Київський національний університет ім. Тараса Шевченка

Кафедра мережевих та інтернет технологій

**Практична робота № 2**

**Дисципліна:** Сучасні Інтернет Технології

**Тема:** Робота з даними в ASP.NET Core. Реалізація шаблону Repository

Виконав: Студент групи МІТ-41

Полюхович Андрій

**Хід виконання**

1. Створити базовий та конкретний інтерфейси репозиторію, що визначає базові методи для роботи з даними (отримання всіх записів, пошук за умовою, додавання, оновлення, видалення).

Створюємо два інтерфейси: базовий (IRepository) з базовими методами та конкретний (IUserRepository) з конкретними методами.

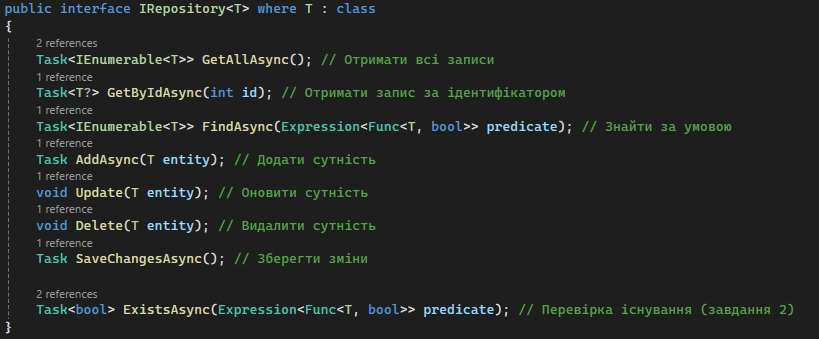


рис. 2.1 – базовий інтерфейс (IRepository)

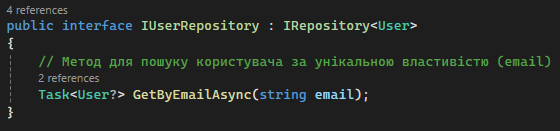


рис.2.2 – конкретний інтерфейс

1. Додати базовий метод для перевірки існування сутності за умовою (наприклад, ExistsAsync()). Пояснити, у яких випадках він може бути корисним.

Метод ExistsAsync() є корисним для перевірки наявності запису в базі даних за заданою умовою перед виконанням операцій вставки, оновлення чи видалення. Це допомагає уникнути дублювання даних або можливих помилок.

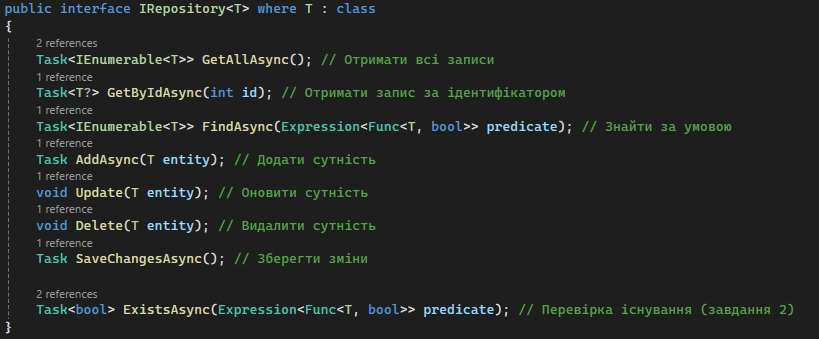


рис. 2.3 – ExistsAsync()

1. Реалізувати базовий клас репозиторію, який інкапсулює роботу з контекстом бази даних та забезпечує виконання CRUD-операцій.

Для створення цього класу в папці Repositories ми створюємо клас Repository, який успадковує наш базовий інтерфейс та реалізує його методи

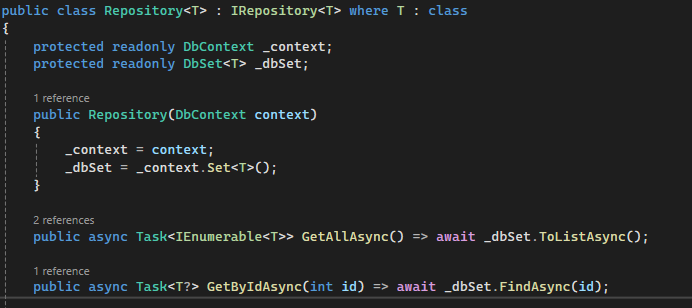


рис. 2.4 – базовий клас репозиторію

1. Створити конкретний клас репозиторію для веб-застосунку та розширити його функціональність методом, що виконує пошук користувача за унікальною властивістю (наприклад, email).

Створюємо клас UserRepository, у якому успадковуємо наш конкретний репозиторій та розширюємо функціональність методом пошуку користувача за допомогою email.

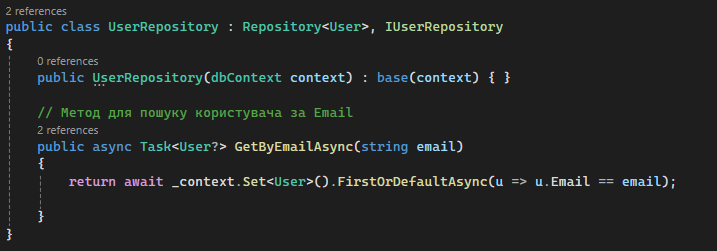


рис. 2.5 – конкретний клас репозиторію

1. Зареєструвати залежності (інтерфейсу репозиторію та його реалізації) у контейнері впровадження залежностей.

У файлі program.cs ми реєструємо наш сервіс репозиторію.



рис. 2.6 – реєстрація залежностей

1. Інтегрувати репозиторій у контролер: реалізувати метод, який отримує дані, використовуючи репозиторій.

Створюємо файл UsersController в папці Controllers та Інтегруємо репозиторій додавши туди залежність.

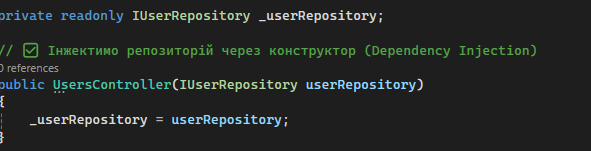


рис. 2.7 – реєстрація залежностей

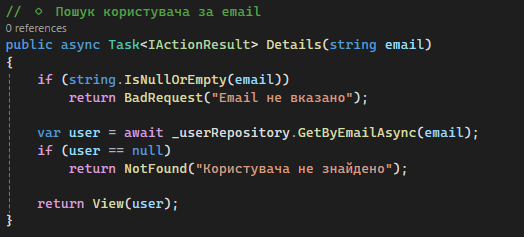


рис. 2.8 – метод репозиторію

**Висновок:** У ході виконання лабораторної роботи було створено базовий та конкретний інтерфейс, базовий та конкретний клас репозиторію. Зареєстровано залежності у контейнері впровадження залежностей. Інтегровано репозиторій у контролер.