Київський національний університет ім. Тараса Шевченка

Кафедра мережевих та інтернет технологій

**Лабораторна робота № 1**

**Дисципліна: Сучасні інтернет технології**

**Тема:** Структура проєкту ASP.NET CORE веб-застосунку. Розділення моноліту на модулі (шари)

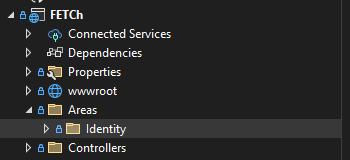
Виконав: Студент групи МІТ-41

Пугач Назар

**Хід роботи**

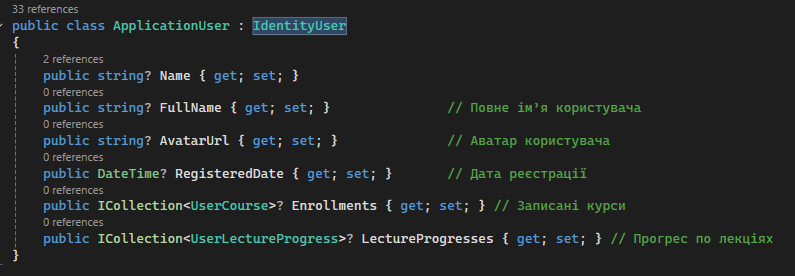
**Завдання 1.1:** Створити рішення (Visual Studio solution), у якому містяться проєкт ASP.NET Core, забезпечивши наявність автентифікації Individual Accounts.

Було створено **ASP.NET Core MVC застосунок** у Visual Studio. При створенні обрано шаблон з **Authentication – Individual Accounts**. Це забезпечило інтеграцію **ASP.NET Core Identity** для реєстрації та входу користувачів.

  
Рисунок 1.1 – Перевірка наявності папки Identity.

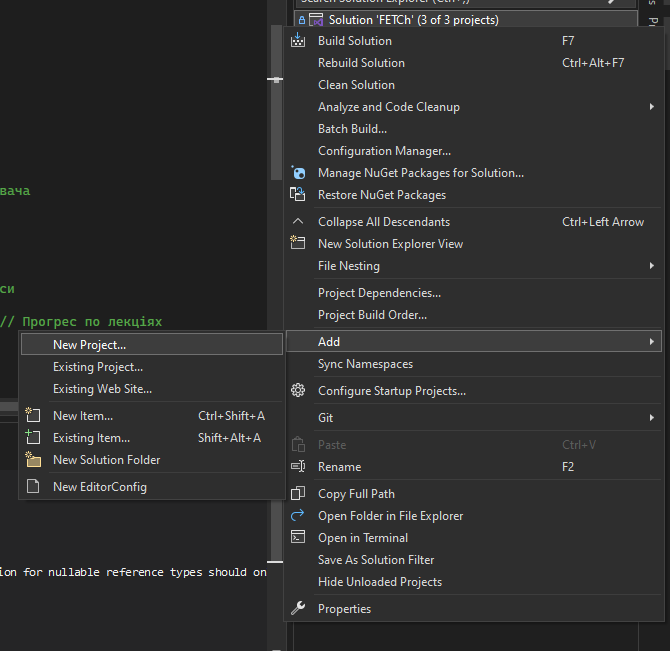
**Завдання 1.2:** Розширити клас автентифікованого користувача IdentityUser принаймні двома властивостями FirstName та LastName.

Створимо клас що наслідує клас IdentityUser.

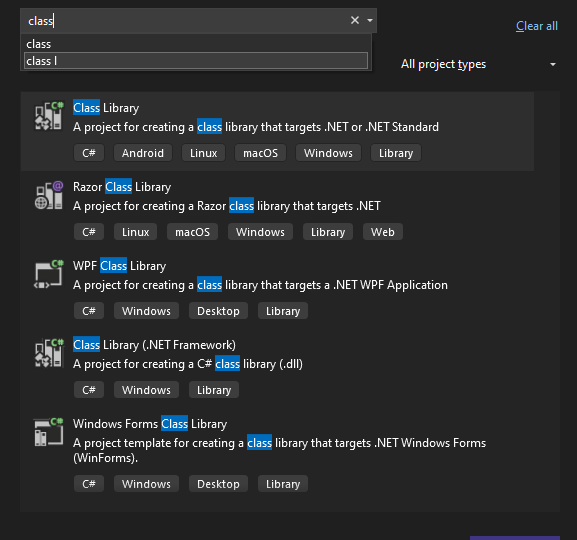
  
Рисунок 1.2 – Розширити клас автентифікованого користувача IdentityUser.

**Завдання 1.3:** Розділити ASP.NET Core застосунок на модулі, винісши контекст даних та моделі даних в окремий проєкт в складі одного й того ж рішення. Забезпечити належні ідентифікатори просторів імен класів нового модуля; забезпечити досяжність контексту даних та моделей даних для веб-застосунку.

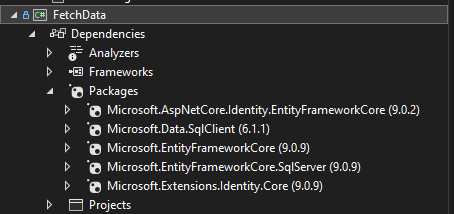
Спочатку створимо модулі в солюшині.

  
Рисунок 1.3 – Створення додатковиих модулів.

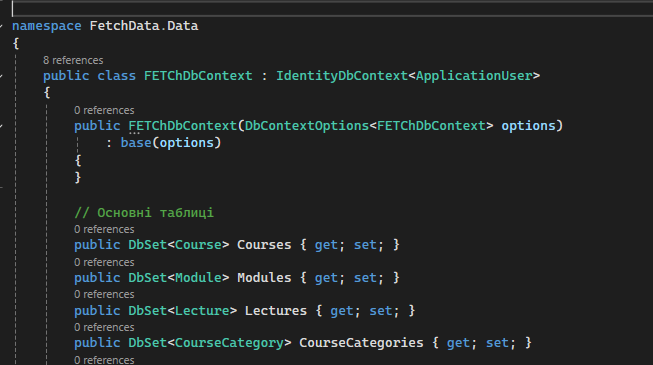
Створимо модулі типу Class Library для DbContext та Models.

  
Рисунок 1.4 – Створення додатковиих модулів.

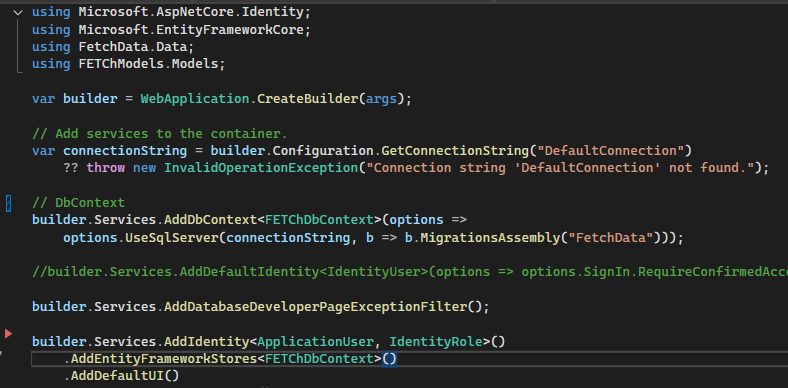
Завантажимо необхідні пакети в модулі. Для модуля з DbContext це пакети з EntityFramework та Identity.

  
Рисунок 1.5 – Додавання необхідних пакетів.

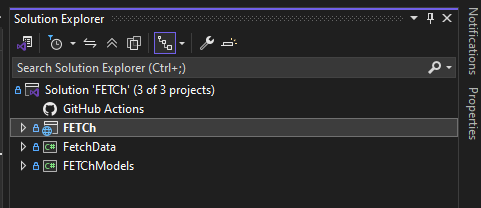
Створимо контекст даних.

  
Рисунок 1.6 – Контекст даних.

Додамо контекст даних в проєкт. Для цього требу додати модуль з контекстом даних в основний проекст, та змінити контекст даних в Program.cs.

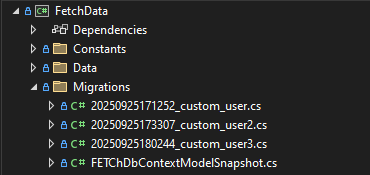
  
Рисунок 1.7 – Підключення контексту даних.

Виконаємо все те саме для Models.

  
Рисунок 1.8 – Результат поділу застосунка на шари.

**Завдання 1.4:** Виконати міграцію, перевірити оновлення структури таблиць відповідної бази даних.

Щоб виконати міграцію необхідно відкрити Package Manager Console, вибрати Default project в якому знаходиться наш контекст та виконати команди add-migration [назва міграцію] та update-database. В папці Migrations мають з’явитись відповідні класи.

  
Рисунок 1.9 – Виконання міграцій.

**Завдання 1.5:** Створити репозиторій на GitHub, зафіксувати зміни у проєкті та надіслати посилання на репозиторій викладачу для перевірки.

[Посилання на репозиторій](https://github.com/mulykoV/FETCh)

**Висновок:** Під час виконання лабораторно роботи було створено рішення (Visual Studio solution), у якому містяться проєкт ASP.NET Core; забезпечена наявність автентифікації Individual Accounts; розширено клас автентифікованого користувача IdentityUser властивостями; розділено ASP.NET Core застосунок на модулі; внесен контекст даних та моделі даних в окремий проєкт в складі одного й того ж рішення; забезпечено належні ідентифікатори просторів імен класів нового модуля; забезпечено досяжність контексту даних та моделей даних для веб-застосунку; виконано міграцію, перевірено оновлення структури таблиць відповідної бази даних; створено репозиторій на GitHub, зафіксовано зміни у проєкті.