

PROJEKT *Asystent AI dla Testera Oprogramowania*

1. Opis projektu

Projekt to prosty interfejs webowy umożliwiający użytkownikowi:

- Wczytanie pliku z kodem źródłowym lub wklejenie kodu do pola tekstowego.
- Wybór rodzaju analizy jakości kodu, np. SOLID, OOP, DRY, KISS, Prawo Demeter, wykrywanie „code smells”, generowanie testów jednostkowych i dokumentacji.
- Możliwość wpisania własnej instrukcji (promptu) dla modelu AI.
- Wystanie kodu i promptu do modelu AI Gemini 2.0 i otrzymanie analizy zwrotnej.
- Wyświetlenie odpowiedzi modelu oraz możliwość pobrania jej jako plik tekstowy.

2. Technologie użyte w projekcie

- Backend: **Python Flask**
- Frontend: HTML, CSS (własny styl), JavaScript (dynamiczne interakcje)
- AI: Google Gemini 2.0 (Google Generative AI API)
- Obsługa plików po stronie klienta: FileReader API w JavaScript

3. Struktura plików

- **index.html** – główny plik HTML interfejsu użytkownika
- **style.css** – stylizacja interfejsu (gradienty, responsywność, przyciski)
- **app.py** – serwer Flask, komunikacja z API Gemini
- **script.js** – obsługa logiki po stronie klienta (wybór analizy, wczytywanie plików, wysyłanie zapytania, pobieranie odpowiedzi)

4. Szczegóły implementacji

a) Backend (app.py)

- Konfiguracja klucza API do Google Gemini:

```
genai.configure(api_key="klucz_API")
```

```
model = genai.GenerativeModel("gemini-2.0-flash-exp")
```

- Endpoint / zwraca stronę główną (index.html).
- Endpoint /analize przyjmuje JSON z kodem, typem analizy i promptem.
- Tworzy pełny prompt łącząc bazowy prompt ekspercki z wybranym typem analizy i kodem.
- Wywołuje model Gemini, zwraca wynik w JSON.
- Obsługa błędów i zwracanie komunikatów w przypadku problemów.

b) Frontend (index.html + style.css + script.js)

- Umożliwia wczytanie pliku lub wklejenie kodu.
- Przyciskami wyboru analiz ustawia odpowiedni prompt do pola „Własna instrukcja”.
- Po kliknięciu „Wyślij do Gemini” wysyła kod i prompt do backendu przez fetch API.
- Wyświetla odpowiedź w polu tekstowym i umożliwia pobranie jej pliku.
- Stylizacja przyjazna dla oka, responsywna, gradienty i efekty hover.

5. Instrukcja uruchomienia

1. Załóż wirtualne środowisko Python.
2. Zainstaluj wymagane biblioteki:

```
pip install flask google-generativeai
```

3. Wstaw swój klucz API do app.py w miejscu `genai.configure(api_key="...")`.
4. Uruchom serwer Flask:

python app.py

5. Otwórz przeglądarkę pod adresem <http://127.0.0.1:5000/>.
6. Korzystaj z aplikacji.



