МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

БУДІВНИЦТВА І АРХІТЕКТУРИ

Кафедра інформаційних технологій

**Лабораторна робота № 7**

З дисципліни: Інструментальні засоби розробки програмного забезпечення

Виконав:

Студент групи ІСТм-24

Йовенко Дмитро

Київ-2025

**Мета:** інтеграція DevOps-практик у проєкт "Трекер корисних звичок" за допомогою GitHub Actions.

**Постановка задачі**

1. Налаштувати CI для автоматичної компіляції та тестування при внесенні змін до репозиторію.
2. Налаштувати CD для автоматизації компіляції, тестування та розгортання застосунку.
3. Автоматизувати створення бінарних файлів.
4. Інтегрувати інструменти моніторингу помилок і автоматизації звітності.
5. Оформити звіт із описом виконаних дій, труднощів і висновків.

**Хід роботи**

**1.1. Налаштування неперервної інтеграції (CI)**

Для автоматизації компіляції та тестування створено робочий процес ci.yml у .github/workflows. Процес запускається при push до гілок master і feature/\* або при pull\_request до main. Кроки включають:

* Клонування репозиторію.
* Встановлення .NET 8.
* Відновлення залежностей (dotnet restore).
* Компіляцію рішення (dotnet build).
* Виконання юніт-тестів із HabitsTracker.Test (dotnet test).

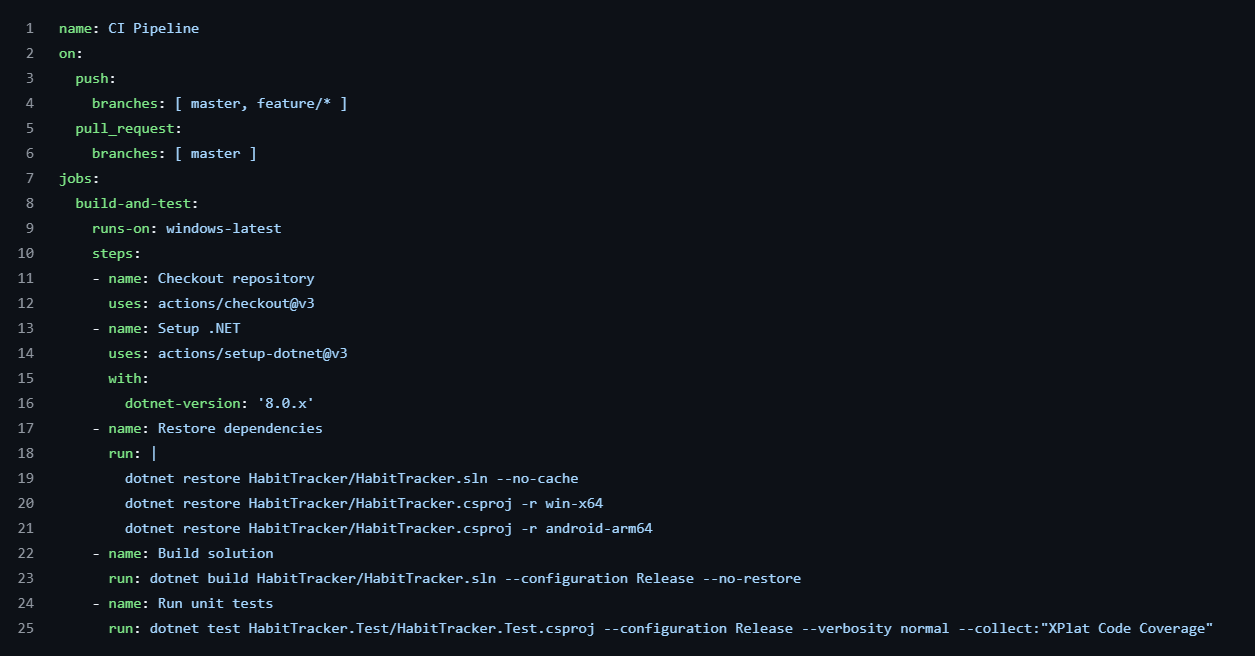


Рисунок 7.1 – Розроблений CI Pipeline

Для роботи цього робочого процесу у HabitTracker.csproj, додатково було визначено RuntimeIdentifiers для платформ Windows – win-x64, Android – android-arm64.

Приклад успічного виконання робочого процесу ci.yml зображено на рисунку 7.2.

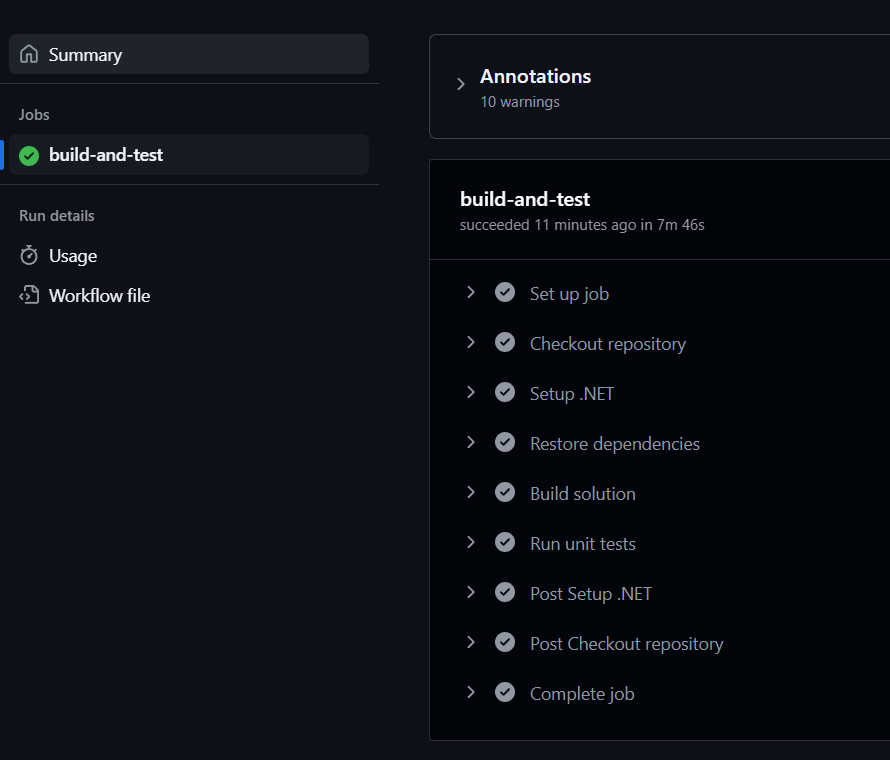


Рисунок 7.2 – Успішне виконання CI Pipeline

**1.2. Налаштування неперервної доставки (CD)**

Для автоматизації розгортання створено робочий процес cd.yml, який запускається при push до main. Кроки включають:

* Компіляцію Android APK у режимі Release (dotnet publish -f net8.0-android).
* Збереження APK як артефакт для подальшого використання.

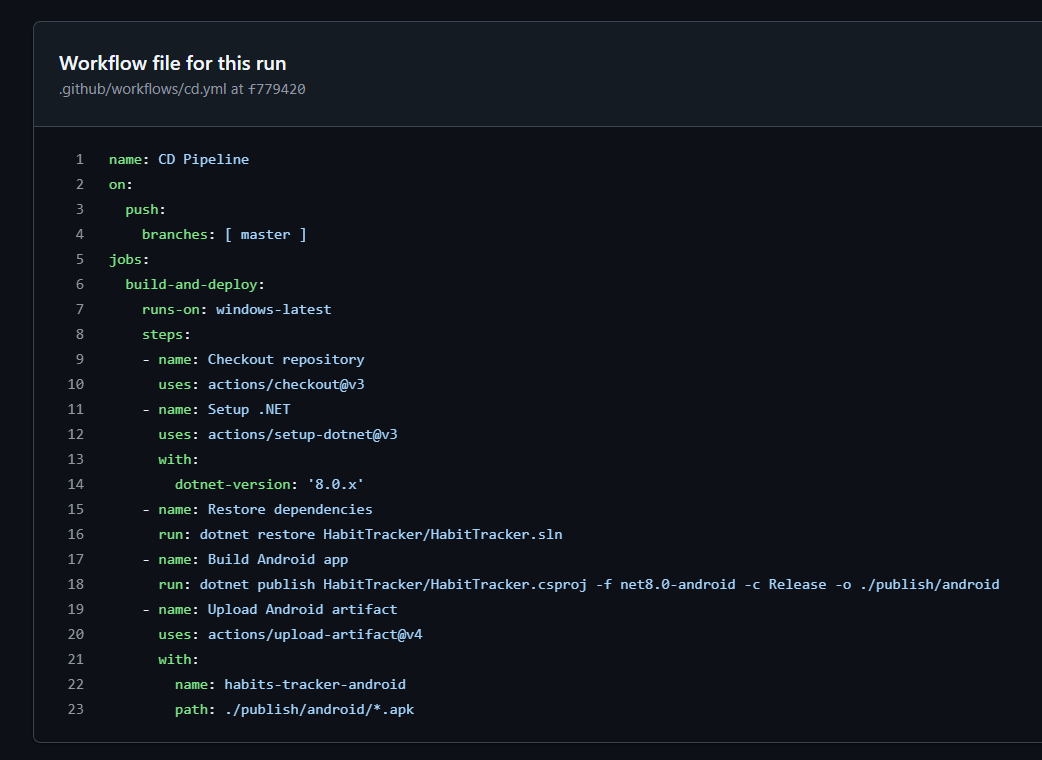


Рисунок 7.3 – Розроблений CD Pipeline

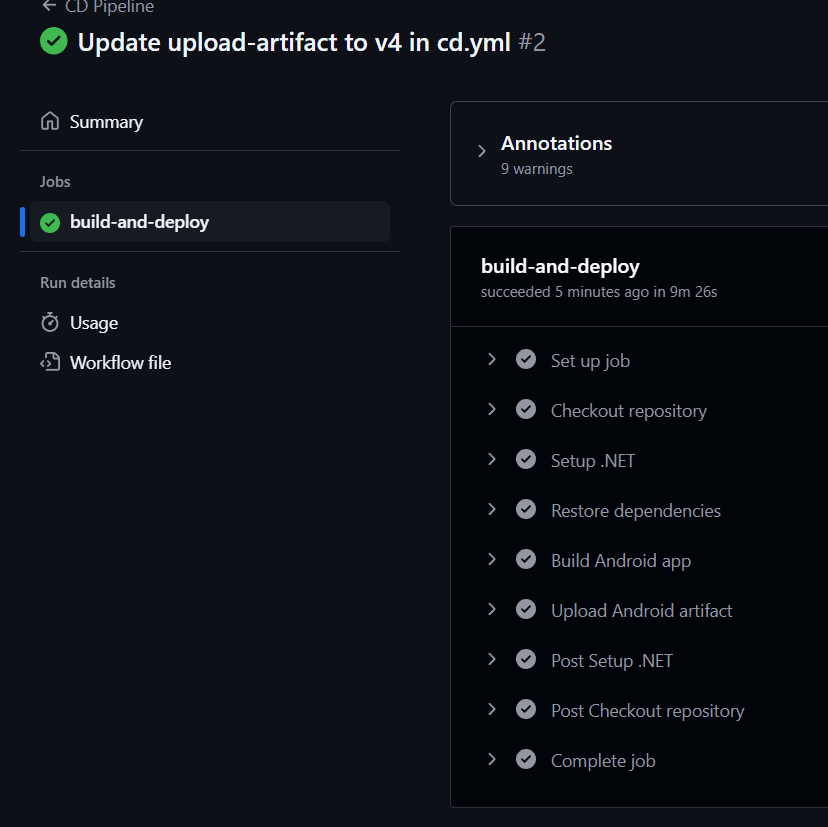
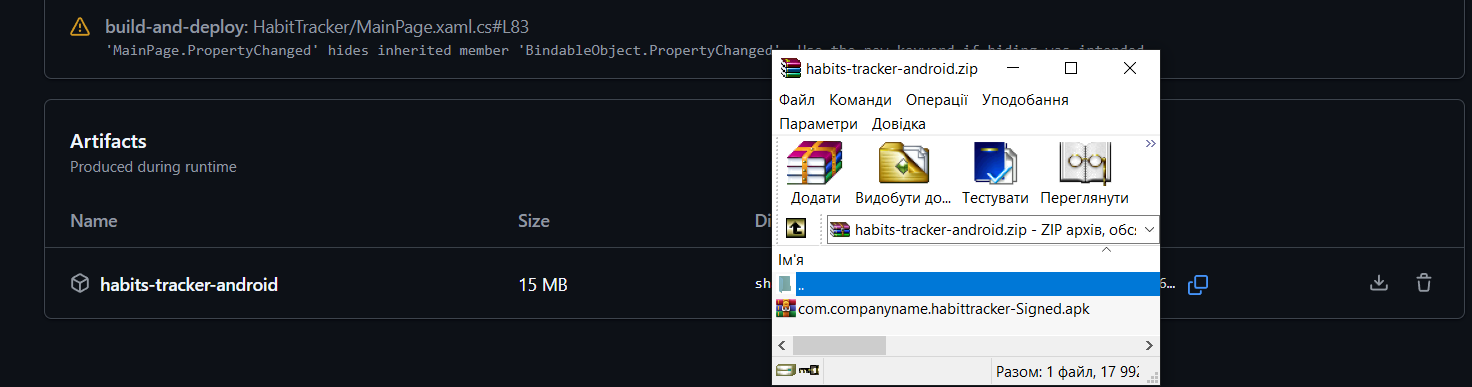


Рисунок 7.4 –Успішне виконання CD Pipeline

**1.3. Автоматизація створення бінарних файлів**

Створення APK для Android автоматизовано в cd.yml через dotnet publish. Артефакти зберігаються в GitHub Actions для ручного завантаження(рис.7.5).



Риснок 7.5 – Доступний для скачування APK у Artifacts

**1.4. Автоматизація моніторингу та звітності**

Для моніторингу помилок інтегровано Sentry через пакет Sentry.Maui. Налаштування виконано в MauiProgram.cs із DSN від Sentry. У якості тестового застосування було додано відправку повідомлення при натисканні «Add Habit»(рис.7.6).

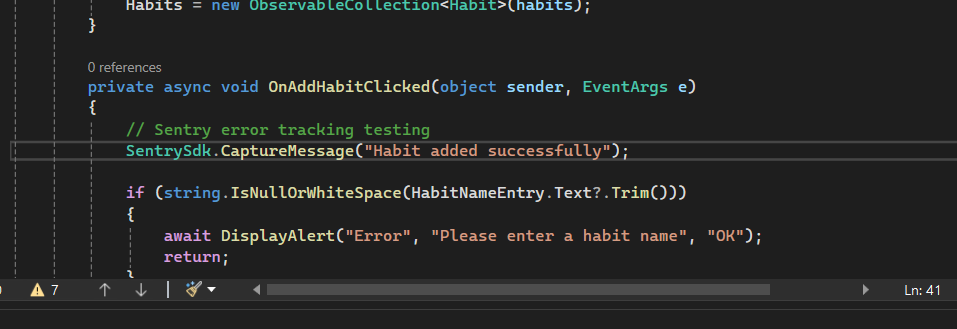


Рисунок 7.6 – Відправка повідомлення про додавання звички

Тепер коли будь-який користувач натисне “Add Habit” прийде сповіщення на панель Sentry.io(рис7.7).

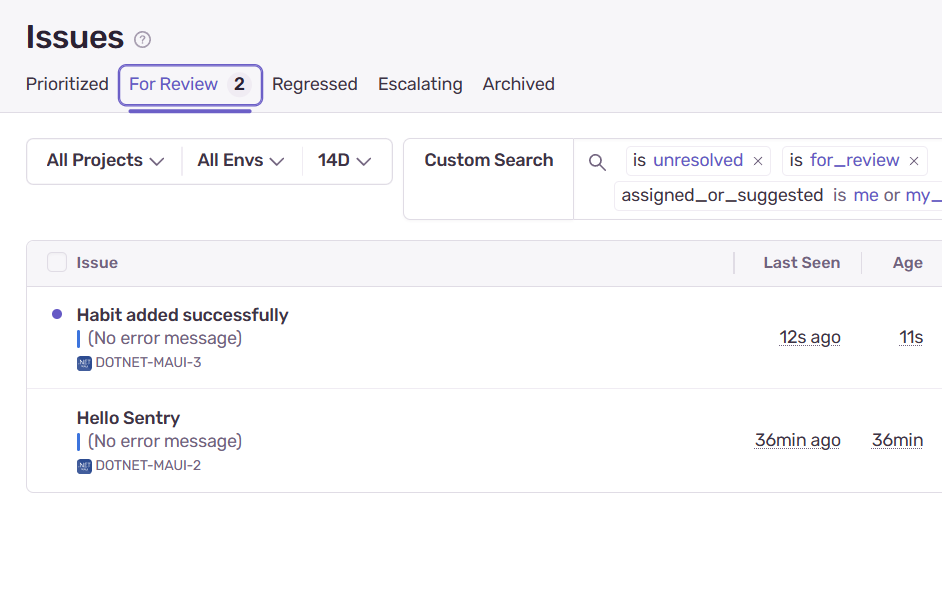


Рисунок 7.7 – Повідомлення на Sentry.io

Також сповіщення приходять на електронну пошту, що зображено на рисунку 7.8.

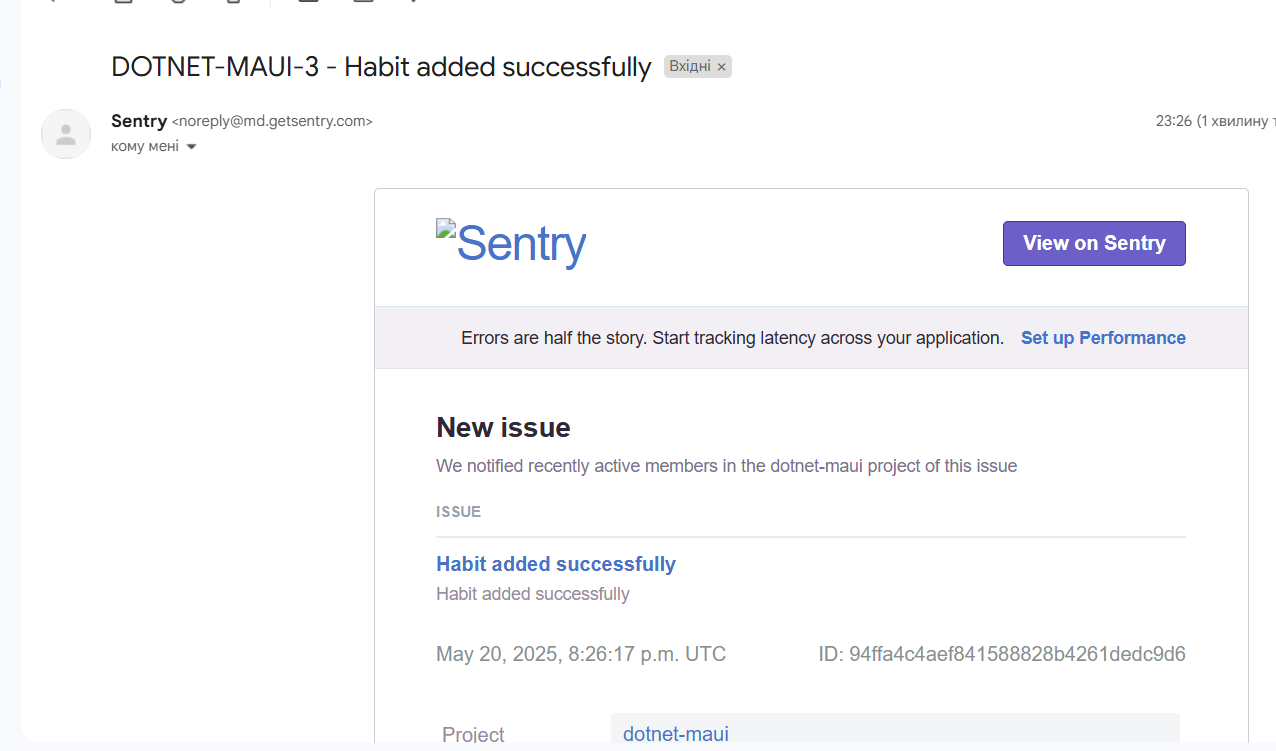


Рисунок 7.8 – Сповіщення про активність у застосунку

Функціонал цієї платформи дозволяє у подальшому розширити мождивості моніторингу, та зворотнього зв’язку.

Усі зміни було поступово закомічено та завантажено у репозиторі на GitHub(рис7.9)

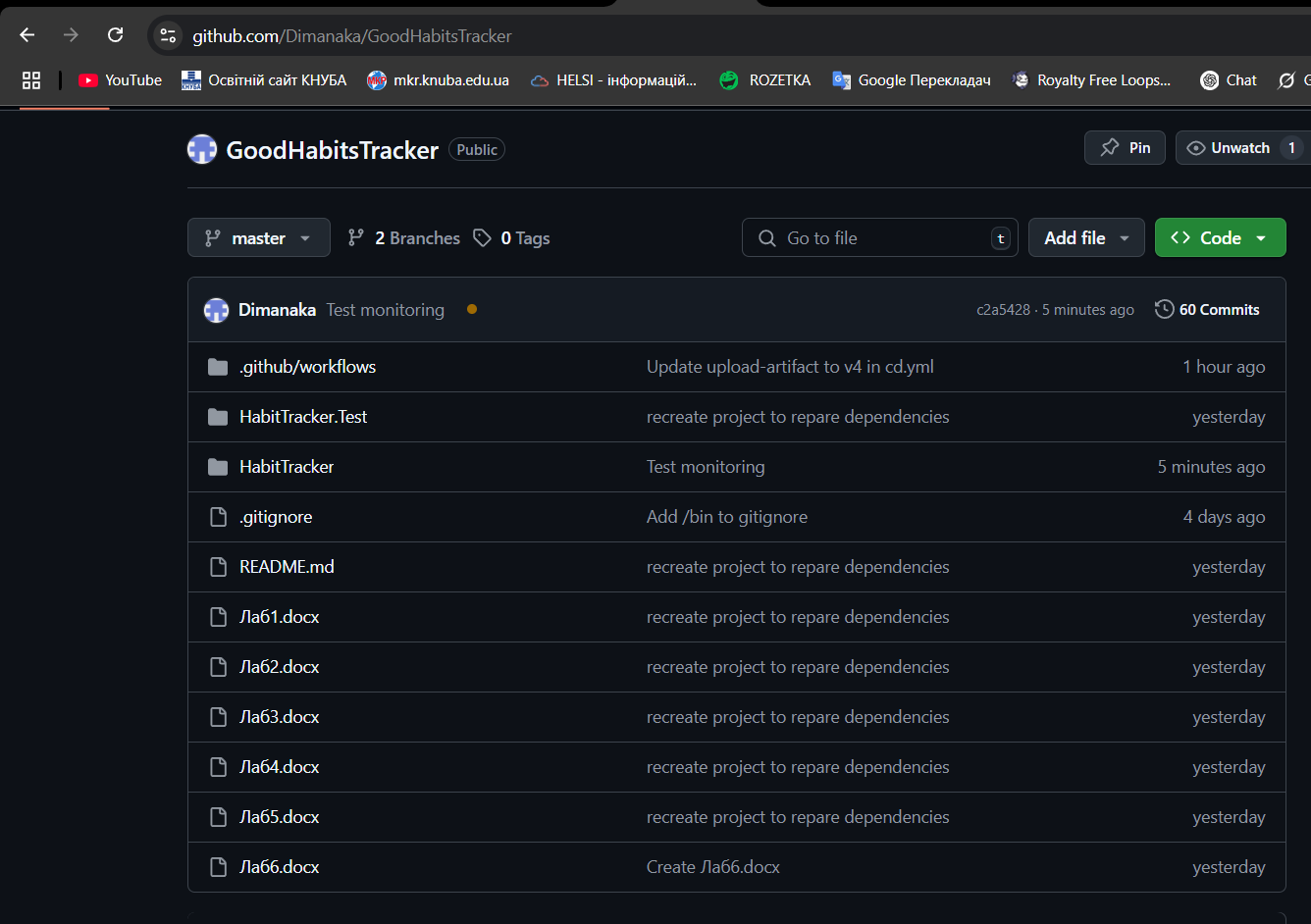


Рисунок 7.9 – Репозиторій на GitHub

**Висновок**

Лабораторна робота №7 дозволила інтегрувати DevOps-практики в проєкт "Трекер корисних звичок" за допомогою GitHub Actions. Налаштовано CI для автоматичного тестування, CD для розгортання APK, автоматизацію створення бінарних файлів і моніторинг через Sentry. Використання GitHub Actions спростило автоматизацію, а Sentry уможливили моніторинг і звітність.

Навички роботи з GitHub Actions, CI/CD і моніторингу будуть корисними для автоматизації реальних проєктів, скорочення часу випуску та підвищення якості.