

Zespół:

Kamil Kurowski [github](#)

Mateusz Pieróg [github](#)

Projekt z baz danych

Ogólny zarys

- Serwis z filmami wzorujący się na portalu Filmweb
- Baza filmów, osób ze świata kina oraz użytkowników serwisu
- Każdy użytkownik ma swoją bibliotekę, może oznaczać jako obejrzane, ulubione, filmy "do obejrzenia" oraz oceniać w skali 1-5
- Dane filmów są prawdziwe (pochodzące z zewnętrznego API)
- Wyszukiwanie w zależności od filtrów (rok, gatunek, platforma)
- Logowanie i rejestracja
- Edycja własnego profilu
- Użytkowanie strony w sposób autoryzowany (z użyciem jwt Token)

Technologie

- Dokumentowa baza danych: MongoDB
- Frontend: React.js, Bootsrtap, HTML + CSS
- Backend: Node.js, Express.js, Mongoose, JWT

Dokumentacja

[Link do google docs](#)

Uruchomienie

Database

Zaimportować pliki z folderu /database do usługi bazodanowej MongoDB (Atlas, Compass). Kolekcje nazwać odpowiednio movies, peoples, users.

Backend

W folderze /backend utworzyć plik .env i umieścić w nim następującą zawartość:

```
ATLAS_URI=  
JWT_SECRET=  
JWT_EXPIRE=
```

gdzie

- ATLAS_URI to uri bazy danych, powinno być widoczne w Atlasie lub Compassie
- JWT_SECRET to prywatny klucz JWT Tokenu, powinien być wygenerowany dla serwera i nie udostępniany publicznie, można go wygenerować przy pomocy polecenia `node -e "console.log(require('crypto').randomBytes(256).toString('base64'));"`
- JWT_EXPIRE to czas wygaśnięcia JWT Tokenu, np.: (60min)

W folderze /backend wywołać komendę `npm i` a następnie uruchomić server poleceniem `node server.js` lub `npx nodemon server.js`. Teraz server działa lokalnie na porcie 5000.

Frontend

W folderze /frontend wywołać komendę `npm i` a następnie uruchomić poleceniem `npm start`. Serwis działa teraz lokalnie na porcie 3000 (`localhost:3000/`)

API

Do części endpointów wymagany token.

[/movie/](#)

[Link do kodu](#)

GET /movie/ zwraca wszystkie filmy

GET /movie/id/{ id } zwraca film o podanym id

GET /movie/count/ zwraca liczbę filmów

POST /movie/add/ dodaje nowy film

POST /movie/id/{ id }/rate dodaje rate filmu o podanym id

GET /movie/id/{ id }/title zwraca tytuł filmu o podanym id

DELETE /movie/if/{ id }/ pobiera wszystkie filmy

GET /movie/filters/{ attribute } zwraca wszystkie możliwe wartości atrybutu, możliwe atrybuty to: runtime, year, genre, platform

POST /movie/filtered/ zwraca filmy z uwzględnieniem filtrów przesłanych jako ciało zapytania, uwzględniając którą stronę aktualnie przegląda użytkownik

POST /movie/filtered/count/ zwraca liczbę filmów z uwzględnieniem filtrów przesłanych jako ciało zapytania

[/people/](#)

[Link do kodu](#)

GET /people/ zwraca wszystkich ludzi

GET /people/page/{ number }/ zwraca ludzi ze strony number

GET /people/count/ zwraca liczbę ludzi

GET /people/id/{ id } zwraca osobę o podanym id

POST /people/add/ dodaje osobę do bazy

POST /filtered zwraca osoby z uwzględnieniem filtrów przesłanych jako ciało zapytania, uwzględniając którą stronę aktualnie przegląda użytkownik

POST /people/filtered/count/ zwraca liczbę osób z uwzględnieniem filtrów przesłanych jako ciało zapytania

[/users/](#)

[Link do kodu](#)

[Link do kodu](#)

GET /users/ zwraca wszystkich użytkowników

POST /users/add/ dodaje nowego użytkownika

GET /users/username/ zwraca nazwę aktualnie zalogowanego użytkownika

GET /users/user/ zwraca aktualnie zalogowanego użytkownika

GET /user/profile/ zwraca profil użytkownika

POST /user/profile/{ attribute }/ aktualizuje podany atrybut z profilu użytkownika, możliwe atrybuty to: firstname, lastname, description, avatar

[/auth/](#)

[Link do kodu](#)

POST /auth/login logowanie

POST /auth/register rejestracja

[/library/](#)

[Link do kodu](#)

GET /library/ zwraca bibliotekę zalogowanego użytkownika

GET /library/{ section }/ zwraca sekcję biblioteki zalogowanego użytkownika, możliwe sekcje to towatch, favourites, seen

[/libmodifying/](#)

[Link do kodu](#)

Poniżej dentyfikator filmu oraz użytkownika przesyłane jako ciało zapytania.

POST /libmodifying/{ section }/checkstate/ zwraca informacje czy podany film znajduje się w sekcji danego użytkownika

POST /libmodifying/{ section }/add/ dodaje film do biblioteki użytkownika

POST /libmodifying/{ section }/remove/ usuwa film z biblioteki użytkownika

POST /libmodifying/rate/add/ dodaje ocene filmu

POST /libmodifying/rate/value/ zwraca wartość oceny

Struktura bazy danych

Wypełniona przykładowymi danymi

Tabela przechowująca dane Użytkownika

```
{
  "_id": {
    "$oid": "60bf83dd53f4b673682ac6c4"
  },
  "userName": "testowy",
  "email": "testowy@gmail.com",
  "password":
"$2a$10$ruH9UJzsrnH5nSi0JeNkw01huxKum00eWn3aEr34.gkpEs3kvgh7a",
  "avatar":
"https://upload.wikimedia.org/wikipedia/commons/2/29/Houghton_STC_22790_-_General_Historie_of_Virginia%2C_New_England%2C_and_the_Summer_Isles%2C_John_Smith.jpg",
  "firstName": "John",
  "lastName": "Smith",
  "description": "Opis testowy",
  "createdAt": {
    "$date": {
      "$numberLong": "1623163869601"
    }
  },
  "updatedAt": {
    "$date": {
      "$numberLong": "1623164123836"
    }
  },
  "library": {
    "toWatch": [
      {
        "movieID": "424",
        "title": "Schindler's List"
      },
      {

```

```

        "movieID": "447362",
        "title": "Life in a Year"
    },
    ],
    "favourites": [
        {
            "movieID": "278",
            "title": "The Shawshank Redemption"
        }
    ],
    "seen": [
        {
            "movieID": "278",
            "title": "The Shawshank Redemption",
            "rate": {
                "$numberInt": "4"
            }
        },
        {
            "movieID": "238",
            "title": "The Godfather",
            "rate": {
                "$numberInt": "2"
            }
        }
    ]
}

```

Tabela przechowująca dane dotyczące osób związanych z filmami

```

{
    "_id": "60bd149a954e2552b1484c2b",
    "id": 3490,
    "name": "Adrien Brody",
    "birthday": "1973-04-14T00:00:00.000Z",
    "deathday": "1970-01-01T00:00:00.000Z",
    "place_of_birth": "New York City, New York, USA",
    "functions": {
        "director": [],
        "writer": [],
        "actor": [
            {
                "title": "The Pianist",
                "id": 423
            }
        ]
    },
    "photo":
    "https://image.tmdb.org/t/p/w500/1dBItgLFBNGEXnI48VvnnN2vFaX.jpg",
}

```

"biography": "Adrien Nicholas Brody was born in Woodhaven, Queens, New York, the only child of retired history professor Elliot Brody and Hungarian-born photographer Sylvia Plachy. He accompanied his mother on assignments for the Village Voice, and credits her with making him feel comfortable in front of the camera. Adrien attended the American Academy of Dramatic Arts and LaGuardia High School for the Performing Arts in New York."

Tabela przetrzymująca dane dotyczące filmu

```
{
  "_id": "60bd0bd9954e2552b13b3dc2",
  "id": 278,
  "title": "The Shawshank Redemption",
  "date": "1994-09-23T00:00:00.000Z",
  "runtime": 142,
  "adult": false,
  "revenue": 28341469,
  "platforms": [
    {
      "name": "HBO Max",
      "logo":
        "https://image.tmdb.org/t/p/w500/aS2zvJWn9mwiC0eaaCkIh4wleZS.jpg"
    }
  ],
  "updatedAt": "2021-06-08T14:55:02.078Z",
  "rate": {
    "sum": 9,
    "amount": 2
  },
  "genre": [
    "Drama",
    "Crime"
  ],
  "plot": "Framed in the 1940s for the double murder of his wife and her lover, upstanding banker Andy Dufresne begins a new life at the Shawshank prison, where he puts his accounting skills to work for an amoral warden. During his long stretch in prison, Dufresne comes to be admired by the other inmates -- including an older prisoner named Red -- for his integrity and unquenchable sense of hope.",
  "cover":
    "https://image.tmdb.org/t/p/w500/q6y0Go1tsGEsmtFryDOJo3dEmqu.jpg",
  "budget": 25000000,
  "homepage": "",
  "directors": [
    {
      "directorName": "Anonymous director",
      "id": 4027,
      "name": "Frank Darabont",
      "photo":

```

```

"https://image.tmbd.org/t/p/w500/nvOqAQhKtXdHtczgqanoCfltsxJ.jpg"
    },
    ],
    "writers": [],
    "actors": [
        {
            "actorName": "Anonymous actor",
            "id": 504,
            "name": "Tim Robbins",
            "photo":
"https://image.tmbd.org/t/p/w500/hsCu1JUzQQ4p17uFxAVFLOs9yHh.jpg",
            "character": "Andy Dufresne"
        },
        {
            "actorName": "Anonymous actor",
            "id": 192,
            "name": "Morgan Freeman",
            "photo":
"https://image.tmbd.org/t/p/w500/oIciQWr8VwKoR8TmA1owaiZFyb.jpg",
            "character": "Ellis Boyd \"Red\" Redding"
        },
        {
            "actorName": "Anonymous actor",
            "id": 4029,
            "name": "Bob Gunton",
            "photo":
"https://image.tmbd.org/t/p/w500/rr2KDCKK4t0f5YhZibCpLCAsJxc.jpg",
            "character": "Warden Samuel Norton"
        },
        {
            "actorName": "Anonymous actor",
            "id": 6573,
            "name": "William Sadler",
            "photo":
"https://image.tmbd.org/t/p/w500/rWeb2kjYCA7V9MC9kRwRpm57YoY.jpg",
            "character": "Heywood"
        },
        {
            "actorName": "Anonymous actor",
            "id": 6574,
            "name": "Clancy Brown",
            "photo":
"https://image.tmbd.org/t/p/w500/tiuVx3mDJwWieFqNadrtyOhhMBN.jpg",
            "character": "Captain Byron T. Hadley"
        },
        {
            "actorName": "Anonymous actor",
            "id": 6575,
            "name": "Gil Bellows",
            "photo":

```

```
"https://image.tmdb.org/t/p/w500/eC0Iv2nSGnWTHdn88NoMyNOKWyR.jpg",
  "character": "Tommy"
},
{
  "actorName": "Anonymous actor",
  "id": 6576,
  "name": "Mark Rolston",
  "photo":
"https://image.tmdb.org/t/p/w500/hcrNRiptYMRXgkJ9k76BlQu6DQp.jpg",
  "character": "Bogs Diamond"
},
{
  "actorName": "Anonymous actor",
  "id": 6577,
  "name": "James Whitmore",
  "photo":
"https://image.tmdb.org/t/p/w500/kjEaGEeZaSFPL5siFi1MA9Q6Tmh.jpg",
  "character": "Brooks Hatlen"
},
{
  "actorName": "Anonymous actor",
  "id": 12645,
  "name": "Jeffrey DeMunn",
  "photo":
"https://image.tmdb.org/t/p/w500/eSouAAduXaziFpqLw59AG55jtGY.jpg",
  "character": "1946 D.A."
},
{
  "actorName": "Anonymous actor",
  "id": 92119,
  "name": "Larry Brandenburg",
  "photo":
"https://image.tmdb.org/t/p/w500/lhZhiyOPMykW7yWW3WCzBK2jIXi.jpg",
  "character": "Skeet"
}
]
}
```