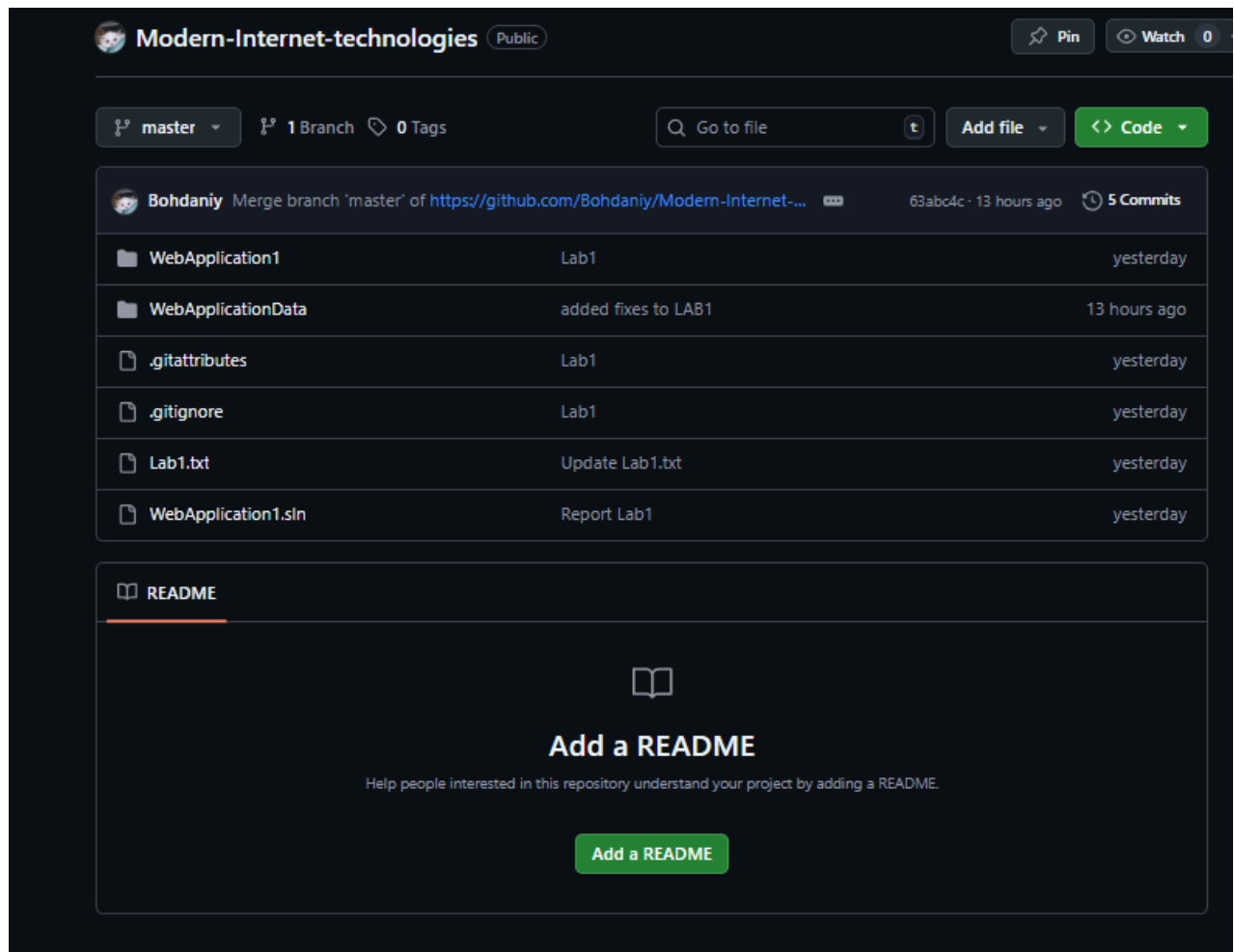


Рис 5.1 Створений репозиторій для лабораторних робіт

## Висновок

У ході виконання лабораторної роботи було розроблено повноцінний вебдодаток на платформі ASP.NET Core з реалізованою системою автентифікації та реєстрації. Для відповідності вимогам проєкту, стандартну модель користувача IdentityUser було розширено, додавши власні властивості FirstName та LastName, що дозволило зберігати персоналізовану інформацію.

З метою покращення архітектури та модульності застосунку, всю логіку доступу до даних, включно з контекстом ApplicationDbContext та моделями, було винесено в окремий проєкт-бібліотеку WebApplicationData. За допомогою інструментів Entity Framework Core було успішно згенеровано та застосовано міграції, що оновили схему бази даних відповідно до кастомізованої моделі користувача.

Весь проєкт було розміщено у віддаленому репозиторії на GitHub.