

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ  
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. ТАРАСА  
ШЕВЧЕНКА  
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
Кафедра мережевих та інтернет технологій  
СУЧАСНІ ІНТЕРНЕТ ТЕХНОЛОГІЇ  
РОБОТА З ДАНИМИ В ASP.NET CORE.  
РЕАЛІЗАЦІЯ ШАБЛОНУ REPOSITORY  
Лабораторне заняття №2  
Заяць Діани Юріївни

Хід виконання роботи:

### 1.1. Реалізація шаблону MVC та початок роботи в ASP.NET Core

Створюється новий проєкт у Microsoft Visual Studio на основі шаблону ASP.NET Core Web App (Model-View-Controller).

Під час налаштування вказується тип автентифікації Individual Accounts, що забезпечує індивідуальну реєстрацію користувачів.

The screenshot displays the 'Additional information' configuration window for an ASP.NET Core Web App (Model-View-Controller). The window has a dark theme. At the top, there are tabs for different project types: C#, Linux, macOS, Windows, Cloud, Service, and Web. The 'Web' tab is selected. Below the tabs, the 'Framework' is set to '.NET 8.0 (Long Term Support)'. The 'Authentication type' is set to 'Individual Accounts'. There are two checkboxes: 'Configure for HTTPS' (checked) and 'Enable container support' (unchecked). The 'Container OS' is set to 'Linux'. The 'Container build type' is set to 'Dockerfile'. There are two more checkboxes: 'Do not use top-level statements' (unchecked) and 'Enlist in .NET Aspire orchestration' (unchecked). The 'Aspire version' is set to '9.0'.

Рис. 1.1.1 – Вибір типу автентифікації для ASP.NET Core застосунку

## 1.2. Розділення ASP.NET Core застосунку на модулі

Щоб структурувати застосунок і забезпечити розділення відповідальностей, контекст даних і моделі переносяться до окремого проєкту в межах одного рішення.

Для цього в рішенні створюється новий проєкт типу Class Library (наприклад, SlayLib), до якого переміщуються папки Models та Data з основного веб-проєкту (WebAppCore).

Далі необхідно скоригувати простори імен (namespaces) у переміщених класах відповідно до нового розташування файлів.

Наприклад, для класу ApplicationDbContext, що розташований у папці Data нового модуля, належним простором імен буде SlayLib.Data.

Через NuGet встановлюються всі потрібні бібліотеки, після чого новий модуль підключається до основного веб-проєкту за допомогою Project Reference у розділі Dependencies.

Це забезпечує доступ до моделей і контексту даних із веб-застосунку.

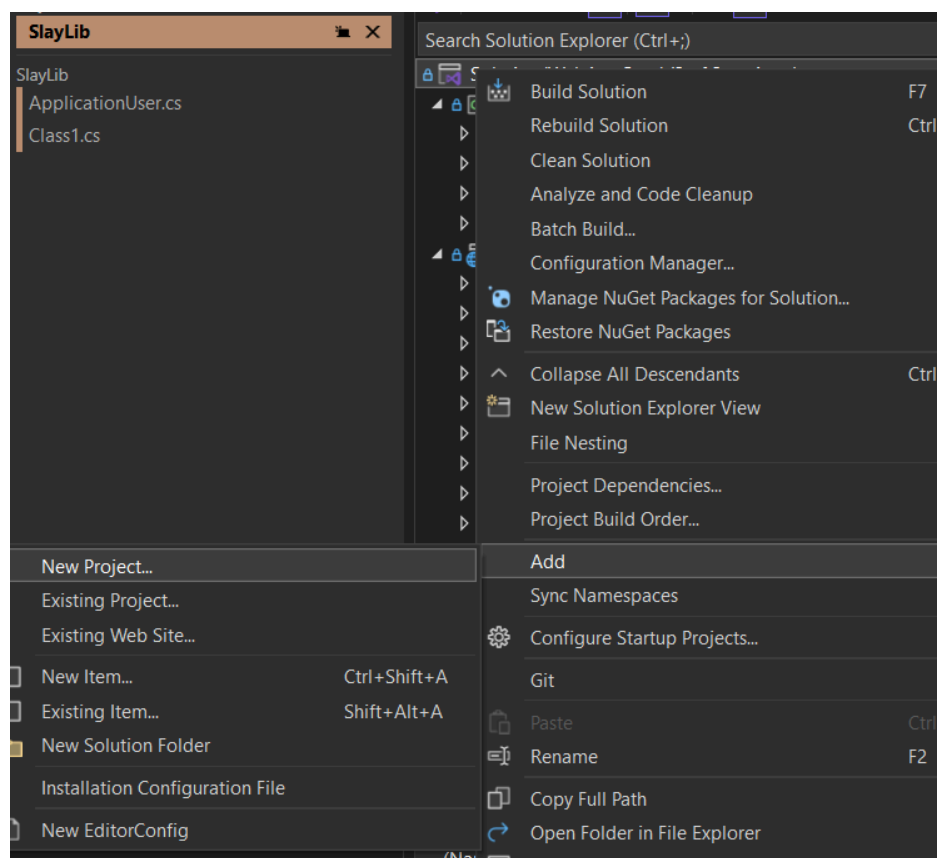


Рис 1.2.1 – Створення нового проєкту

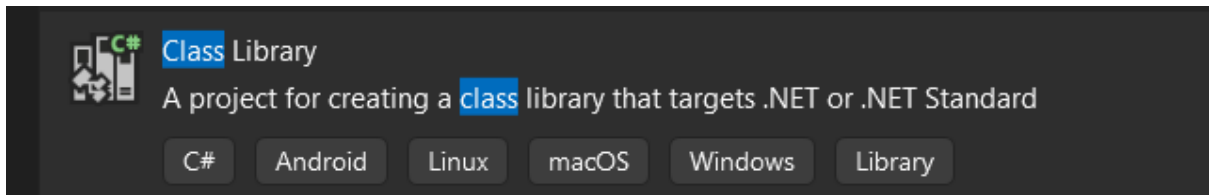


Рис 1.2.2 – Додавання нового модуля у вигляді проєкту бібліотеки класів

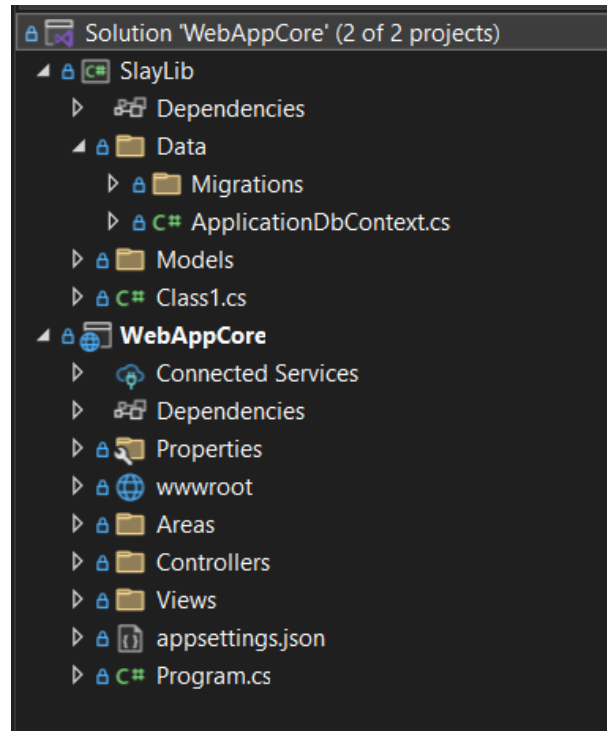


Рис 1.2.3 – Розділений ASP.NET Core застосунок на модулі

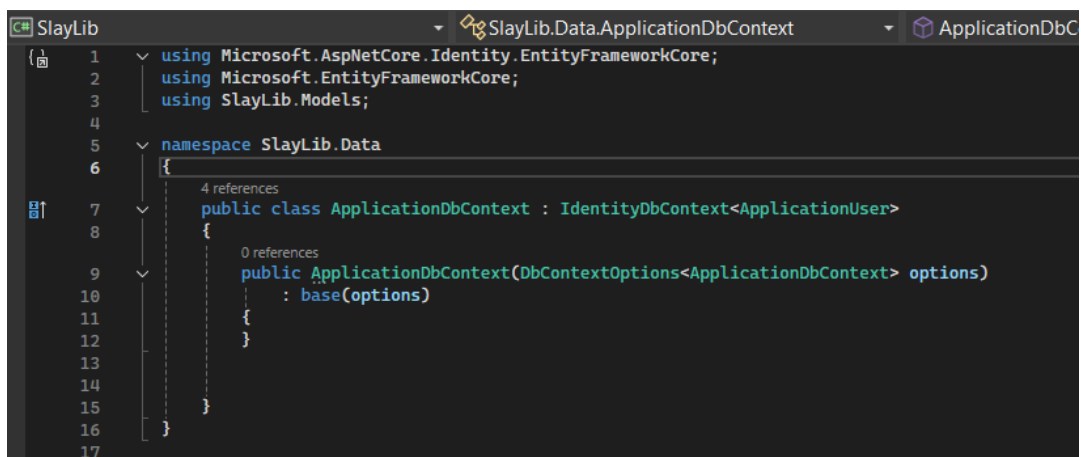


Рис 1.2.4 – Скорегований ідентифікатор просторів імен переміщених класів

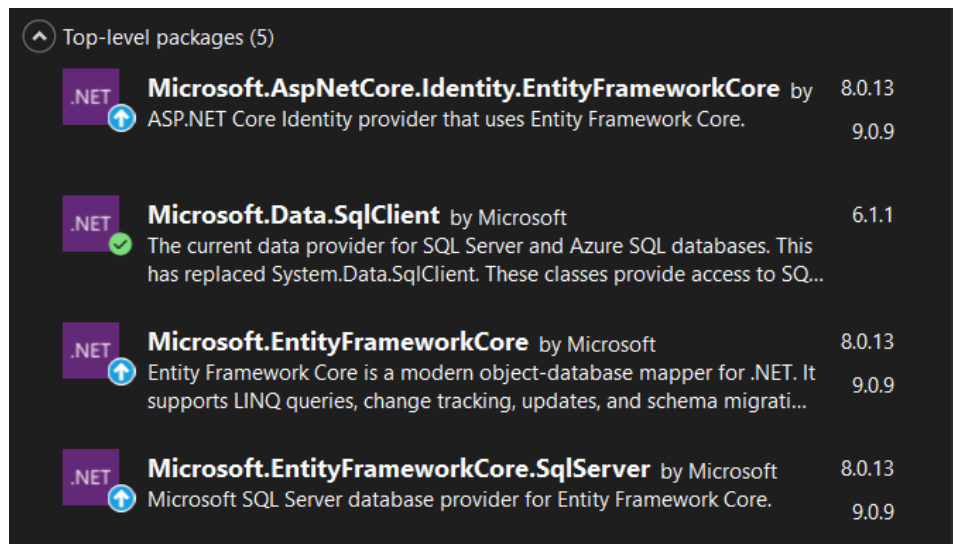


Рис 1.2.5 – Встановлення необхідних бібліотек у менеджері NuGet

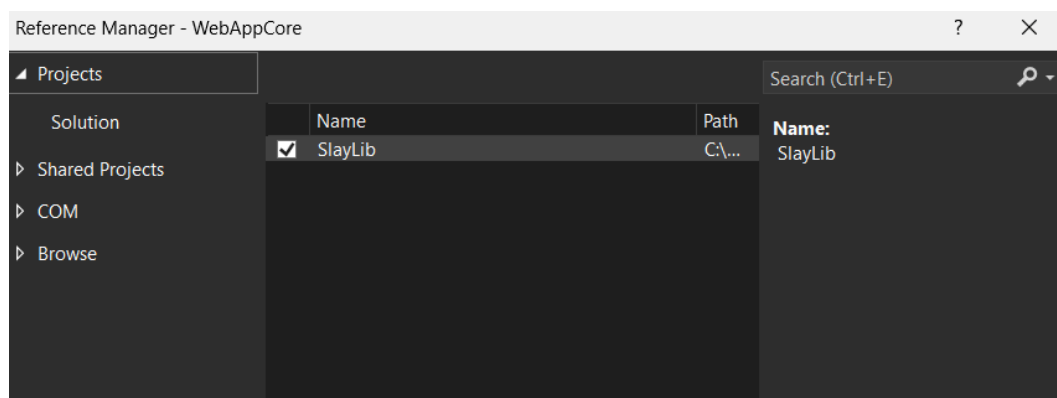


Рис 1.2.6 – Забезпечення доступності контексту та моделей у веб-застосунку

### 1.3. Розширення класу користувача IdentityUser

У папці Models створюється власний клас користувача, який наслідує IdentityUser.

До нього додаються нові властивості — FirstName та LastName, що дозволяють зберігати ім'я та прізвище користувача.

```

1  using Microsoft.AspNetCore.Identity;
2
3  namespace SlayLib.Models
4  {
5      1 reference
6      public class ApplicationUser : IdentityUser
7      {
8          0 references
9          public string FirstName { get; set; }
10         0 references
11         public string LastName { get; set; }
12     }
13 }

```

Рис. 1.3.1 – Розширений клас користувача

#### 1.4. Виконання міграцій та оновлення структури бази даних

Для внесення змін у модель користувача створюється нова міграція за допомогою Entity Framework Core.

Після виконання міграцій у базі даних з'являються нові поля `FirstName` та `LastName` у відповідній таблиці.

Під час переходу на власний тип користувача `ApplicationUser` можуть виникнути помилки в реєстрації, логіні чи логауті.

Основна причина — залишки старого типу `IdentityUser` у класах `Register`, `Login`, `Logout`.

Проблему вирішено шляхом заміни всіх згадок `IdentityUser` на `ApplicationUser`, перевірки коректності `Dependency Injection` у `Program.cs` і повторного виконання міграцій.

Після цього робота автентифікації повністю відновлюється: користувачі можуть реєструватися та входити в систему.

```

Package Manager Console
Package source: All Default project: SlayLib

PM> Add-Migration InitIdentity
Build started...
Build succeeded.
To undo this action, use Remove-Migration.
PM> Update-Database
Build started...
Build succeeded.
Microsoft.EntityFrameworkCore.Database.Command[20101]
  Executed DbCommand (17ms) [Parameters=[], CommandType='Text', CommandTimeout='30']
  SELECT 1
Microsoft.EntityFrameworkCore.Database.Command[20101]
  Executed DbCommand (11ms) [Parameters=[], CommandType='Text', CommandTimeout='30']
  SELECT OBJECT_ID(N'[_EFMigrationsHistory]');
Microsoft.EntityFrameworkCore.Database.Command[20101]
  Executed DbCommand (1ms) [Parameters=[], CommandType='Text', CommandTimeout='30']
  SELECT 1
Microsoft.EntityFrameworkCore.Database.Command[20101]
  Executed DbCommand (0ms) [Parameters=[], CommandType='Text', CommandTimeout='30']
  SELECT OBJECT_ID(N'[_EFMigrationsHistory]');
Microsoft.EntityFrameworkCore.Database.Command[20101]
  Executed DbCommand (2ms) [Parameters=[], CommandType='Text', CommandTimeout='30']
  SELECT [MigrationId], [ProductVersion]
  FROM [_EFMigrationsHistory]
  ORDER BY [MigrationId];
Microsoft.EntityFrameworkCore.Migrations[20402]
  Applying migration '20251007171151_InitIdentity'.
Applying migration '20251007171151_InitIdentity'.
Microsoft.EntityFrameworkCore.Database.Command[20101]
  Executed DbCommand (53ms) [Parameters=[], CommandType='Text', CommandTimeout='30']
  ALTER TABLE [AspNetUsers] ADD [FirstName] nvarchar(max) NOT NULL DEFAULT N'';
Microsoft.EntityFrameworkCore.Database.Command[20101]
  Executed DbCommand (3ms) [Parameters=[], CommandType='Text', CommandTimeout='30']
  ALTER TABLE [AspNetUsers] ADD [LastName] nvarchar(max) NOT NULL DEFAULT N'';
Microsoft.EntityFrameworkCore.Database.Command[20101]
  Executed DbCommand (7ms) [Parameters=[], CommandType='Text', CommandTimeout='30']
  INSERT INTO [_EFMigrationsHistory] ([MigrationId], [ProductVersion])
  VALUES (N'20251007171151_InitIdentity', N'8.0.13');

Done.
PM>

```

Рис 1.4.1 – Успішне виконання міграцій

Update Script File: dbo.AspNetUsers.sql*					
	Name	Data Type	Allow Nulls	Default	
	NormalizedEmail	nvarchar(256)	<input checked="" type="checkbox"/>		
	EmailConfirmed	bit	<input type="checkbox"/>		
	PasswordHash	nvarchar(MAX)	<input checked="" type="checkbox"/>		
	SecurityStamp	nvarchar(MAX)	<input checked="" type="checkbox"/>		
	ConcurrencyStamp	nvarchar(MAX)	<input checked="" type="checkbox"/>		
	PhoneNumber	nvarchar(MAX)	<input checked="" type="checkbox"/>		
	PhoneNumberConfirmed	bit	<input type="checkbox"/>		
	TwoFactorEnabled	bit	<input type="checkbox"/>		
	LockoutEnd	datetimeoffset(7)	<input checked="" type="checkbox"/>		
	LockoutEnabled	bit	<input type="checkbox"/>		
	AccessFailedCount	int	<input type="checkbox"/>		
	FirstName	nvarchar(MAX)	<input checked="" type="checkbox"/>	(N'')	
	LastName	nvarchar(MAX)	<input checked="" type="checkbox"/>	(N'')	
			<input type="checkbox"/>		

Рис 1.4.2 - таблиця відповідної бази даних

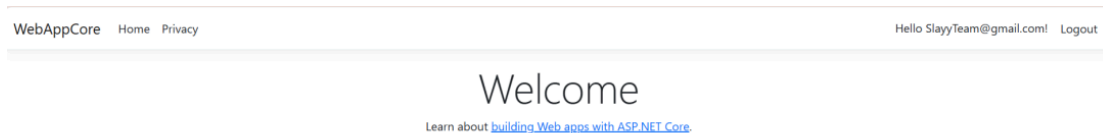


Рис 1.4.3 – Головна сторінка веб-застосунку після входу користувача

## 1.5 Створення репозиторію на GitHub проєкту

Завершальний етап - створення репозиторію проєкту на GitHub для зберігання коду, контролю версій і спільної роботи над проєктом.

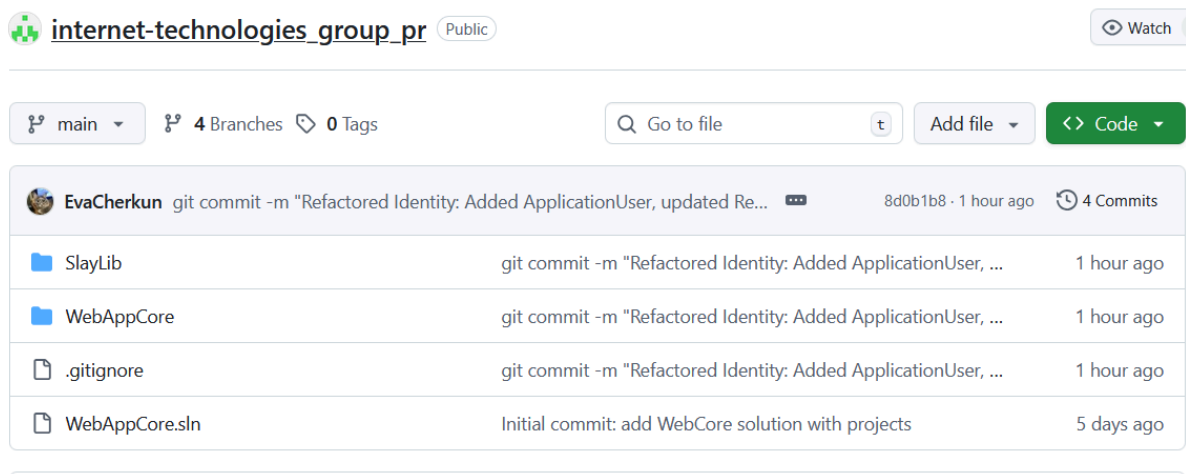


Рис. 1.5.1 – Репозиторій проєкту на GitHub

## **Висновок**

У ході роботи створено рішення Visual Studio з проєктом ASP.NET Core із використанням шаблону MVC та автентифікації Individual Accounts. Клас IdentityUser було розширено властивостями FirstName та LastName. Застосунок структуровано на модулі шляхом винесення моделей і контексту даних у окремий проєкт SlayLib, де було налаштовано правильні простори імен і підключення через Project Reference.

Проведено міграцію бази даних із оновленою структурою таблиць, протестовано автентифікацію користувачів і створено репозиторій на

GitHub.

Таким чином, у лабораторній роботі продемонстровано навички роботи з ASP.NET Core Identity, Entity Framework Core, модульною архітектурою та системою контролю версій Git.