**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**БУДІНИЦТВА І АРХІТЕКТУРИ**

**КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**Звіт з лабораторної роботи №6**

**Дисципліни «QA тестування»**

**Варіант 23**

**Студента 2-го курсу**

**групи ІСТ-23**

**Шепель Р.С.**

**Перевірив:**

**Асистент кафедри**

**Ачкасов І.А.**

**Київ-2024**

**Лабораторна робота №6**

**Аудит якості коду та процесів розробки**

Забезпечити відповідність Python – коду стандартам якості, зокрема читабельність, ефективності та безпеки. Підвищити підтримуваність та читабельність коду, а також виявити можливі проблеми у процесах розробки.

Основний стандарт кодування для мови програмування Python що визначає рекомендації з форматування та стилю є PEP 8.

Для аудиту використав інструмент pylint для перевірки відповідності стандартів PEP 8.

Для автоматичного форматування коду відповідно PEP 8 використав інструмент Black.

Для перевірки безпеки використав Bandit.

Для використання інструменту аналізу якості коду pylint потрібно завантажити його через Windows PowerShell, команда для його завантаження «pip install pylint»

Для роботи з інструментом Black, потрібно завантажити його через Windows PowerShell, команда для його завантаження «pip install Black»

Для роботи з інструментом для перевірки безпеки Bandit, потрібно завантажити його через Windows PowerShell, команда для його завантаження «pip install Bandit».

1. Аналіз якості коду за допомогою інструменту pylint. Рис. 1.1.



Рис. 1.1. Аналіз якості коду

Після аналізу якості коду він має оцінку 5.71/10, що свідчить про наявність багатьох невідповідностей PEP 8.

1. Застосування автоматичного форматування коду Black. Рис. 1.2.

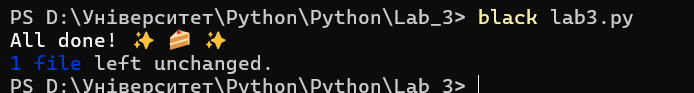


Рис. 1.2. Застосування інструменту Black.

Після використання інструменту Black. Маємо такі результати «All done!» наш файл було від форматовано під стандарти PEP 8.

1. Повторний аналіз якості коду після використання автоматичного форматування Black. Рис. 1.3.

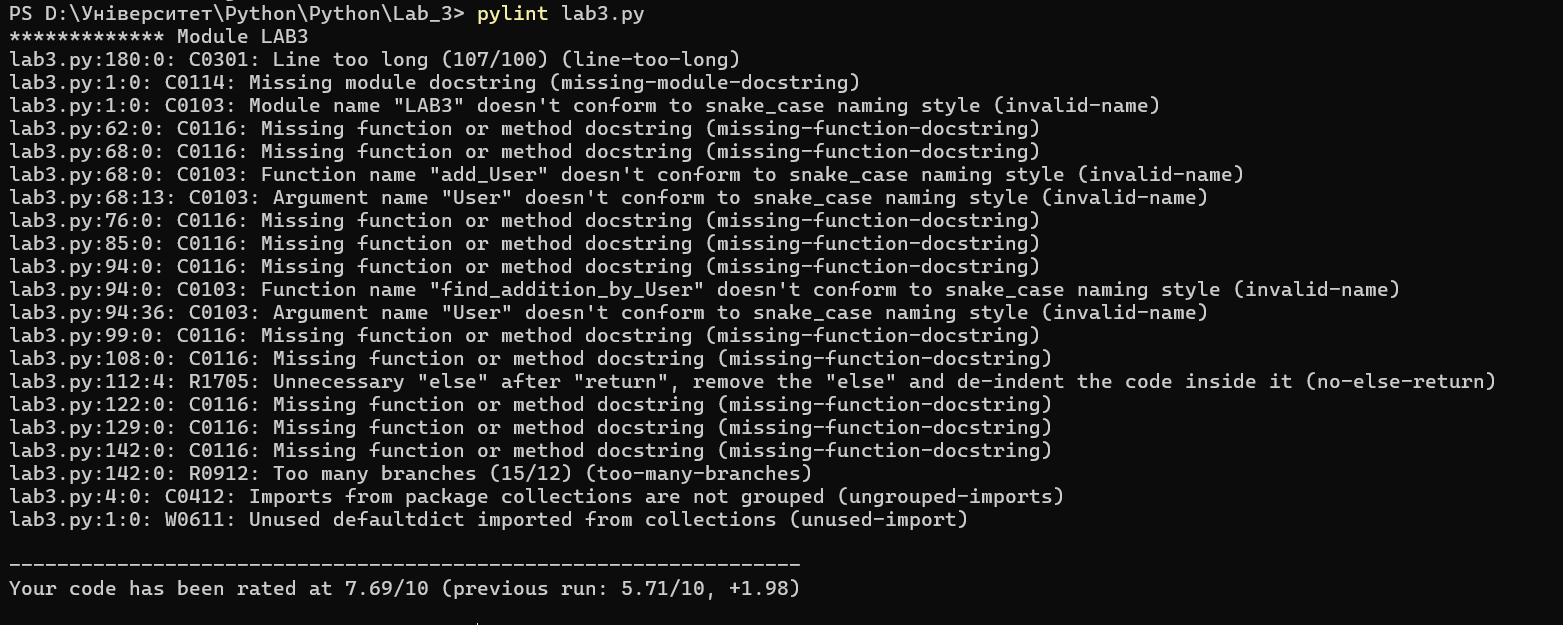


Рис. 1.3. повторне використання інструменту для аналізу якості коду

Після використання інструменту для автоматичного форматування коду Black. Маємо оцінку коду 7.69 що є краще ніж попередні результат 5.71. Тобто код на мові Python файл «lab3.py» свідчить про покращення коду стандарту РЕР 8.

1. Використання інструменту Bandit для аналізу безпеки Python-коду

Для використання інструменту треба ввести команду через PowerShell

«bandit -r D:\...» де параметр -r вказує bandit, що потрібно рекурсивно перевірити всі файли Python у вказаній директорії. D:\... шлях до проекту. Рис. 1.4.

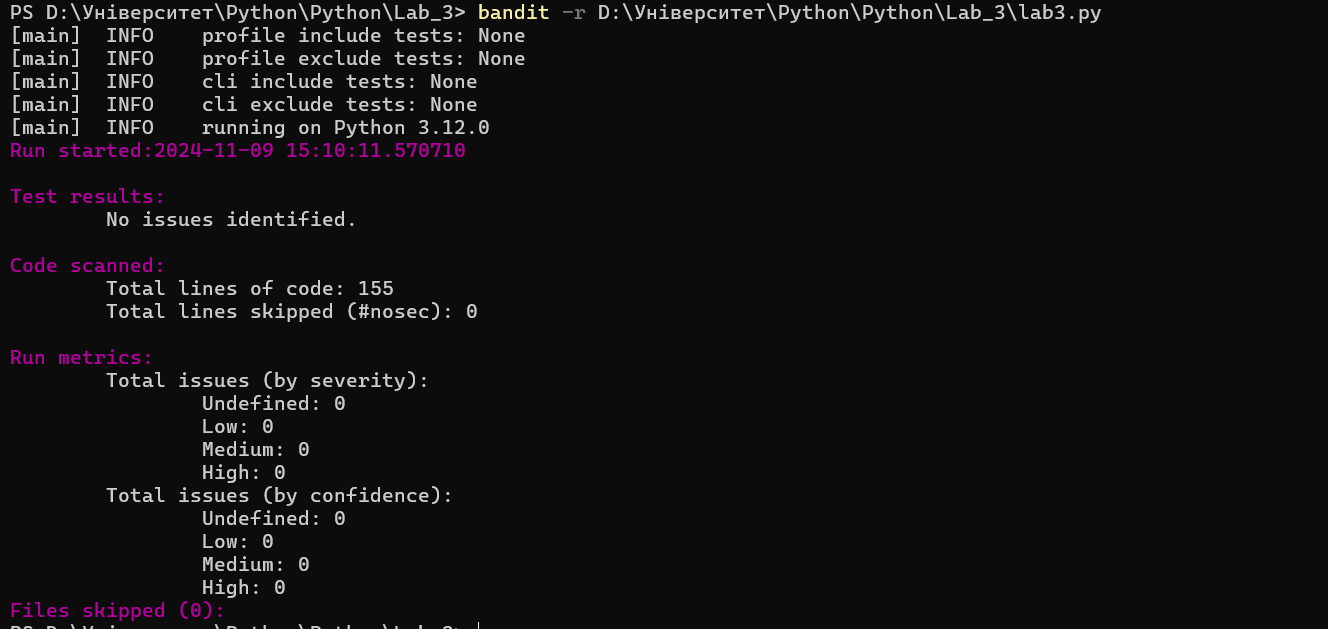


Рис.1.4 Запуск інструменту bandit

Інструмент bandit має декілька корисних параметрів: -1(показати проблеми середнього рівня ризику); -11(показати тільки високо-ризикові проблем); -111(показати всі виявлені вразливості.);

Для створення звіту у форматі json. «bandit -r D:\... -f json -o b\_rep.json». де -f для визначення формату, -о для визначення імені файлу.

Після тестування кожного параметру bandit маємо такі результати. Рис.1.5-1.7

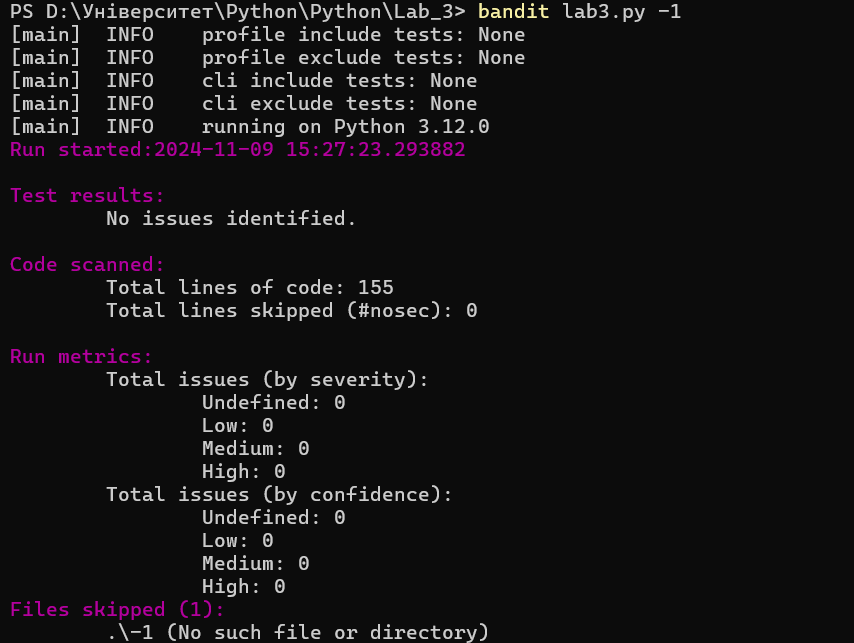


Рис.1.5 Результат тестування з параметром -1

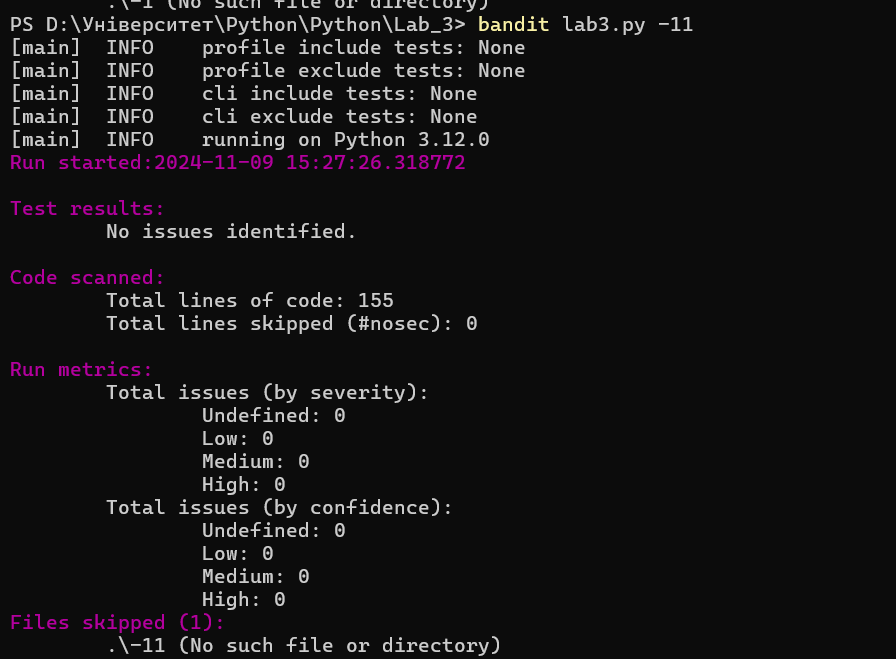


Рис.1.6 Результат тестування з параметром -11

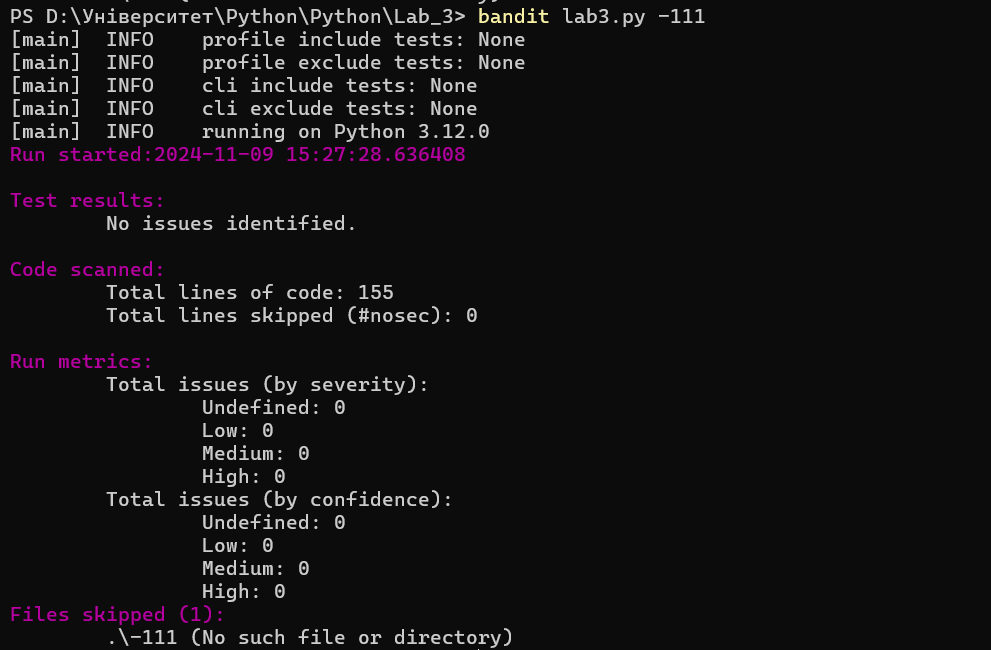


Рис.1.7 Результат тестування з параметром -111

Створення звіту в json файлі. Рис.1.8.

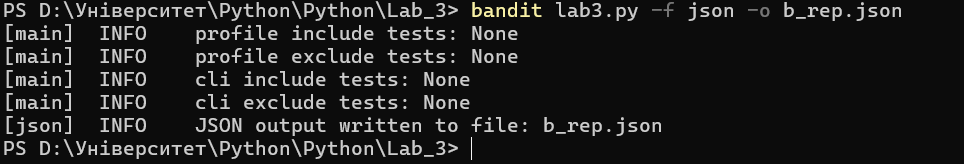


Рис.1.8 звіт в форматі json

Після виконання команди було створено звіт у форматі b\_rep.json. рис.1.9



Рис.1.9. Звіт у файлі

Звіт показує, що у файлі Python-коду немає виявлених проблем, помилок або вразливостей, статичний аналіз завершився успішно без зауважень.

Ми аналізували процеси розробки, включаючи керування версіями, тестування та впровадження, та розробляли рекомендації для їх поліпшення.

На основі результатів аудиту було розроблено конкретні рекомендації для вдосконалення якості коду та процесів розробки і склали план дій для їх впровадження.