

ЛАБОРАТОРНА РОБОТА №6

Системи безперервної інтеграції та розгортання

Предмет:

Технології та інструменти розробки програмного забезпечення

Виконав: Багній І.А.

Група: КП-33

Викладач: Олещенко Л.М.

Рік: 2025

Мета роботи

Отримати навички застосування типових інструментів безперервної інтеграції та розгортання програмного забезпечення (CI/CD).

Завдання роботи

1. У репозиторії GitHub, використовуючи GitHub Actions (або інший інструмент CI/CD), створити Workflow для автоматизації процесів інтеграції, розгортання програмного забезпечення. При надсиланні змін у репозиторій Workflow повинен виконувати:
 - Виведення інформації про автора
 - Автоматичну збірку додатку
 - Публікацію релізу в репозиторії GitHub
2. Протестувати працездатність Workflow, виконавши зміни у репозиторії.
3. Підготувати звіт з лабораторної роботи.

1. Створення Workflow файлу

Було створено файл `.github/workflows/ci.yml`, який відповідає за workflow. Файл було надіслано через Git у репозиторій та запущено.

1.1. Код ci.yml

```
name: CI Build and Release

permissions:
  contents: write

on:
  push:
    branches:
      - main
      - dev

jobs:
  build:
    runs-on: windows-latest

    steps:
      - name: Checkout code
        uses: actions/checkout@v4

      - name: Show author info
        run: |
          echo "Commit author:"
          git log -1 --pretty=format:"%an <%ae>"
          echo ""
          echo "Commit message:"
          git log -1 --pretty=format:"%s"

      - name: Setup Python
        uses: actions/setup-python@v4
        with:
          python-version: '3.11'

      - name: Install dependencies
        run: |
          python -m pip install --upgrade pip
          pip install -r requirements.txt

      - name: Build project
        run: |
          echo "Building SmartNotes application..."
          python -m py_compile smartnotes/app.py
          python -m py_compile smartnotes/gui.py
          echo "Build completed successfully!"

      - name: Run tests
        run: |
          python -m smartnotes.app list
          echo "Tests passed!"

      - name: Create GitHub Release
        uses: softprops/action-gh-release@v1
        with:
          tag_name: auto-release-${{ github.run_number }}
          name: Auto Release ${{ github.run_number }}
          body: |
            Automated release from CI/CD workflow

            **Commit:** ${{ github.sha }}
            **Author:** ${{ github.actor }}
            **Branch:** ${{ github.ref_name }}

            This release was automatically created by GitHub Actions.
          draft: false
          prerelease: false
        env:
          GITHUB_TOKEN: ${ secrets.GITHUB_TOKEN }
```

1.2. Опис Workflow

Workflow налаштований для автоматичного запуску при push в гілки `main` та `dev`. Він виконує наступні кроки:

- **Checkout code** - отримує код з репозиторію
- **Show author info** - виводить інформацію про автора коміту та повідомлення коміту

- **Setup Python** - налаштовує Python середовище версії 3.11
- **Install dependencies** - встановлює необхідні залежності з requirements.txt
- **Build project** - компілює Python файли додатку SmartNotes
- **Run tests** - запускає базові тести для перевірки працездатності
- **Create GitHub Release** - автоматично створює реліз у репозиторії GitHub

2. Тестування Workflow

Для перевірки працездатності workflow було виконано push змін у репозиторій.

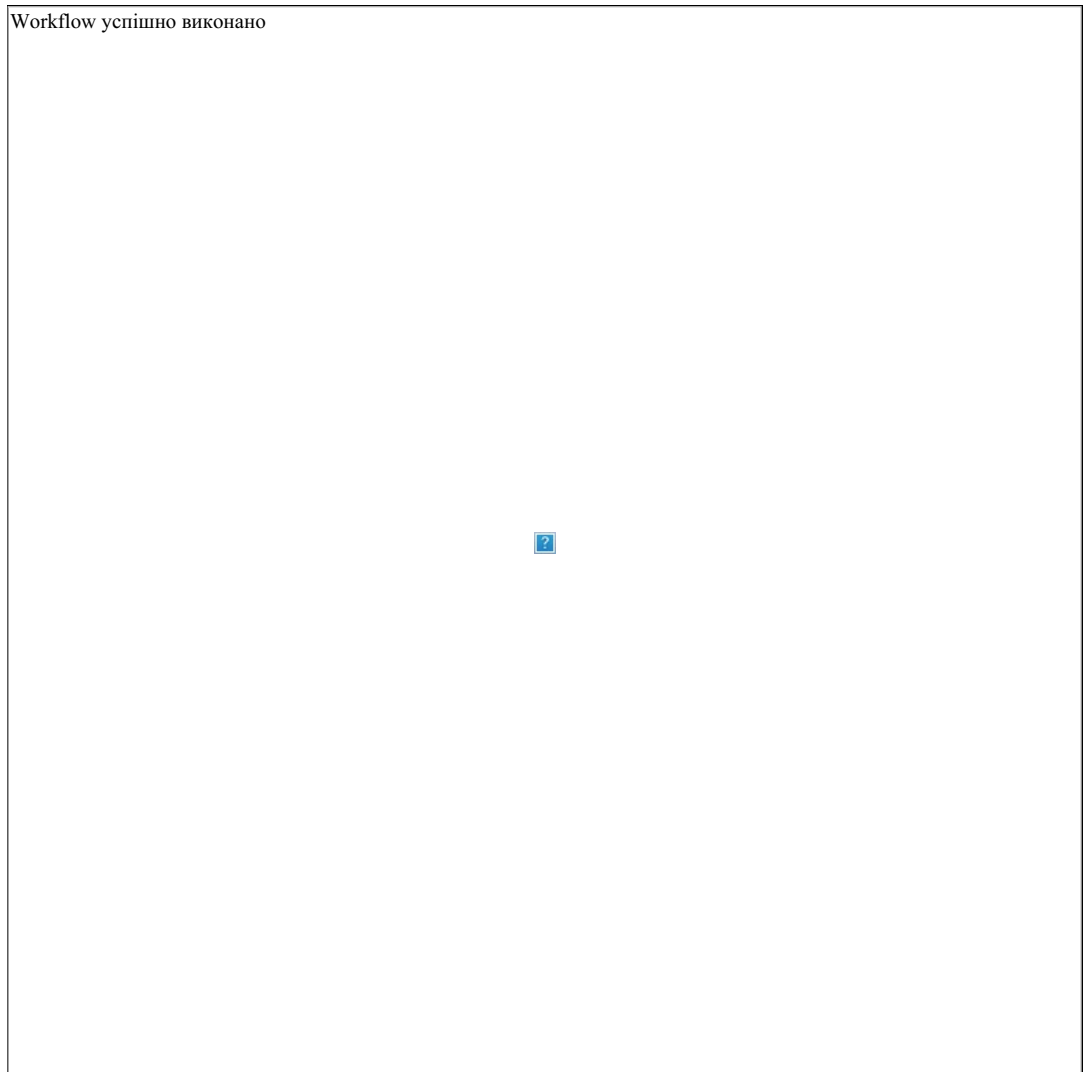
2.1. Виконання Git команд

```
$ git add .  
$ git commit -m "Add CI/CD workflow"  
$ git push origin main
```


2.2. Результати виконання Workflow

Після надсилання файлів workflow автоматично запустився. У розділі Actions GitHub можна побачити процес виконання та результати.

Рисунок 1 - Workflow успішно виконано



Інформація про автора

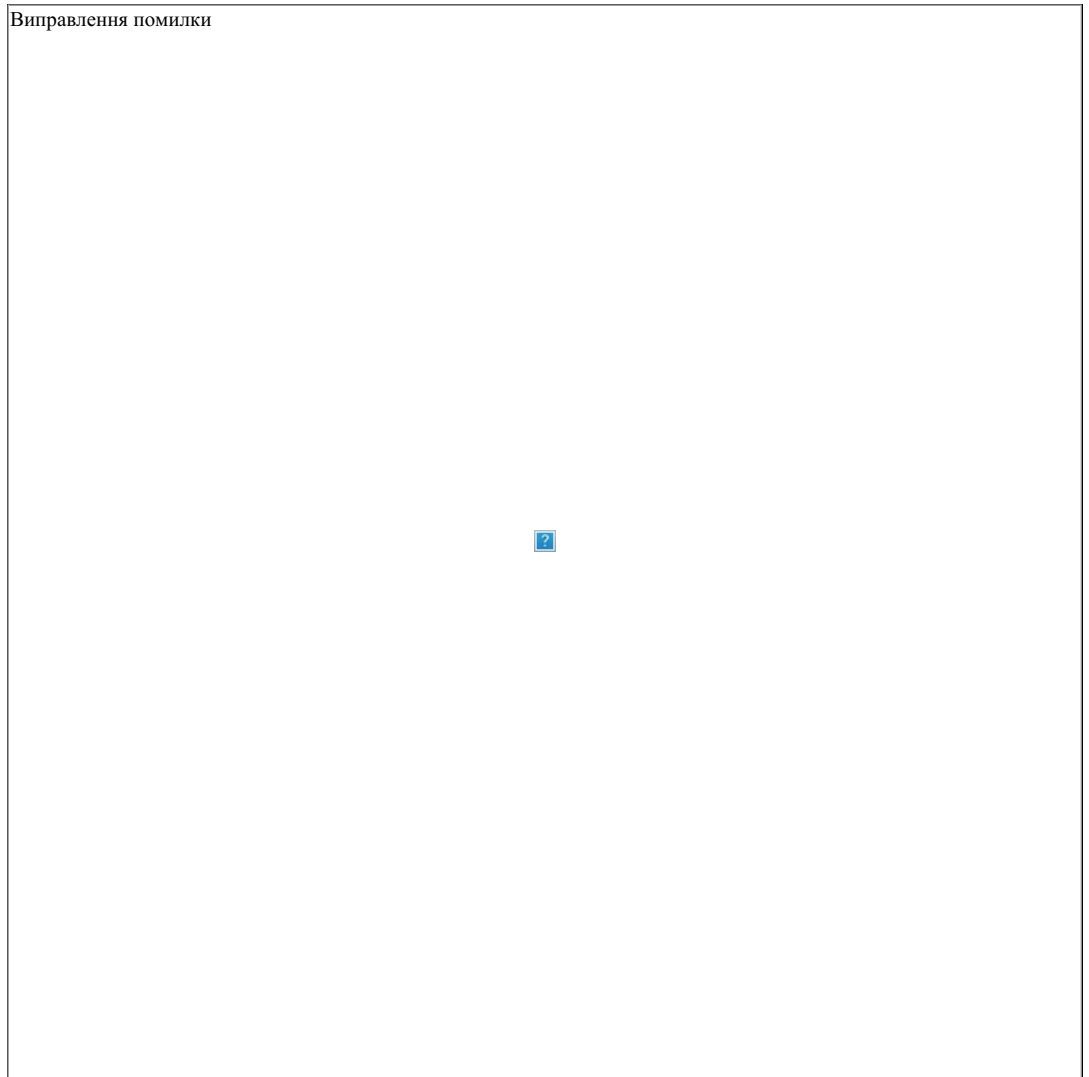


Збірка проекту



Деталі виконання





3. Висновок

Під час виконання лабораторної роботи отримано практичні навички налаштування процесу безперервної інтеграції та розгортання (CI/CD) на платформі GitHub Actions. Створено workflow файл, який автоматизує збірку Python додатку SmartNotes при кожному оновленні репозиторію. Працездатність системи перевірено шляхом внесення змін у код, автоматично генерується новий реліз з готовою програмою.

GitHub Actions забезпечує надійну автоматизацію процесів розробки, дозволяючи швидко виявляти помилки та автоматично розгортати нові версії програми.