**Описание**

| 1 | Добавих файла .github/workflows/ci.yml, за да автоматизирам проверките при push и pull request към main. |
| --- | --- |

| 2 | Workflow-ът инсталира Python, зависимостите от requirements.txt и стартира pytest, за да провери unit тестовете. |
| --- | --- |

| 3 | Успехът показва зелена отметка в GitHub. Грешките спират workflow-а и показват червена индикация. |
| --- | --- |

**JUnit Test**

| 1 | Създадох тестова функция test\_get\_status\_code\_success, която проверява дали програмата връща правилен статус код при валиден URL. |
| --- | --- |

| 2 | Тестовете се стартират автоматично от pytest. |
| --- | --- |

| 3 | Успешните тестове връщат "PASSED", а при грешка се показва описание на проблема. |
| --- | --- |

Feedback/**Reflection**

| Какво научих | Научих как да настроя GitHub Actions за автоматично тестване. |
| --- | --- |

| Трудности | Най-трудно беше симулирането на различни сценарии в тестовете, особено за обработка на грешки. |
| --- | --- |

| Подобрения | Бих добавил повече тестови сценарии и по-сложна логика за обработка на грешки. |
| --- | --- |

| Неразбираемо | В началото не разбирах добре как се използва pytest за тестване на външни зависимости. |
| --- | --- |