МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

БУДІВНИЦТВА І АРХІТЕКТУРИ

Кафедра інформаційних технологій

**Лабораторна робота № 4**

З дисципліни: Інструментальні засоби розробки програмного забезпечення

Виконав:

Студент групи ІСТм-24

Йовенко Дмитро

Київ-2025

**Мета:** реалізація нового функціоналу в мобільному застосунку "Трекер корисних звичок" із використанням .NET MAUI та SQLite, а також поглиблення навичок роботи з системою контролю версій Git.

**Хід роботи**

У цій роботі реалізовано відображення списку звичок, збережених у базі даних, та виконано аналіз змін за допомогою інструментів git log, git diff, git blame і git status. Додатково створено та вирішено конфлікт злиття, щоб навчитися управляти спільною роботою над проєктом.

**1. Реалізація функціоналу відображення списку звичок**

Для реалізації відображення списку звичок виконано наступні кроки:

**1.1. Оновлення інтерфейсу користувача**

Модифіковано MainPage.xaml для додавання списку звичок за допомогою елемента CollectionView(рис.4.1).

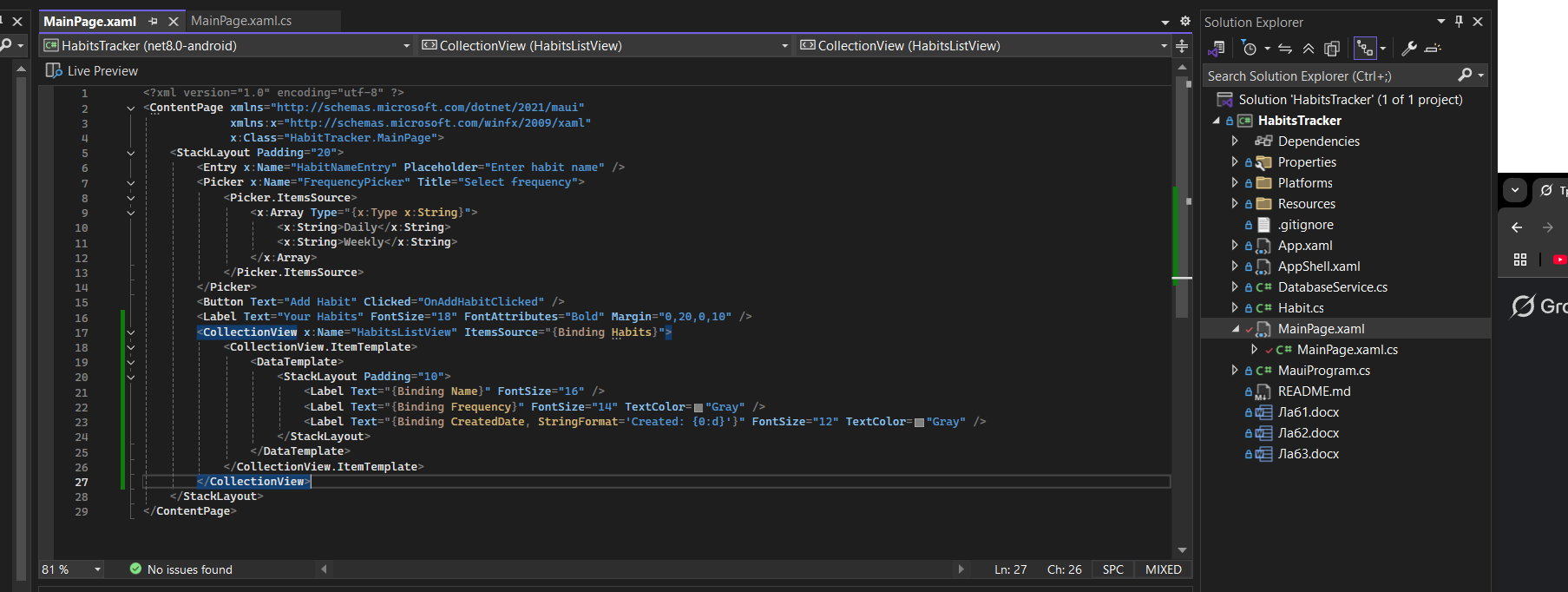


Рисунок 4.1 – Оновлена розмітка інтерфейсу

**1.2. Оновлення логіки роботи сторінки**

Оновлено MainPage.xaml.cs для завантаження списку звичок із бази даних та оновлення інтерфейсу після додавання нової звички(рис.4.2):

* + Додано ObservableCollection<Habit> для динамічного оновлення списку звичок.
  + Реалізовано INotifyPropertyChanged для підтримки прив’язки даних.
  + Метод LoadHabits завантажує звички з бази даних і оновлює Habits.
  + Виклик LoadHabits після додавання нової звички оновлює інтерфейс.

**Використані методи та технології:**

* **SQLite-net-pcl**: Для отримання списку звичок через GetHabitsAsync.
* **MVVM (частково)**: Використано прив’язку даних через BindingContext і ObservableCollection для підготовки до повноцінної MVVM-архітектури.
* **Асинхронне програмування**: Використано async/await для роботи з базою даних

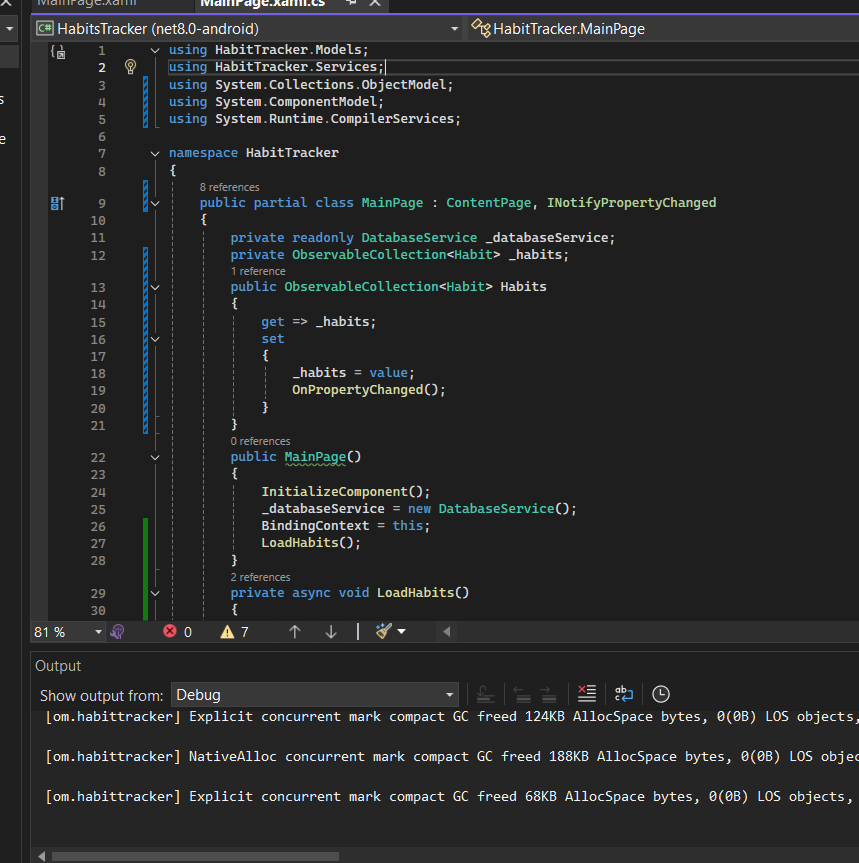


Рисунок 4.2 – Оновлена логіка роботи інтерфейсу

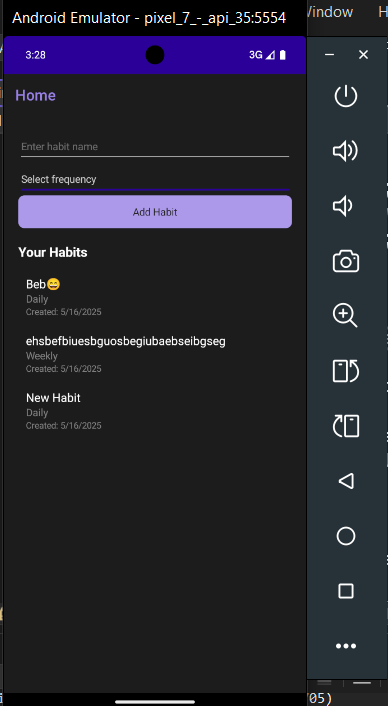


Рисунок 4.3 – Оновлений інтерфейс

**2. Робота з системою контролю версій**

**2.1. Використання інструментів Git**

Після реалізації функціоналу виконано аналіз змін за допомогою інструментів Git:

**Git status**:

Результат: Виявлено зміни в MainPage.xaml і MainPage.xaml.cs.

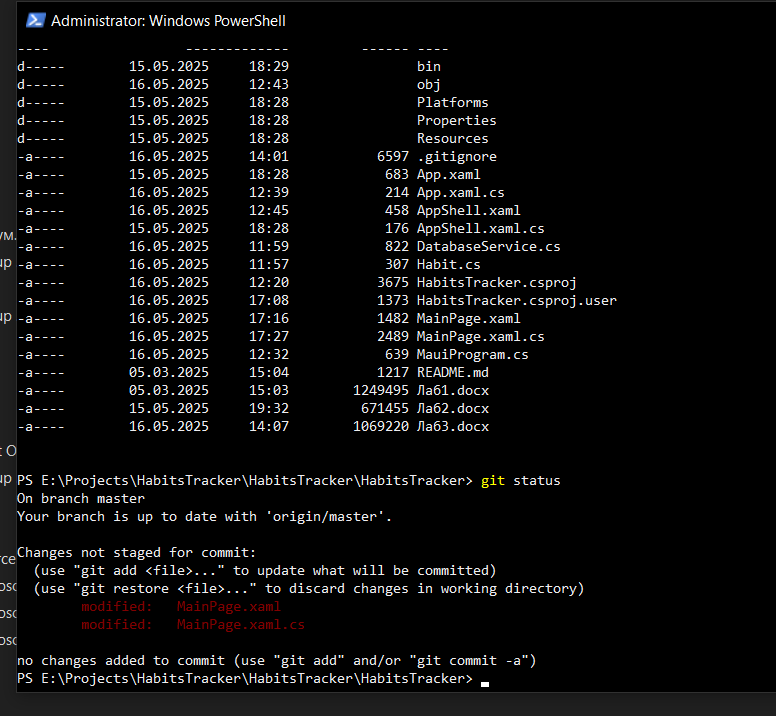


Рисунок 4.4 – Результат виконання git status

**Git log:**

Результат: Переглянуто історію, включаючи попередні коміти.

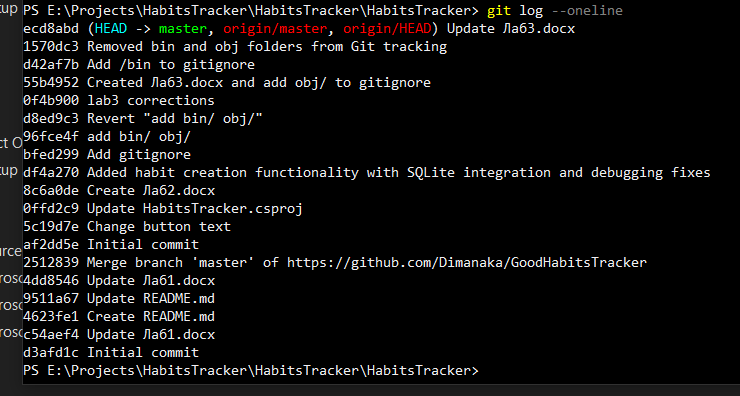


Рисунок 4.5 – Перегляд комітів за допомогою git log –oneline

**Git diff:**

Результат: Виявлено додавання ObservableCollection, INotifyPropertyChanged і методу LoadHabits.

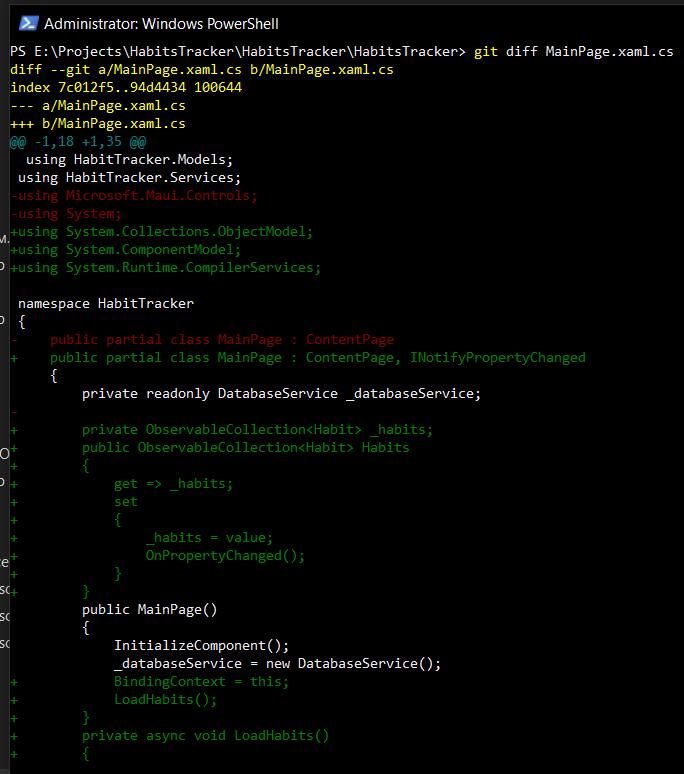


Рисунок 4.6 – Порівняння змін за допомогою git diff.

**Git blame**:

Результат: Підтверджено, що нові рядки коду для відображення списку звичок додані мною, інші як ще не закомічені.

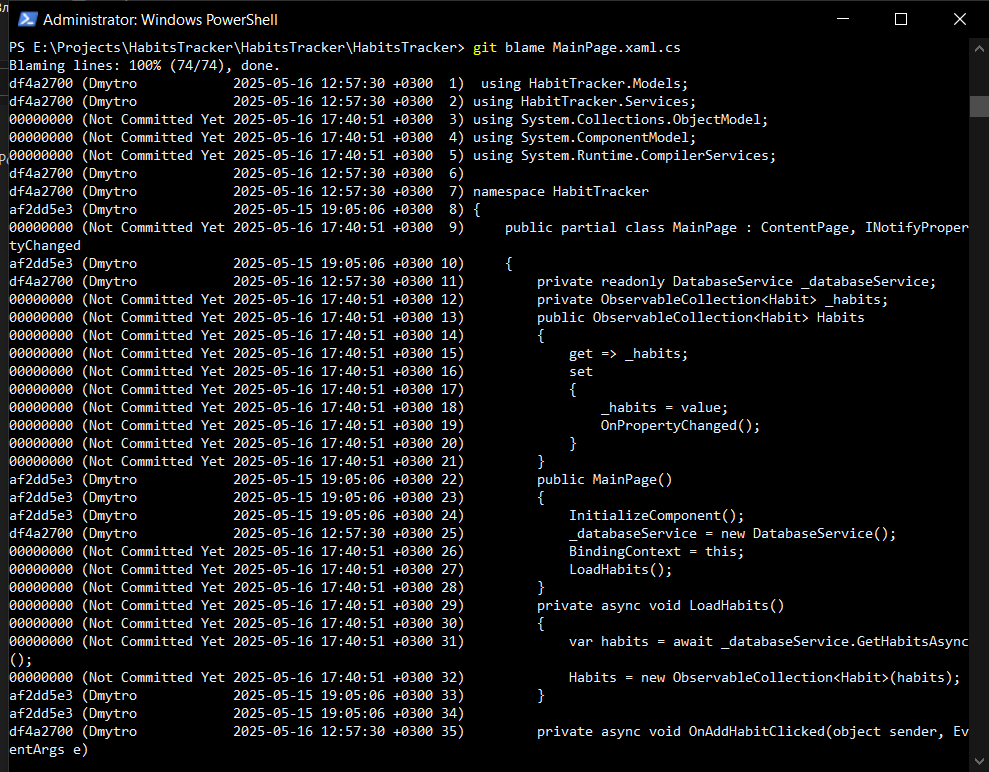


Рисунок 4.7 – Аналіз авторства за допомогою git blame

**Git commit, push:**

Результат: Після перевірок, проект закомічено та завантажено на GitHub

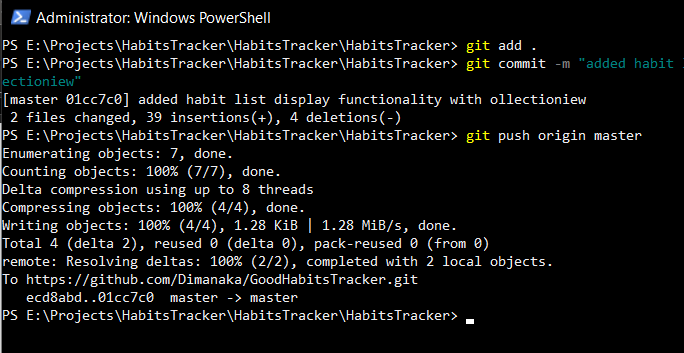


Рисунок 4.8 – Коміт змін на GitHub

**2.2. Симуляція та вирішення конфлікту злиття**

Для симуляції конфлікту злиття виконано наступні кроки:

1. **Створення нової гілки**:

Створено гілку feature/habit-list для внесення змін:

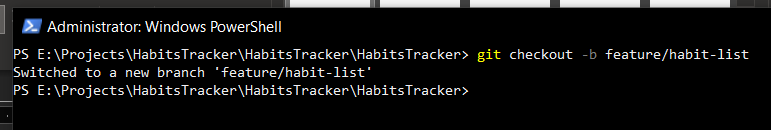


Рисунок 4.9 – Створення гілки feature/habit-list

У файлі MainPage.xaml.cs додано коментар до методу LoadHabits:

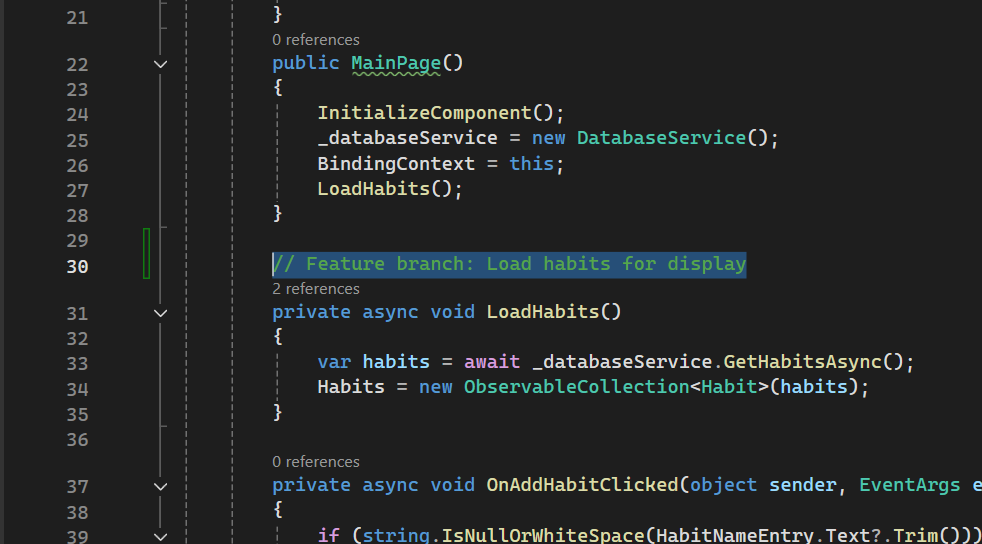


Рисунок 4.10 – Внесені зміни у файл MainPage.xaml.cs

Закомічено зміни:

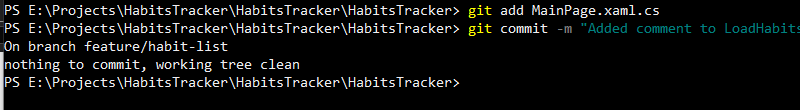


Рисунок 4.11 – Зроблено коміт гілки feature/habit-list

1. **Внесення змін у гілці main**:

Переключено на гілку main:

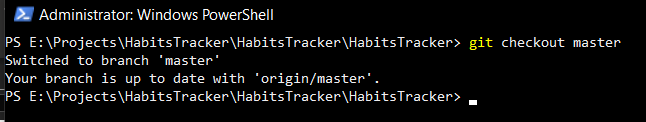


Рисунок 4.12 – Виконання git checkout для зміни гілки

Внесено зміни у код:

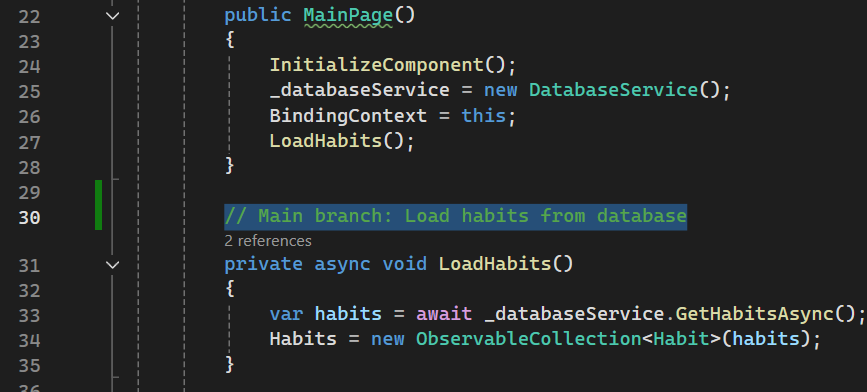


Рисунок 4.13 – Додано коментар відмінний від іншої гілки

Закомічено та завантажено на GitHub зміни:

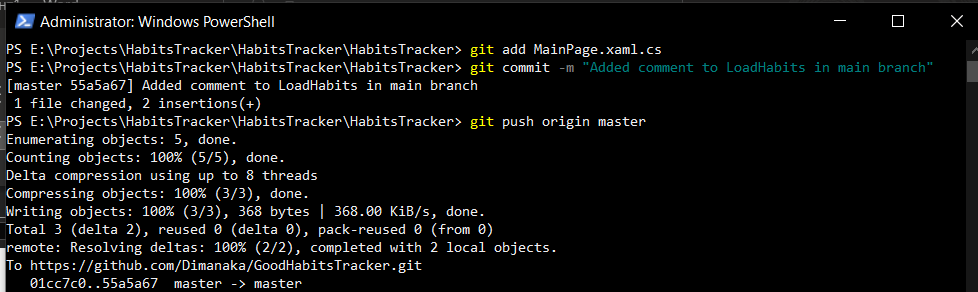


Рисунок 4.14 – Коміт оновлень на GitHub

1. Симуляція конфлікту:

Повертаємося до гілки feature/habit-list та спробуєм об’єднати гілки:

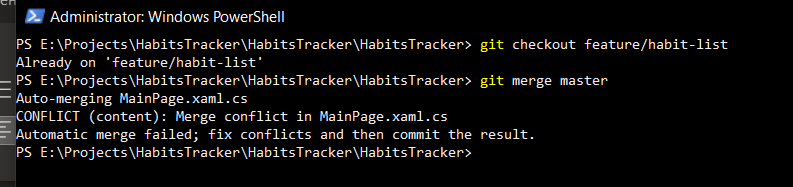


Рисунок 4.15 – Перемикаємо гілку та git merge для злиття гілок

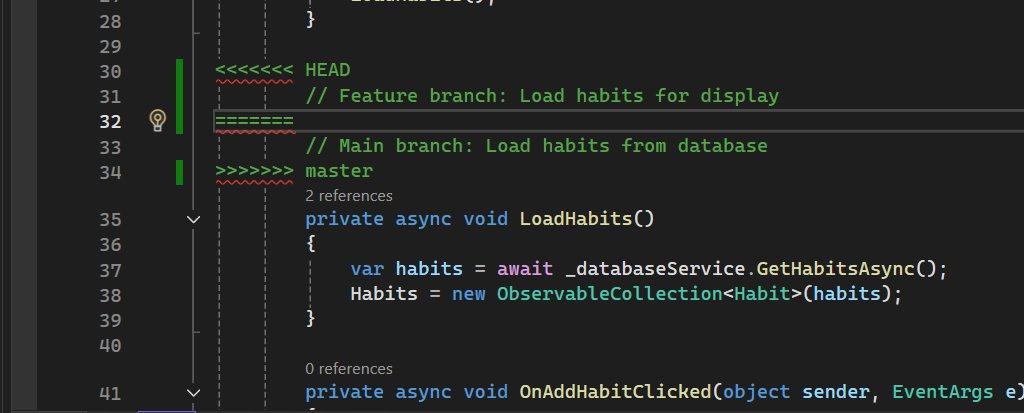


Рисунок 4.16 – Конфлікт помічений у файлі

Виправлення конфлікту вручну

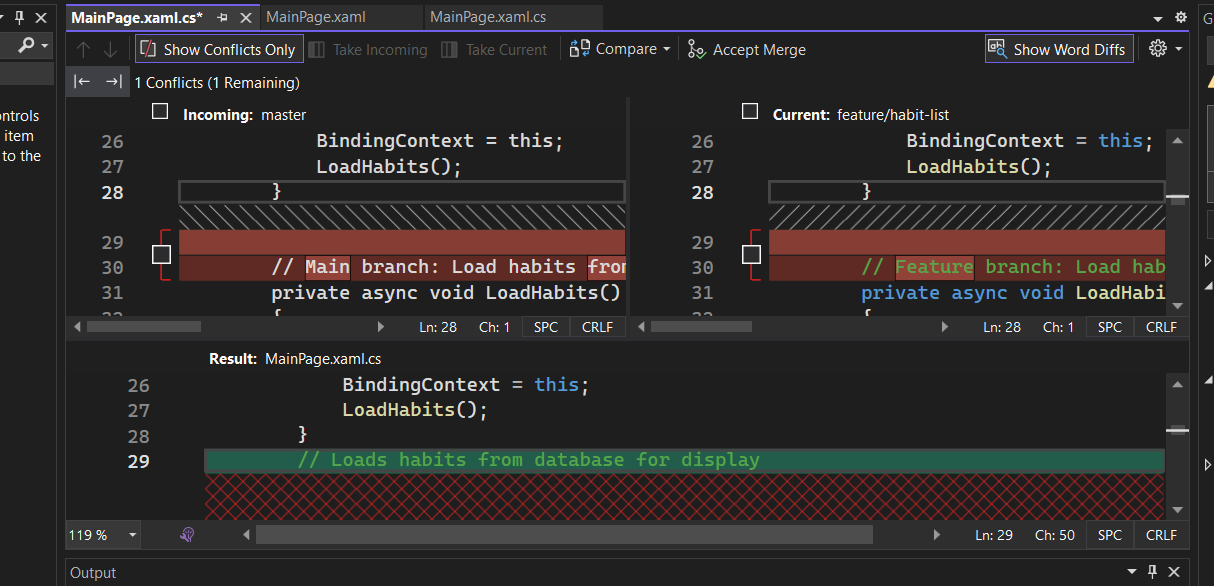


Рисунок 4.17 – Виправлення конфлікту за допомогою Visual Studio

Коміт змін у новій гілці

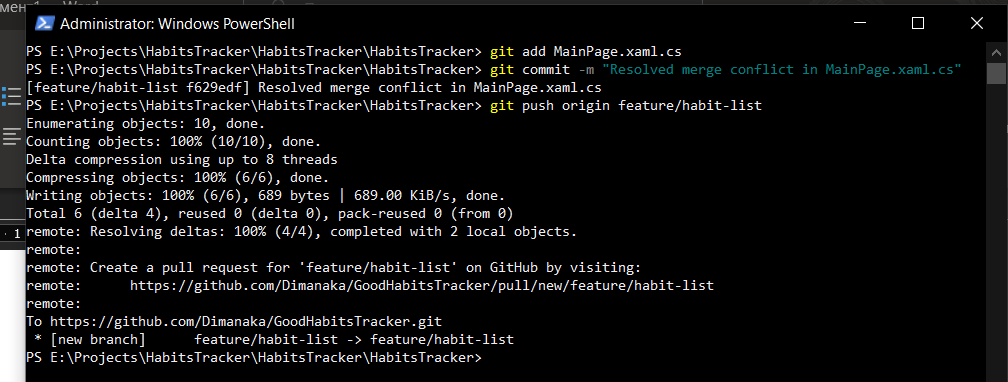


Рисунок 4.17 – Завантаження оновлень у гілкі на GitHub

Успішне злиття гілок на github:

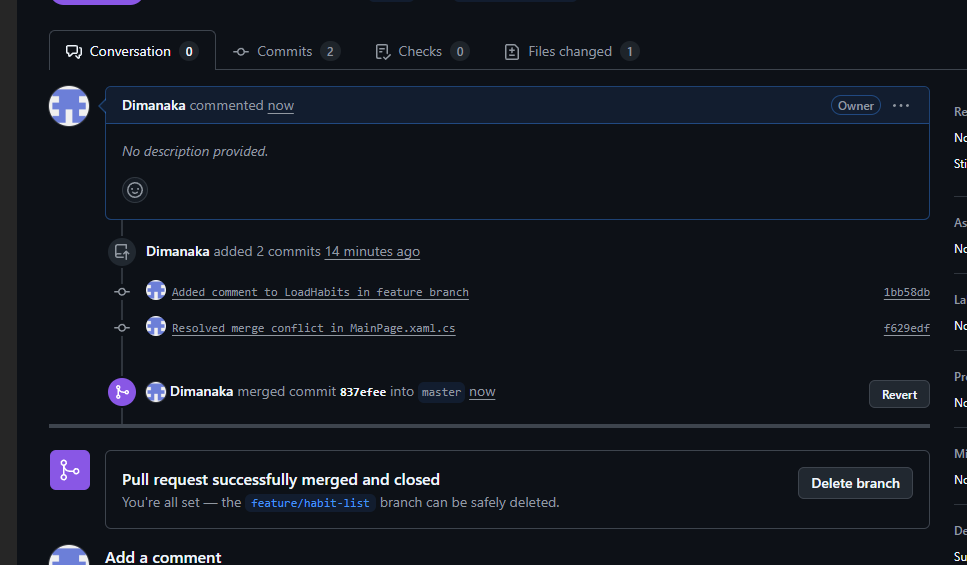


Рисунок 4.18 – Успішне злиття гілок на GitHub

**Висновок**

У результаті лабораторної роботи успішно реалізовано функціонал відображення списку звичок у мобільному застосунку "Трекер корисних звичок" із використанням .NET MAUI, SQLite і CollectionView. Функціонал протестовано на Android-емуляторі, що підтвердило коректне завантаження та відображення звичок із бази даних. Використання ObservableCollection і часткове впровадження MVVM забезпечило динамічне оновлення інтерфейсу.

Робота з Git дозволила поглибити навички управління версіями. Інструменти git status, git log, git diff і git blame допомогли проаналізувати стан проєкту, історію змін і авторство коду. Симуляція конфлікту злиття та його вирішення через git merge і ручне редагування файлу дала практичний досвід управління конфліктами, що є важливим для командної розробки.