Київський національний університет ім. Тараса Шевченка

Кафедра мережевих та інтернет технологій

**Лабораторна робота № 2**

**Дисципліна: Сучасні інтернет технології**

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

Виконав: Студент групи МІТ-41

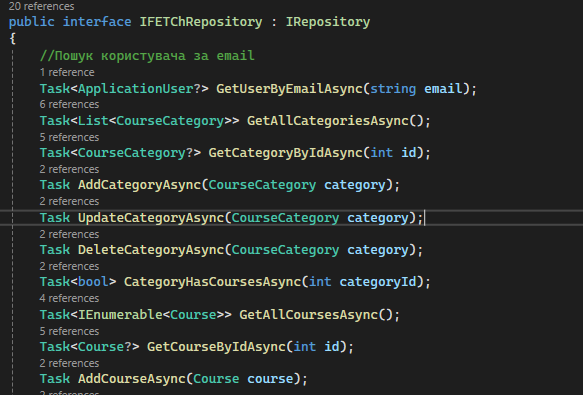
Пугач Назар

**Хід роботи**

**Завдання 2.1:** Створити базовий та конкретний інтерфейси репозиторію, що визначає базові методи для роботи з даними: отримання всіх записів, пошук за умовою, додавання, оновлення, видалення.

В модулі що відповідає за дані свторюємо два інтерфейси IRepository та IFETChRepository. В IRepository будуть базові методи а в IFETChRepository будуть конкретні методи для нашого застосунку.

  
Рисунок 2.1 – Інтерфейс IRepository

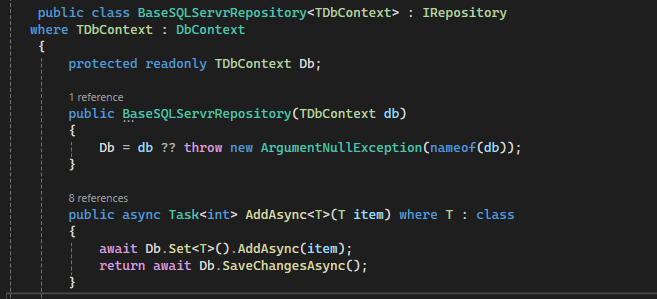
  
Рисунок 2.2 – Інтерфейс IFETChRepository

**Завдання 2.2:** Додати базовий метод для перевірки існування сутності за умовою ExistsAsync(). Пояснити, у яких випадках він може бути корисним.

Метод ExistsAsync() використовується для перевірки, чи існує в базі запис, що відповідає певній умові, наприклад перед додаванням нового запису, щоб уникнути дублювання.

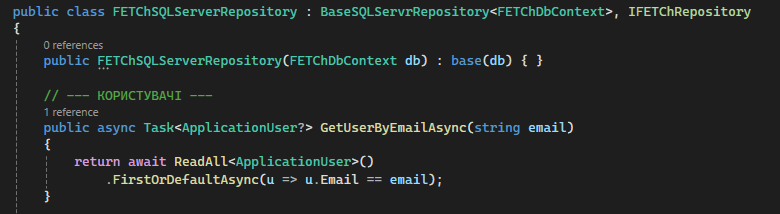
**Завдання 2.3:** Реалізувати базовий клас репозиторію, який інкапсулює роботу з контекстом бази даних та забезпечує виконання CRUD-операцій.

Базовий клас має наслідувати IRepository та імплементувати всі його методи.

  
Рисунок 2.3 – Базовий клас репозиторію

**Завдання 2.4:** Створити конкретний клас репозиторію для веб-застосунку та розширити його функціональність методом, що виконує пошук користувача за унікальною властивістю.

Конкретний клас репозиторію має наслідувати IFETChRepository та імплементувати всі його методи.

  
Рисунок 2.4 – Конкретний клас репозиторію

Метод для розширення це метод пошуку користувача по його електронній пошті.

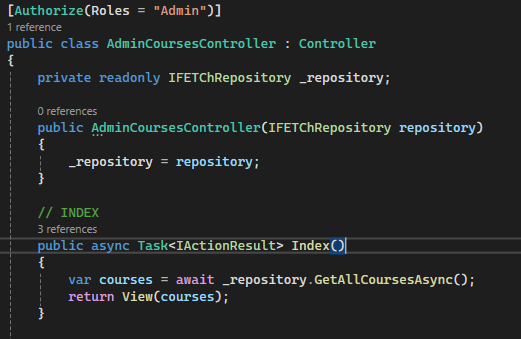
**Завдання 2.5:** Зареєструвати залежності інтерфейсу репозиторію та його реалізації у контейнері впровадження залежностей.

В Program.cs потрібно зареєструвати відповідний сервіс репозиторію.

  
Рисунок 2.5 – Впровадження залежностей інтерфейсу репозиторію та його реалізації у контейнері

**Завдання 2.6:** Інтегрувати репозиторій у контролер: реалізувати метод, який отримує дані, використовуючи репозиторій.

Щоб використати репозиторій в контролері необхідно додати залежність у контролер та використати метод репозиторію.

  
Рисунок 2.6 – Використання репозиторію в контролері для отримання даних

**Завдання 2.7:** Зафіксувати зміни у проєкті на GitHub.

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

**Висновок:** Під час виконання лабораторної роботи було реалізовано патерн Repository для ASP.NET CORE застосунку.