МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

БУДІВНИЦТВА І АРХІТЕКТУРИ

Кафедра інформаційних технологій

**Лабораторна робота № 5**

З дисципліни: Інструментальні засоби розробки програмного забезпечення

Виконав:

Студент групи ІСТм-24

Йовенко Дмитро

Київ-2025

**Мета:** розробка нової функції для мобільного застосунку "Трекер корисних звичок" у окремій гілці репозиторію з використанням моделі гілкування "Feature Branch Workflow".

**Хід роботи**

**1. Опис реалізованої функції**

Обрано функцію видалення звичок, яка дозволяє користувачу видаляти звички зі списку, відображеного в CollectionView. Функція включає:

* Додавання кнопки "Delete" до кожного елемента списку звичок.
* Видалення вибраної звички з бази даних SQLite.
* Оновлення інтерфейсу після видалення.

**1.1. Створення гілки для розробки**

У цій роботі будемо максимально користуватися інтеграцією git з Visual Studio, тому створюємо гілку за допомогою меню Git→New Branch.

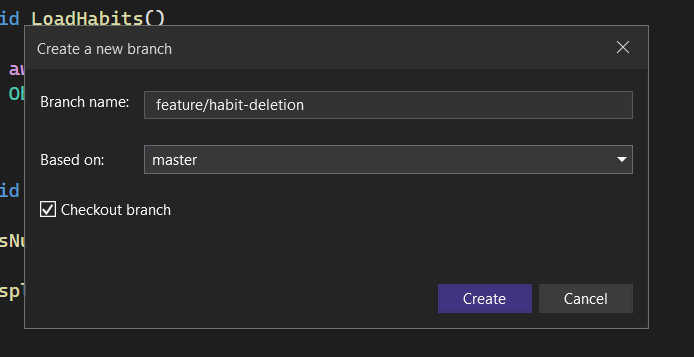


Рисунок 5.1 – Створення гілки feature/habit-deletion.

**1.2. Оновлення моделі даних і сервісу**

**Оновлення DatabaseService.cs**:

Додано метод DeleteHabitAsync для видалення звички за її ідентифікатором:

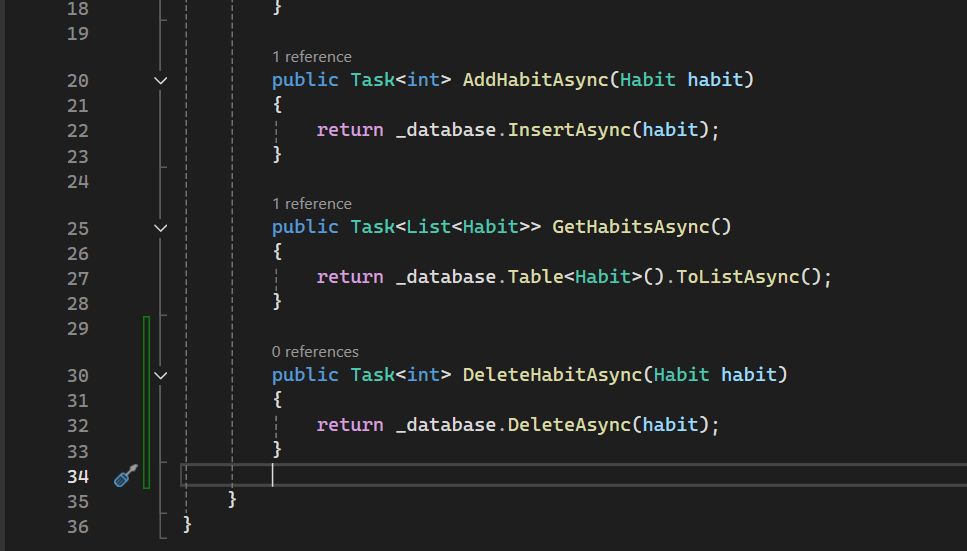


Рисунок 5.2 – Додані зміни до моделі бази даних

**Модель Habit.cs**:

Модель залишилася без змін, оскільки ідентифікатор (Id) уже підтримує операції видалення

**1.3. Оновлення інтерфейсу користувача**

Оновлено MainPage.xaml, додавши кнопку видалення до кожного елемента в CollectionView:

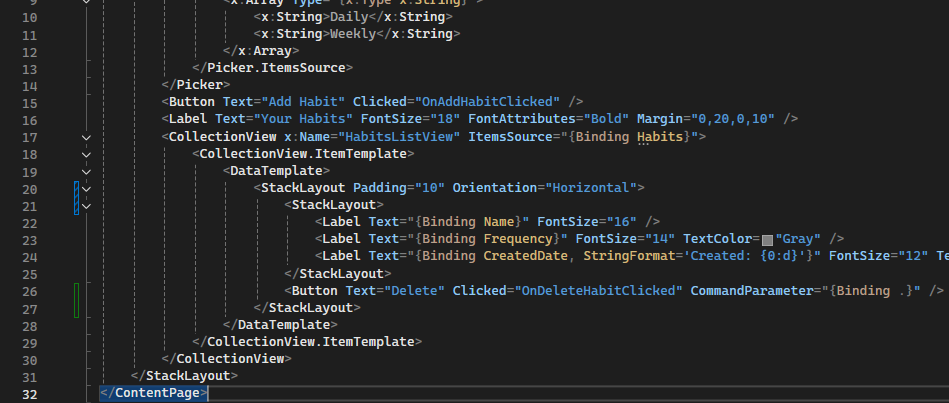


Рисунок 5.3 – Додані зміни у розмітку інтерфейсу

**1.4. Оновлення MainPage**

Оновлено MainPage.xaml.cs, додавши обробник для видалення звичок:

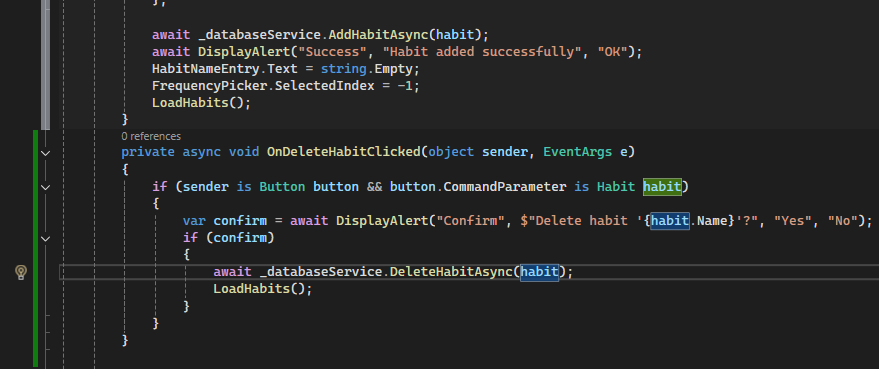


Рисунок 5.4 – Додана функіця у логіку інтерфейсу

**Технічні деталі**:

* Додано метод OnDeleteHabitClicked, який отримує об’єкт Habit через CommandParameter.
* Використано DisplayAlert для підтвердження видалення.
* Після видалення викликається LoadHabits для оновлення списку.

**1.5. Тестування та документація**

1. **Тестування**

Додані зміни було збережено та протестовано на реальному пристрої підключеному у режимі налагодження. У якості тестового пристрою Lenovo tb505xl на кастомній прошивці LinageOs (близька до AOSP) яка працює на Android 13.

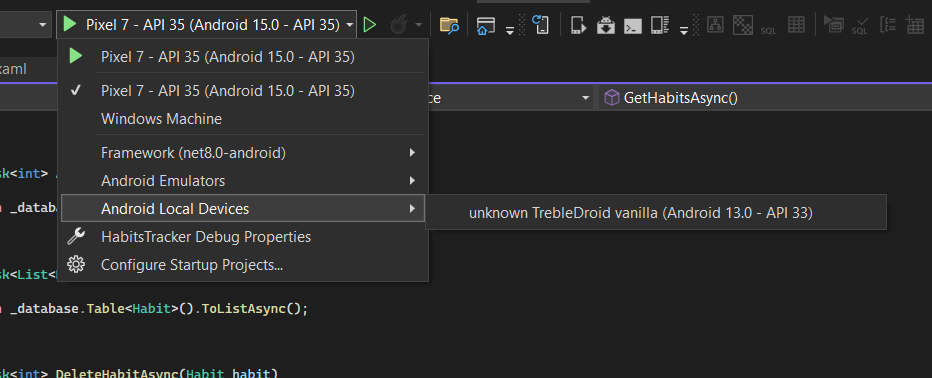


Рисунок 5.5 – Відображення локального пристрою у меню налагодження

Робота з застосунком проводиться прямо на пристрої, але переглядати інтерфейс можна у середовищі розробки у вікні XAML Live Preview.

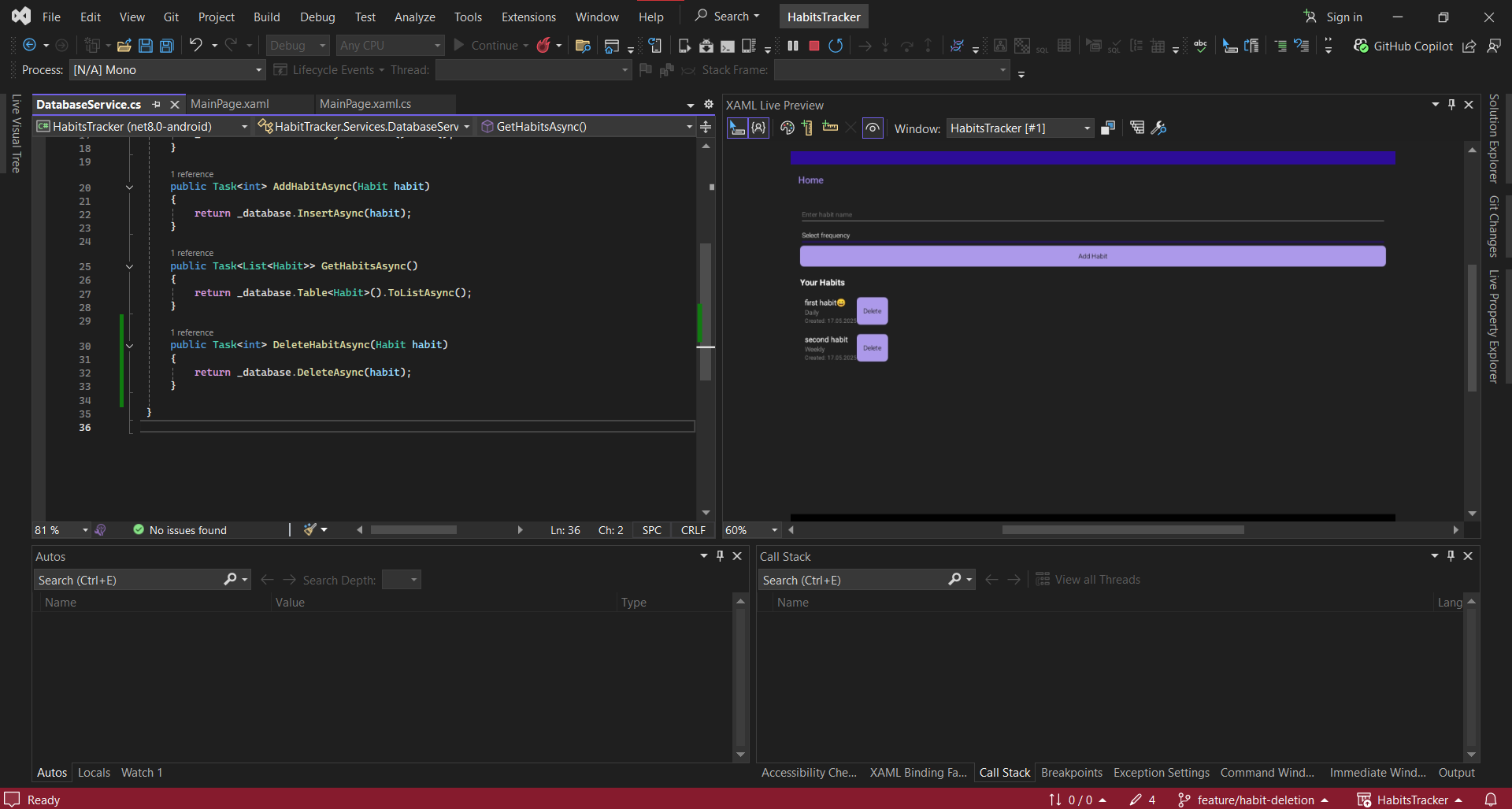


Рисунок 5.6 – Перегляд інтерфесу програми у XAML Live Preview

Для тестування спочатку було додано дві звички (рис.5.6), потім видалено «second habit»(рис.5.7), та перевірено оновлення списку звичок(рис.5.8)

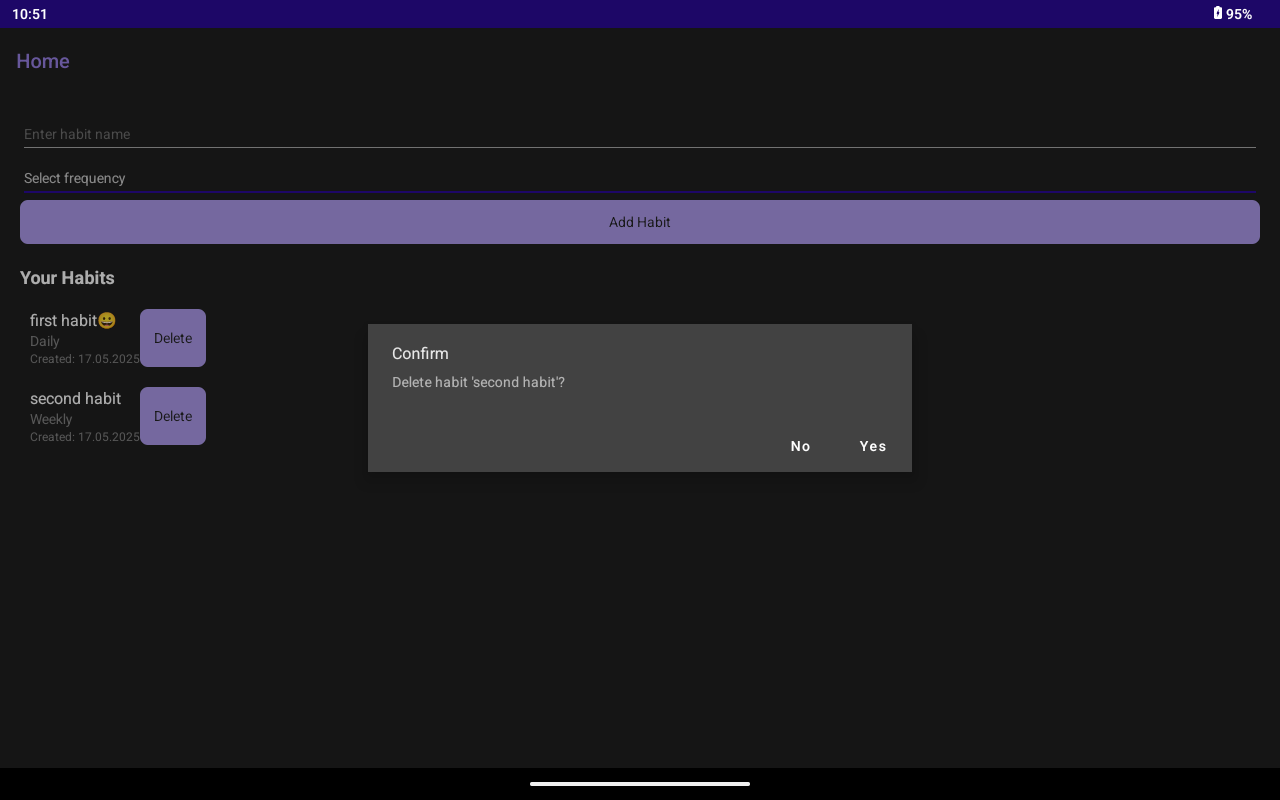


Рисунок 5.7 – Вікно підтвердження видалення

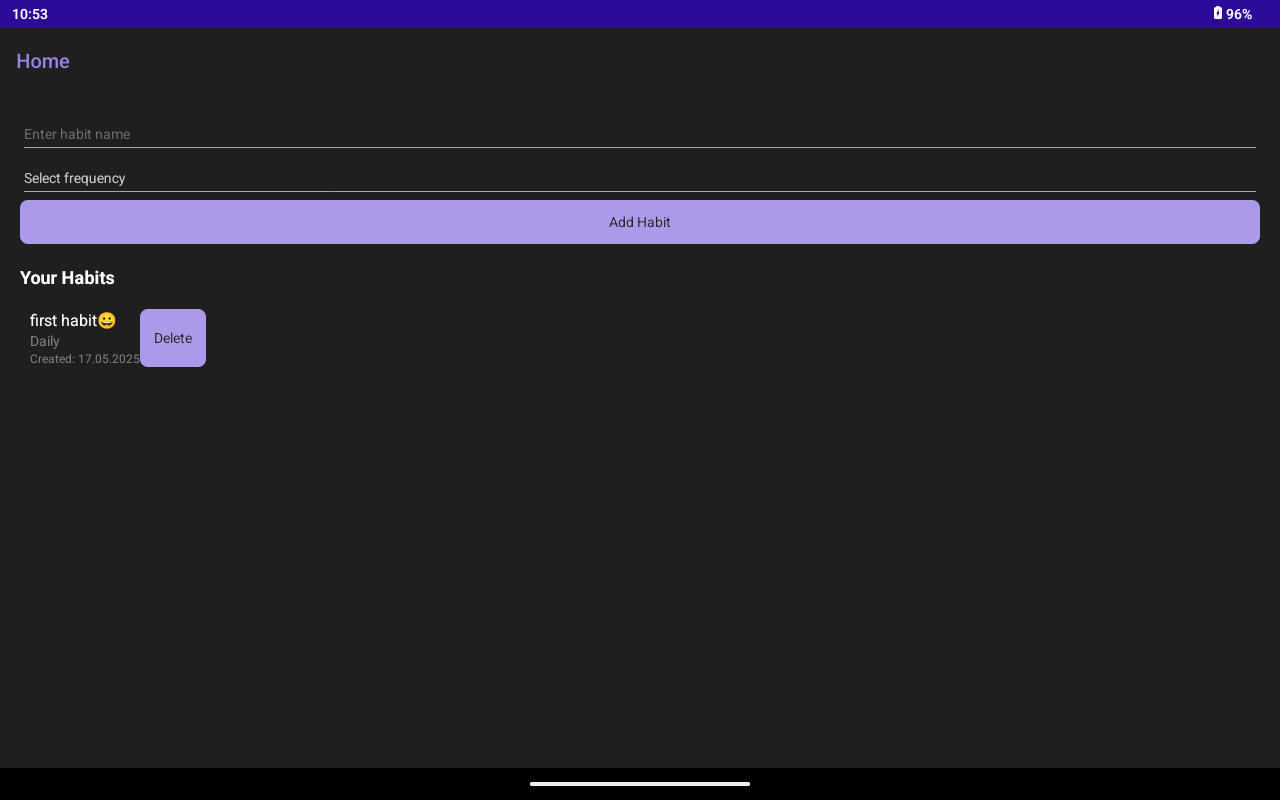


Рисунок 5.8 – Оновлений список звичок після видалення

1. **Документація:**

Оновлено файл README.md у гілці feature/habit-deletion, додавши опис функції:

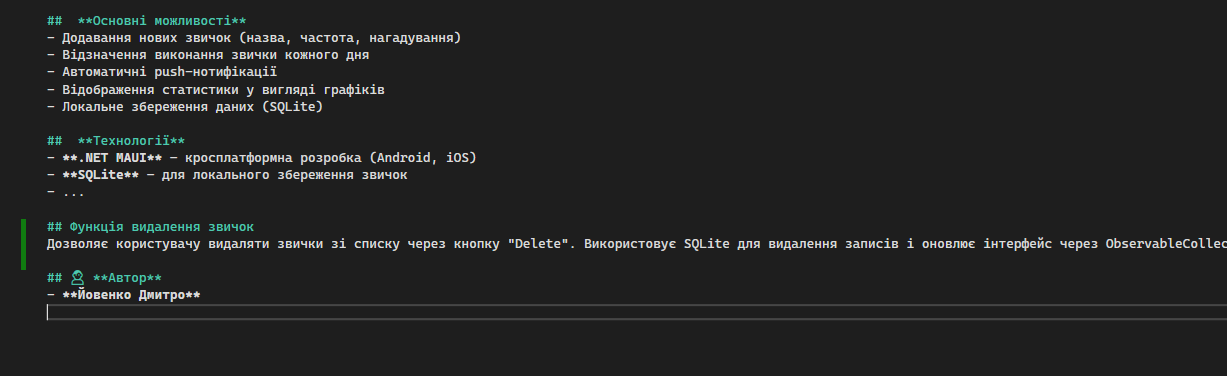


Рисунок 5.9 – Оновлення README.md

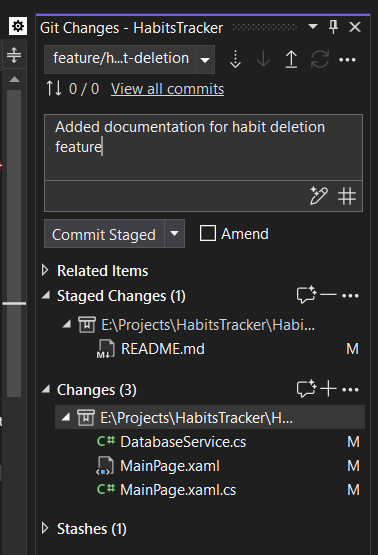


Рисунок 5.10 – Коміт змін у README.md

**2. Робота з системою контролю версій**

**2.1. Управління гілками**

1. **Створення та робота в гілці**:

Розробка велася в гілці feature/habit-deletion. Регулярно перевірявся статус, та в кінці всі зміни закомічено(рис 5.11)

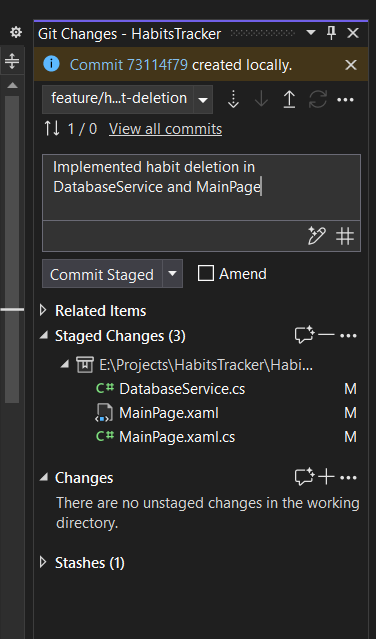


Рисунок 5.11 – Коміт змін у гілці feature/habit-deletion

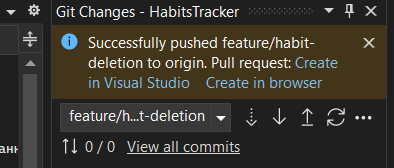


Рисунок 5.12 – Пуш змін на GitHub

**2.2. Процес злиття з основною гілкою**

**Створення Pull Request**:

На GitHub створено Pull Request для злиття feature/habit-deletion у master.

Перевірено, що немає конфліктів, і виконано злиття через інтерфейс GitHub.

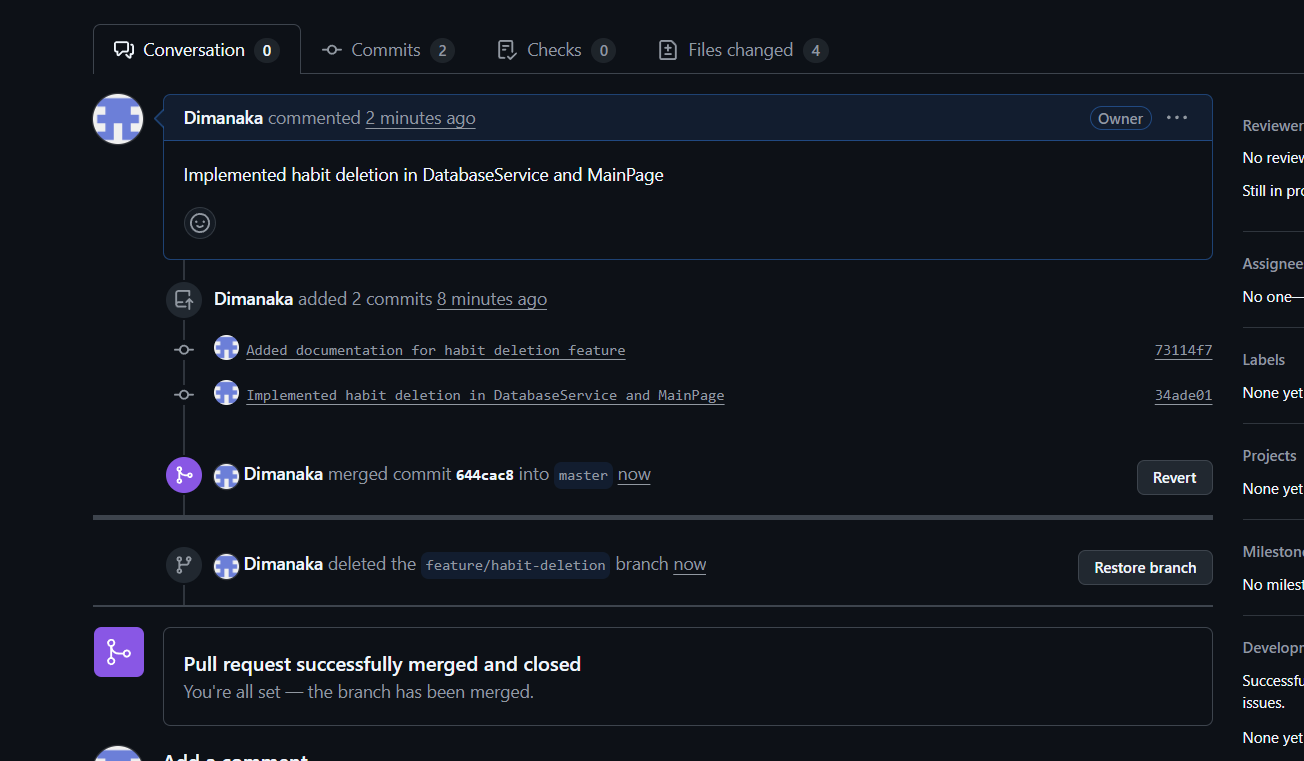


Рисунок 5.13 – Pull Request для злиття feature/habit-deletion.

**Посилання на репозиторій:** <https://github.com/Dimanaka/GoodHabitsTracker>

**Висновок**

У результаті лабораторної роботи успішно реалізовано функцію видалення звичок у мобільному застосунку "Трекер корисних звичок" із використанням .NET MAUI та SQLite. Функція дозволяє видаляти звички зі списку через кнопку "Delete" із підтвердженням, оновлюючи інтерфейс і базу даних. Розробка велася в ізольованій гілці feature/habit-deletion за моделлю "Feature Branch Workflow", що забезпечило безпечну інтеграцію змін.

Робота з Git поглибила навички управління гілками, синхронізації змін і злиття через Pull Request. Використання інтеграції Git у Visual Studio дозволило ефективно координувати розробку та підтримувати актуальність проєкту, взаємодіючи з системою контроля версій через зручний інтерфейс. Процес злиття пройшов без конфліктів.