SPECYFIKACJA WYMAGAŃ OPROGRAMOWANIA

Aplikacja integrująca TMDB API

---

1. Charakterystyka oprogramowania

1.1 Nazwa skrócona

CineMate

1.2 Nazwa pełna

Aplikacja do rekomendacji i analizy popularności filmów z wykorzystaniem The Movie Database (TMDB) API

1.3 Opis i cele systemu

Celem projektu jest stworzenie aplikacji webowej umożliwiającej pobieranie, przetwarzanie oraz prezentowanie danych filmowych pochodzących z serwisu The Movie Database (TMDB) za pomocą oficjalnego interfejsu API. Aplikacja ma charakter edukacyjny i demonstracyjny.

System umożliwia użytkownikom przeglądanie popularnych i trendujących filmów, filtrowanie ich według gatunków oraz analizę danych takich jak popularność czy słowa kluczowe. Aplikacja pozwala na przeprowadzanie analiz biznesowych z wykorzystaniem podstawowych wskaźników popularności.

---

2. Prawa autorskie

2.1 Autorzy

Autor: Klaudia Kozakiewicz, Natalia Miklaszewska

2.2 Warunki licencyjne

Oprogramowanie wytworzone przez zespół objęte jest licencją **MIT** i może być wykorzystywane, modyfikowane oraz rozpowszechniane zgodnie z jej zapisami.

Dane filmowe wykorzystywane w systemie pochodzą z **TMDB API** i podlegają licencji **własnej licencji TMDB** (**non-commercial**, **proprietary**).

---

3. Specyfikacja wymagań

3.1 Wymagania funkcjonalne

ID	Nazwa	Opis	Priorytet
F-01	Wyszukiwanie filmów	Wyszukiwanie filmów po tytule z podpowiedziami	1
F-02	Wyświetlanie szczegółów filmu	Wyświetlanie szczegółów (plakat, ocena, czas trwania, gatunki, opis)	1
F-03	Ranking filmów	Wyświetlanie listy TOP filmów wg kategorii	1
F-04	Filtrowanie	Filtrowanie filmów według dekady premiery, oceny, gatunku	1
F-05	Dashboard analityczny	Prezentowanie wykresu popularności, ocen i dominujących gatunków	2
F-06	Analiza finansowa	Prezentowanie budżetu, przychodów oraz ROI filmów	2
F-07	Rekomendacje	Prezentowanie rekomendacji na podstawie wybranego filmu	1
F-08	Nawigacja wielostronicowa	Przechodzenie pomiędzy stronami przy pomocy przycisków	1
F-09	Cache danych	Buforowanie odpowiedzi z TMDB API w celu ograniczenia liczby zapytań i skrócenia czasu odpowiedzi	2

3.2 Wymagania pozafunkcjonalne

ID	Nazwa	Opis	Priorytet
NF-01	Wydajność	Czas odpowiedzi interfejsu < 5s	1

NF-02	Użyteczność	Intuicyjny interfejs, przyciski, podpowiedzi	1
NF-03	Bezpieczeństwo	Ochrona klucza API	1
NF-04	Język	Obsługa języka polskiego (pl-PL)	1
NF-05	Dostępność	Aplikacja dostępna jest poprzez przeglądarkę internetową	1
NF-06	Kompatybilność	Poprawne działanie aplikacji w różnych przeglądarkach (Chrome, Firefox, Edge)	2
NF-07	Skalowalność	Możliwość obsługi wielu użytkowników jednocześnie	2

---

4. Architektura systemu

4.1 Architektura rozwoju

- Kod źródłowy tworzony w Pythonie 3.10+
- Lokalny development przy użyciu środowiska IDE (VSCode)
- Testy lokalne przy użyciu Streamlit i pliku `.env` dla zmiennych środowiskowych
- Zarządzanie pakietami przez `requirements.txt`
- Kontrola wersji za pomocą repozytorium GitHub
- Budowanie i testowanie funkcjonalności w środowisku lokalnym przed wdrożeniem na Streamlit Cloud
- Aplikacja nie wymaga instalacji po stronie użytkownika końcowego.

4.2 Architektura uruchomieniowa

- Użytkownik wchodzi na stronę aplikacji: https://inzynieria-oprogramowania-k4grvgfz3appppcnhaqvdndw.streamlit.app/

- Streamlit Cloud hostuje backend Pythona i UI
- Backend komunikuje się z **TMDB API** poprzez HTTP, używając klucza API ukrytego w zmiennych środowiskowych (Secrets)
- Dane otrzymane z TMDB są przetwarzane w Pythonie przy użyciu bibliotek
- Wyniki (tabele, wykresy, listy rekomendacji) wysyłane są do przeglądarki użytkownika

4.3 Prezentacja technologii

Technologia	Przeznaczenie
Python 3.10+	Język programowania
Streamlit	Serwer aplikacji
TMDB API	Źródło danych
Pandas	Przetwarzanie danych
Altair	Wykresy
Requests	Komunikacja HTTP
Streamlit Searchbox	Autocomplete wyszukiwania
Datetime	Obsługa dat
OS	Obsługa zmiennych środowiskowych (TMDB API key)

---

5. Testy

5.1 Scenariusze testowe

ID	Scenariusz	Warunek	Oczekiwany wynik
T-01	Wyszukiwanie filmu	Użytkownik wpisuje nazwę filmu, np. “Shrek”	Lista podpowiedzi z tytułami

T-02	Przejście do szczegółów filmu	Kliknięcie filmu z wyszukiwarki	Strona szczegółów filmu
T-03	Filtrowanie TOP filmów	wybór np. “Najwyżej oceniane” + “Dramat”	Lista 20 filmów spełniających kryteria
T-04	Dashboardy filmów	Wyświetlenie dashboardów dla wybranej zakładki	Wykres słupkowy + metryki
T-05	Analiza finansowa	Przejście do zakładki Analiza	Wykres budżet vs przychody, ROI
T-06	Rekomendacje	Wybranie filmu i kliknięcie opcji rekomendacji	Lista rekomendowanych filmów
T-07	Uruchomienie aplikacji w sieci	Wejście na stronę Streamlit (aplikacja opublikowana w internecie)	Strona aplikacji wczytuje się poprawnie, UI działa
T-08	Sprawdzanie wczytania API key	Otwarta aplikacja na Streamlit Cloud, zmienna TMDB_API_KEY ustawiona	Aplikacja poprawnie korzysta z TMDB API, brak błędów

5.2 Sprawozdanie z testów

ID	Przebieg	Wynik	Status
T-01	Wpisano film “Shrek” w polu wyszukiwania	Wyświetliła się lista podpowiedzi z tytułami	Zrealizowano
T-02	Kliknięto film z wyszukiwarki	Otworzyła się strona szczegółów filmu “movie.py”	Zrealizowano
T-03	Wybrano “Najwyżej oceniane” + “Dramat”	Lista 20 filmów spełniających kryteria	Zrealizowano
T-04	Wejście w zakładkę “Popularność”, “Oceny”, “Gatunki”	Pojawił się odpowiedni wykres słupkowy i metryki	Zrealizowano
T-05	Analiza finansowa	Przejście do zakładki Analiza	Zrealizowano
T-06	Wybrano film “Shrek” i kliknięto “szukaj rekomendacji”	Wyświetlona lista rekomendowanych filmów	Zrealizowano

T-07	Otworzono opublikowaną aplikację Streamlit w przeglądarce	Strona wczytuje się poprawnie, wszystkie elementy UI działały	Zrealizowano
T-08	1. Na początku po otwarciu aplikacji pojawił się komunikat błędu o braku API key 2. Wprowadzono klucz w Streamlit Cloud 3. Odświeżono aplikację	Aplikacja poprawnie wczytała TMDB API, wywołania API działały, brak błędów	Zrealizowano