

CodeBeautifier – Asystent czytelności kodu z automatyczną aktualizacją Git

Grupa projektowa 3J

31 maja 2025

Tytuł projektu

CodeBeautifier – Asystent czytelności kodu z automatyczną aktualizacją Git

Dane członków grupy 3J

- Kajetan Szymczak, Nr albumu: 95522
- Mariusz Szmondrowski, Nr albumu: 96591
- Adrian Mikołajczyk, Nr albumu: 96192

Opis projektu

Projekt **CodeBeautifier** zakłada stworzenie narzędzia, które automatycznie analizuje i poprawia czytelność kodu Python lokalnie – **za każdym razem, gdy użytkownik uruchamia swój skrypt Python**. Dodatkowo generować będzie ewentualne sugestie co do poprawy kodu pod względem optymalizacji i potencjalnych błędów. System nie wymaga interakcji z zewnętrznymi systemami CI/CD.

Po zakończeniu działania skryptu:

- Kod źródłowy (.py) jest analizowany pod względem formatowania, komentarzy, stylu,
- Tworzony jest plik `clean.py` z wizualnie ulepszonym kodem (bez zmiany logiki),
- Tworzony jest plik `sugestie_ai.txt`, który zawiera ewentualne sugestie optymalizacji, komentarze nt. stylu lub wykrytych potencjalnych błędów, generowane przez AI (np. OpenAI),
- Wszystkie pliki są commitowane i pushowane do repozytorium Git,
- IDE (np. PyCharm) automatycznie odświeża pliki – użytkownik nie musi ich otwierać ręcznie.

Zakres działania

- Uruchamianie automatycznej analizy kodu po wywołaniu skryptu,
- Generowanie poprawionej wizualnie wersji kodu jako osobny plik,
- Generowanie pliku z sugestiami od AI dotyczącymi jakości kodu,
- Automatyczny commit i push zmian do repozytorium Git,
- Natychmiastowe odświeżenie zmian w edytorze (np. PyCharm).
- (Opcjonalnie można dostosować analizę kodu na podstawie swoich wytycznych).

Planowane narzędzia do automatyzacji

- **Black** – formatowanie kodu zgodnie ze standardami PEP8,
- **flake8 / Pylint** – analiza jakości kodu i stylu,
- **OpenAI API** – generowanie komentarzy, docstringów oraz sugestii w pliku `sugestie_ai.txt`,
- **Git CLI / GitPython** – commit i push zmian automatycznie,
- **PyCharm** – automatyczne odświeżanie zmian w edytorze,
- **beautify.py** – główny skrypt analizujący i aktualizujący kod.

Pliki wyjściowe

- `my_code.py` – oryginalny kod,
- `my_code_clean.py` – wersja poprawiona wizualnie,
- `sugestie_ai.txt` – plik z sugestiami AI dotyczącymi poprawy jakości kodu,
- `log.txt` – log z działania narzędzi (opcjonalnie),
- Pliki commitowane i pushowane automatycznie do Git.

Cel edukacyjny

Projekt pozwala studentom praktycznie poznać:

- Automatyzację analizy i poprawy czytelności kodu,
- Obsługę GIT z poziomu Pythona (Git CLI),
- Integrację AI z codziennym workflow developerskim,
- Tworzenie narzędzi wspierających dobre praktyki programistyczne.

Podsumowanie

CodeBeautifier to lokalny, zautomatyzowany asystent poprawiający estetykę kodu Python. Dzięki integracji z Git oraz automatycznemu commitowaniu i pushowaniu zmian, wspiera profesjonalny workflow i umożliwia płynną pracę w nowoczesnym środowisku programistycznym. Dodatkowo, integracja z AI pozwala generować sugestie poprawy jakości kodu w czasie rzeczywistym.