

Warunki konieczne

- React
- Babel
- ESLint - nie może zwracać żadnych błędów
- Yarn

Dozwolone pluginy

- Axios
- Formik
- Wybrana biblioteka CSS

Termin: **11.06 godz.: 23:59**

W przypadku przekroczenia terminu punktacja z projektu jest zmniejszana o połowę.

Wymagania

- Interfejs
 - zarówno „przeglądarkowy” jak i (responsywny) „mobilny” (oparty o React i SCSS). Żeby wykorzystać SCSS w React zapoznaj się z biblioteką node-sass. **[DST]**
 - interfejs aplikacji powinien być przemyślany i jasno komunikować użytkownikowi, jakie działania podjął (np.: wyświetlanie aktualnego sposobu sortowania lub przycisk cofnij na podstronach). Proszę trzymać się zasady: [Nie każ mi myśleć](#) **[DST]**
 - interfejs powinien spełniać aktualne standardy stylowania. (możesz użyć gotowej biblioteki - w razie wątpliwości zapytaj prowadzącego). **[DST]**
- Funkcjonalność
 - aplikacja musi łączyć się ze stworzonym przez prowadzącego API. **[DST]**
 - aplikacja musi korzystać z komponentów funkcyjnych. Powinna być podzielona na komponenty. Komponenty klasowe będą skutkowały oceną niedostateczną z projektu. **[DST]**
 - Oceniany będzie również styl kodu
- Pobieranie danych
 - Powinna być możliwość wyświetlenia wszystkich danych
 - na liście powinny być widoczne podstawowe dane (razem z obrazkiem) **[DST]**
 - z podziałem na strony (paginacja) **[BDB]**
 - z możliwością masowego usunięcia elementów (poprzez zaznaczenie i naciśnięcie przycisku)
 - Możliwość sortowania
 - Alfabetycznie, wg. daty oraz wg. danych liczbowych **[DST]**
 - Możliwość filtrowania pól
 - o trzech różnych typach (np.: checkbox, dropdown i text oraz inne kombinacje) **[BDB]**
 - Graficzne przedstawienie danych
 - wyświetlenie filmu z okładką **[DST]**
 - wyświetlenie ratingu za pomocą gwiazdek
- Poruszanie się po aplikacji
 - Powinna być możliwość przeglądania danych w strukturze Master - detail (lista wszystkich filmów, po naciśnięciu na dany film pokazują się jego dane szczegółowe - zastosuj wyświetlanie warunkowe komponentów lub modal) **[DST]**
 - W widoku szczegółowym powinna być możliwość zagłosowania na wybrany element
 - głosowanie graficzne za pomocą gwiazdek (tych samych, które służą do wyświetlenia aktualnego ratingu) **[BDB]**
 - Widok szczegółowy powinien umożliwić edycję i usunięcie widocznego filmu po naciśnięciu odpowiednich przycisków
 - Na liście wszystkich danych powinna być możliwość dodania nowego elementu po naciśnięciu przycisku. **[DST]**
- Dodawanie i edycja danych
 - Powinna być możliwość dodawania i edycji danych **[DST]**
 - Powinien istnieć wspólny formularz dla operacji dodawania i edycji.
 - Formularz powinien spełniać wymagania stawiane przez backend (np. odpowiednie typy danych). **[DST]**
 - Oceniana będzie czytelność i funkcjonalność formularza **[DST]**
 - Wymagana jest odpowiednia walidacja danych **[DST]**
- Dodawanie elementów do ulubionych **[BDB]**
 - Powinna być możliwość dodawania wybranych elementów do ulubionych.
 - Lista ulubionych powinna być oddzielona od listy wszystkich elementów.
 - Po odświeżeniu strony lista ulubionych jest resetowana
 - Powinna być możliwość usuwania elementów z listy ulubionych.

API

API do pobrania: [pobierz](#)

Uwaga! Tabela, do której odwołuje się API może delikatnie się zmienić, mogą np. zostać dodane dodatkowe kolumny.

Backend korzysta z [postgresa](#), którego należy pobrać i zainstalować. Zapamiętaj wprowadzone hasło i port podczas instalacji. Następnie zmień dane konfiguracyjne w pliku `src/index.js`, aby móc połączyć się z bazą.