

Dokumentacja Techniczna Aplikacji Fullstack

Nazwa Aplikacji: Movie recommendations **Wersja:** 0.0.1 **Data Utworzenia:** 11.11.2024 **Autor:** Stefan Grzelec, Maks Buhai **Repozytorium:** <https://github.com/Spoky03/movie-recommend>

1. Wprowadzenie

1.1 Opis projektu

Aplikacja do oceniania i otrzymywania rekomendacji filmów na podstawie historii oglądania.

1.2 Cele aplikacji

- Ocenianie filmów.
 - Rekomendacje
-

2. Architektura Systemu

2.1 Ogólny Przegląd

Diagram przedstawiający ogólną architekturę aplikacji (np. diagram komponentów lub przepływu danych).

2.2 Technologie

- **Frontend:** --
- **Backend:** Express.js
- **Baza Danych:** MongoDB

2.3 Wzorce Projektowe

Krótki opis kluczowych wzorców projektowych używanych w aplikacji (np. MVC, MVVM, CQRS).

3. Frontend

4. Backend

4.1 Struktura Plików

src/

- middleware: np authorization
- service: zapytania do TMDB API
- index: główny plik
- config: zmienne środowiskowe i konfiguracja

4.2 Technologie i Biblioteki

- Express.js
- MongoDB
- AXIOS
- JWT
- BCRYPT

4.3 Endpoints API

Dokumentacja endpointów API

Przykład:

Metoda	Endpoint	Opis	Parametry
POST	/api/register	Tworzy nowego użytkownika	username, password
POST	/api/login	Loguje użytkownika i zwraca JWT	username, password
GET	/api/search	Wyszukuje film po nazwie w bazie danych i wykonuje zapytanie do TMDB	query
GET	/api/movies	Zwraca wszystkie filmy z bazy danych	-
GET	/api/similar	Zwraca podobne filmy na podstawie ID	movieId
GET	/api/rate	Zwraca oceny użytkownika	-
POST	/api/rate	Ocena filmu	movieId, score
GET	/api/myMovies	Zwraca ocenione filmy użytkownika	-
GET	/api/friends	Zwraca listę znajomych użytkownika	-
POST	/api/friends	Dodaje znajomego	friendUsername
DELETE	/api/friends	Usuwa znajomego	friendUsername
GET	/api/friends/mutualMovies	Zwraca wspólne filmy znajomych	-
GET	/api/friends/recommendations	Zwraca rekomendacje filmów na podstawie znajomych	friend
POST	/api/watchlist	Dodaje film do listy do obejrzenia	movieId
GET	/api/watchlist	Zwraca listę filmów do obejrzenia	-

4.4 Autoryzacja i Autentykacja

JWT, BCRYPT

4.5 Walidacja Danych

Walidacja danych wejściowych odbywa się na poziomie endpointów API oraz w bazie danych MongoDB.

5. Baza Danych

5.1 Schemat Bazy Danych

Diagram ERD (Entity-Relationship Diagram)

5.2 Opis Tabel

Opis głównych tabel w bazie danych i ich pól.

- **users**: Przechowuje informacje o użytkownikach (username, password)
- **movies**: Przechowuje informacje o filmach (id, title, original_title, etc.)
- **rating**: Przechowuje oceny filmów przez użytkowników (userId, movieId, score)
- **friends**: Przechowuje informacje o znajomych użytkowników (user, friend, status)
- **recommendations**: Przechowuje rekomendacje filmów dla użytkowników (userId, movieId, updatedAt, basedOn)
- **watchlist**: Przechowuje listę filmów do obejrzenia dla użytkowników (userId, movieId)

5.3 Relacje

- **users** -> **friends**: Jeden użytkownik może mieć wielu znajomych.
 - **users** -> **rating**: Jeden użytkownik może ocenić wiele filmów.
 - **users** -> **recommendations**: Jeden użytkownik może mieć wiele rekomendacji.
 - **users** -> **watchlist**: Jeden użytkownik może mieć wiele filmów na liście do obejrzenia.
 - **movies** -> **rating**: Jeden film może być oceniony przez wielu użytkowników.
 - **movies** -> **recommendations**: Jeden film może być rekomendowany wielu użytkownikom.
 - **movies** -> **watchlist**: Jeden film może być na liście do obejrzenia wielu użytkowników.
-

Dokument przygotowany przez: Stefan Grzelec **Data aktualizacji:** 23-01-2025