

DOKUMENTACJA PROJEKTU - FILMFLIX
Wydział: Elektrotechniki, Automatyki i Informatyki Kierunek: Informatyka Laboratorium: PAI2
Beata Kozieł Michał Olejko Grupa: 3ID11A 18.05.2020

## Instrukcja uruchomienia:

Projekt uruchamialiśmy na systemie Windows 10.

### Frontend:

Potrzebne są: Node.js oraz Visual Studio Code.

W katalogu projektu klienta (tj. frontendu) uruchomić terminal (cmd) i wpisać "npm install", który instaluje zależności w lokalnym folderze node\_modules. Następnie należy wpisać polecenie "npm run dev", aby uruchomić aplikację na pierwszym dostępnym porcie poczynając od 8080.

### Backend:

Potrzebne są: IntelliJ oraz serwer MySQL (dodatkowo korzystaliśmy z narzędzia MySQL Workbench).

W pliku application.properties uzupełnić zgodnie ze swoją bazą danych (podkreślone wartości):

```
#w razie potrzeby w linijce poniżej można zmienić port
server.port=90
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
spring.datasource.url=jdbc:mysql://$${MYSQL_HOST:localhost}:3306/filmflix?createDatabase
IfNotExist=TRUE&useUnicode=YES&characterEncoding=UTF-8&serverTimezone=CET
spring.datasource.username=username
spring.datasource.password=password
spring.datasource.initialization-mode=always
```

Następnie należy uruchomić serwer MySQL. Następnie w IntelliJ należy uruchomić VodServiceApplication.

Po uruchomieniu aplikacji należy się zarejestrować (np. <http://localhost:8080/register>), a następnie zalogować.

## Omówienie warstw:

### 1. Strona domyślna ( / ) - logowanie:

Po wprowadzeniu loginu oraz hasła, zatwierdzeniu klawiszem "Enter" lub przyciskiem "Zaloguj" zostaje wywołana metoda "submit()" z pliku LoginCard.vue (style tej strony zapisane są w Login.vue). Na początku tej metody sprawdzane jest (if) czy użytkownik uzupełnił obydwa pola, jeśli nie to wyświetlany jest komunikat "Uzupełnij puste pola". Jeśli tak nie jest (else) to metodą post sprawdzane zostaje czy taki użytkownik istnieje. W tym celu na serwerze zostaje uruchomiona metoda "login" z AuthController (controllers/AuthController.java). Za pomocą AuthenticationManager'a sprawdzana jest poprawność loginu oraz hasła. Jeśli nie istnieje taki użytkownik to rzucony zostaje wyjątek "Incorrect username or password". Taki wyjątek jest obsługiwany na froncie, gdzie wyświetlony zostanie komunikat dla użytkownika "Nieprawidłowe dane".

Jeśli logowanie zostanie przeprowadzone poprawnie to użytkownik zostanie załadowany dzięki metodzie "loadUserByUsername" z "UserService", a wygenerowany Bearer Token zostanie zapisany w plikach cookie, oraz będzie wyświetlony komunikat "Zalogowano prawidłowo", a użytkownik zostanie przekierowany za pomocą VueRouter'a na stronę /home.

Wszystkie "routes" są zapisane w pliku "routes.js".

Dodatkowo na górze strony znajduje się nagłówek z pliku LoginHeader.vue, gdzie w prawej części ekranu znajdziemy przycisk "Zarejestruj się" umożliwiający rejestrację. (1.1)

1.1. Przycisk "Zarejestruj się" poprzez VueRouter przekierowuje do strony /register z pliku Register.vue, gdzie w nagłówku znajduje się przycisk przekierowujący do logowania (shared/RegisterHeader.vue), a w pozostałej części strony RegisterCard.vue zawierający pola na e-mail, hasło oraz potwierdzenie hasła. Po uzupełnieniu tych pól i zatwierdzeniu wywoływana jest metoda "submit()". Na początku tej metody sprawdza się czy żadne z pól nie jest puste, a jeśli jest to wyświetlany jest odpowiedni komunikat. Następnie sprawdza się czy e-mail posiada znak "@", a jeśli nie to również jest wyświetlany komunikat.

Jeśli wszystko zostało wprowadzone prawidłowo to metodą post wywoływana jest metoda "register" znajdująca się w pliku AuthController (controllers/AuthController.java). Z UserService.java (/services) zostanie wywołana metoda registerUser, gdzie sprawdzane jest czy obydwa pola na hasła są wypełnione tym samym ciągiem znaków, a następnie przez UserRepository.java (/db/repositories) obiekt użytkownika jest mapowany na encję. otrzymuje swój unikalny id, oraz zapisywany do bazy danych.

Następnie jest przekierowywany na register/payment (punkt 3).

### 2. Strona główna ( /home )

Głównym plikiem obsługującym /home jest Home.vue, który importuje inne pliki takie jak:

-HomeHeader.vue,

-Banner.vue,

-Sliders.vue.

W samym Home.vue zapisanych jest kilka stylów CSS.

W pliku HomeHeader.vue (/shared/HomeHeader.vue) jest zawarty nagłówek strony /home, gdzie znajdują się przyciski: "Filmy", "Seriale", oraz rozwijany "Profil", w którym znajdziemy "konto" oraz "wyloguj się". (2.1)

W pliku Sliders.vue znajduje się cała reszta strony. (2.2).

2.1) Przycisk "profil" oraz "konto" przekierowują na /login. Po najechnięciu na "profil" rozwijane jest też oczywiście małe menu. Przycisk "wyloguj się" wywołuje logout(), które usuwa Bearer Token z plików cookie, oraz VueRouterem przekierowuje na /login.

2.2) W pliku Sliders.vue wyświetlane są listy filmów w postaci sliderów (awesome-swiper'y Vue) według odpowiednich kategorii. Filmy filtrowane są przez metodę "filterEventsByCategory" - wybiera ona odpowiednie filmy do odpowiedniego slidera. Poszczególne pola sliderów uzupełniane są również obrazkami z serwera, a poruszanie się po nich zapewniają odpowiednie buttony w kształcie strzałek na jego brzegach. Wszystkie obiekty filmów są pobierane przed załadowaniem strony metodą getMovies(), która metodą get zawierającą Bearer Token w headerze odwołuje się do metody getMovies() z MovieController.java (/controllers), gdzie za pomocą MovieService.java, metodą getMovies() za pomocą MovieRepository.java - findAll() zwracane są wszystkie obiekty z tabeli "movie". Następnie metoda getMovies() z pliku Sliders.vue są konwertowane z formatu json i zwracane. Dodatkowo przed załadowaniem strona dostaje listener, który czeka na klawisz Esc - jeśli użytkownik go naciśnie to wywołuje metodę closePopUpWindow() (2.2c).

Metody:

completeOpinions() - 2.2a

showOpinionsPopUp - 2.2b

closePopUpWindow() - 2.2c

showPopUpWindow - 2.2d

getMovies() - opisana wyżej

getOpinions() 2.2e

addOpinion() 2.2f

2.2a) Metoda ta tworzy lub odświeża listę opinii tworząc kolejne div'y dla każdej opinii.

2.2b) Tworzy div'y, przycisk 'dodaj opinie' oraz pole tekstowe. Pokazuje popup window.

2.2c) Czyści popup window z div'ów, a następnie ukrywa popup window.

2.2d) Podobnie jak showOpinionsPopUp() tworzy divy dla popup window i wyświetla go.

2.2e) Wysyła żądanie do serwera o opinie do odpowiedniego filmu i konwertuje je z JSON'a. Jako nagłówek podawany jest Bearer Token z plików cookie jako autoryzacja. Odwołuje się do metody getMovieOpinions()

(controllers/MovieController.java) przyjmującej id filmu. Za pomocą MovieService wyszukuje opinie do odpowiedniego filmu o podanym id.

2.2f) Za pomocą metody post i autoryzacji Bearer Token'em dodaje nową opinię do odpowiedniego filmu. Po stronie serwera wywoływana jest metoda addOpinion() z MovieController.java, która za pomocą OpinionService.java szuka filmu o odpowiednim id, szuka użytkownika który dodaje opinię, a następnie zapisuje ją w bazie danych.

### 3. /register/payment

Obecnie nie jest potrzebne podawanie metody płatności, aby móc się zalogować- wystarczy przejść pierwszą stronę rejestracji, a następnie przejść do logowania. Strona ta posiada nagłówek z pliku RegisterHeader.vue (/shared) , który jest taki sam jak dla strony rejestracji - posiada logo "FILMFLIX" oraz przycisk "Zaloguj się". Resztę strony zawiera RegisterPaymentCard.vue. Podobnie jak na poprzednich stronach, po wprowadzeniu danych sprawdzane jest czy pola nie są puste, zostają wyświetlone odpowiednie komunikaty, a metodą post z załączonym headerem jakim jest "Authorization: this.\$cookie.get("jwt")" wysyłany jest również BearerToken z plików cookie. Metoda jest obsługiwana ze strony serwera w UserController.java (/controlles) - payFor. Tam z UserService wywoływane jest payStreamingPlan, które odpowiedniemu użytkownikowi ustawia pole planPaid na wartość "true", a za pomocą UserRepository nadpisuje zmiany.

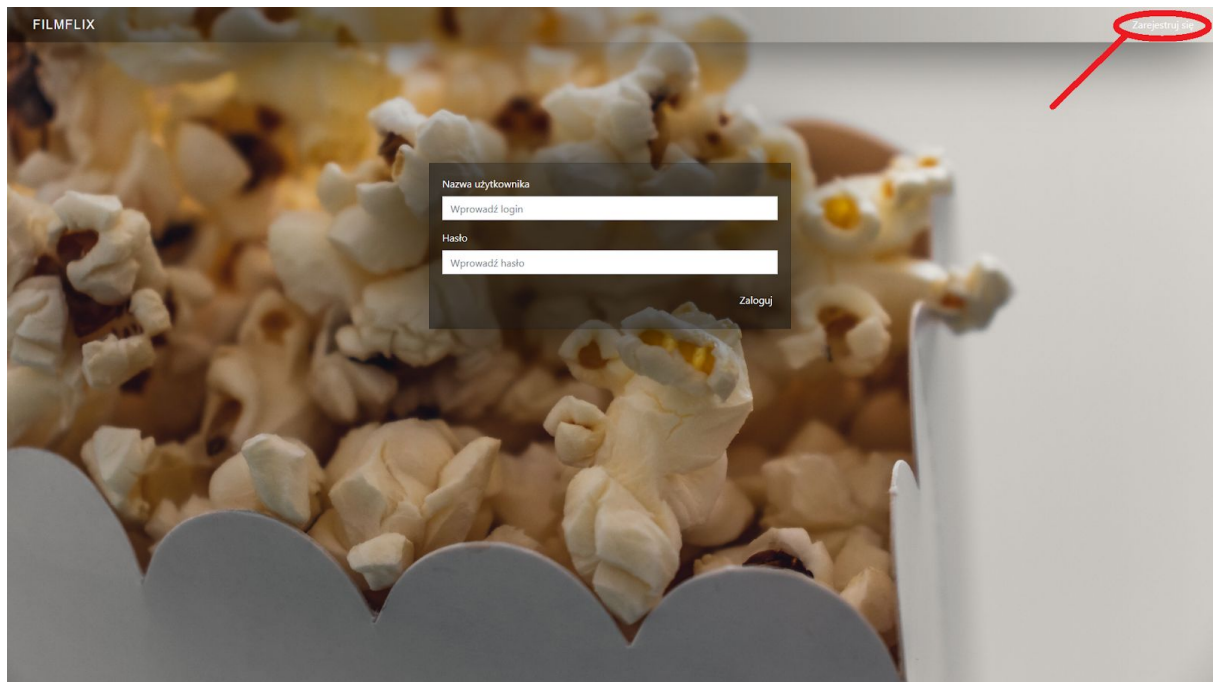
### 4. /play/:resource - komponent Player.vue

Popup window z pliku Sliders.vue zawiera przycisk "Otwórz film", który za pomocą VueRouter przekierowuje do tej strony. Komponent Player.vue zawiera przeglądarkowy odtwarzacz filmów, oraz metodę get pobierającą film z serwera - getVideo(). Obsługiwana jest na backendzie za pomocą getVideo() z MovieController.java. Pobiera ona plik z odpowiedniej ścieżki i korzysta ona z resourceRegion() z VideoService.java. resourceRegion() dzieli plik wideo na części, żeby nie był on wysyłany w całości, a właśnie w częściach. Metoda getVideo korzysta ze statusu http 206 (tj. PARTIAL\_CONTENT), aby je przesyłać. Dzięki temu wątki są używane przez krótszy czas, dając możliwość przetworzenia innych żądań nadchodzących z frontu.

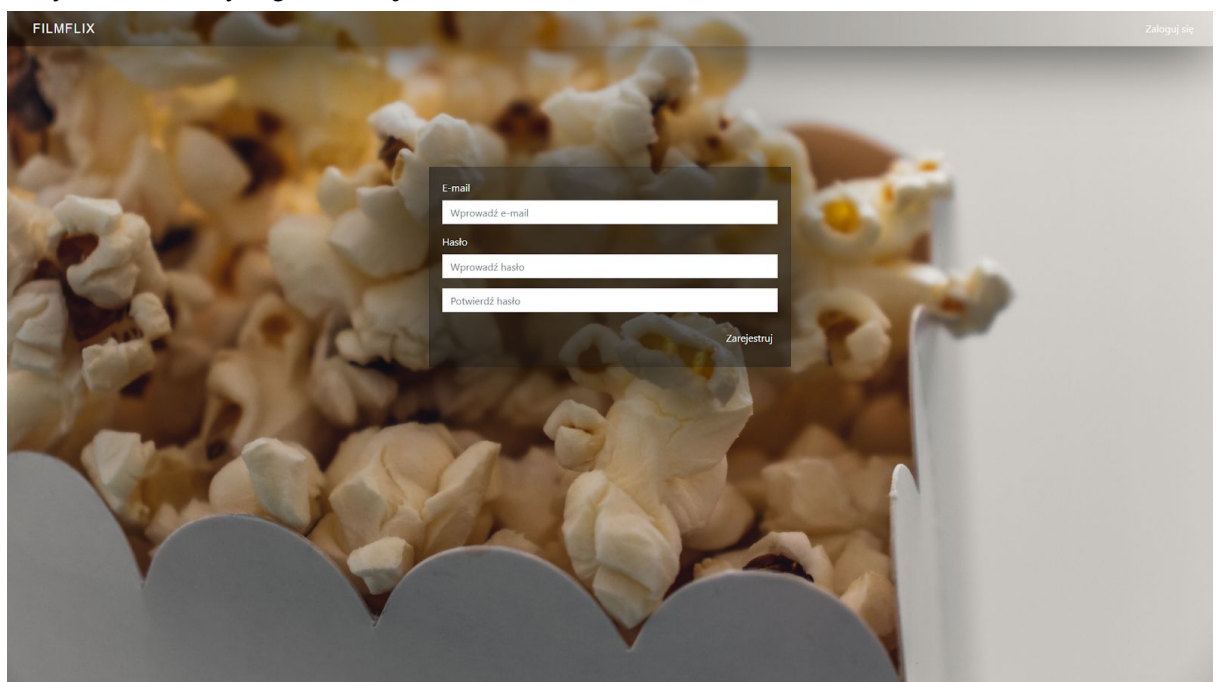
## Instrukcja dla użytkownika:

### 1. Rejestracja

Przed wszystkim, aby jako zwykły użytkownik korzystać z serwisu należy się zarejestrować pod adresem link/register, lub jeśli widzimy stronę główną, czyli logowanie to należy wybrać przycisk kryjący się pod białym napisem “Zarejestruj się”. Widać to na poniższym zrzucie ekranu:



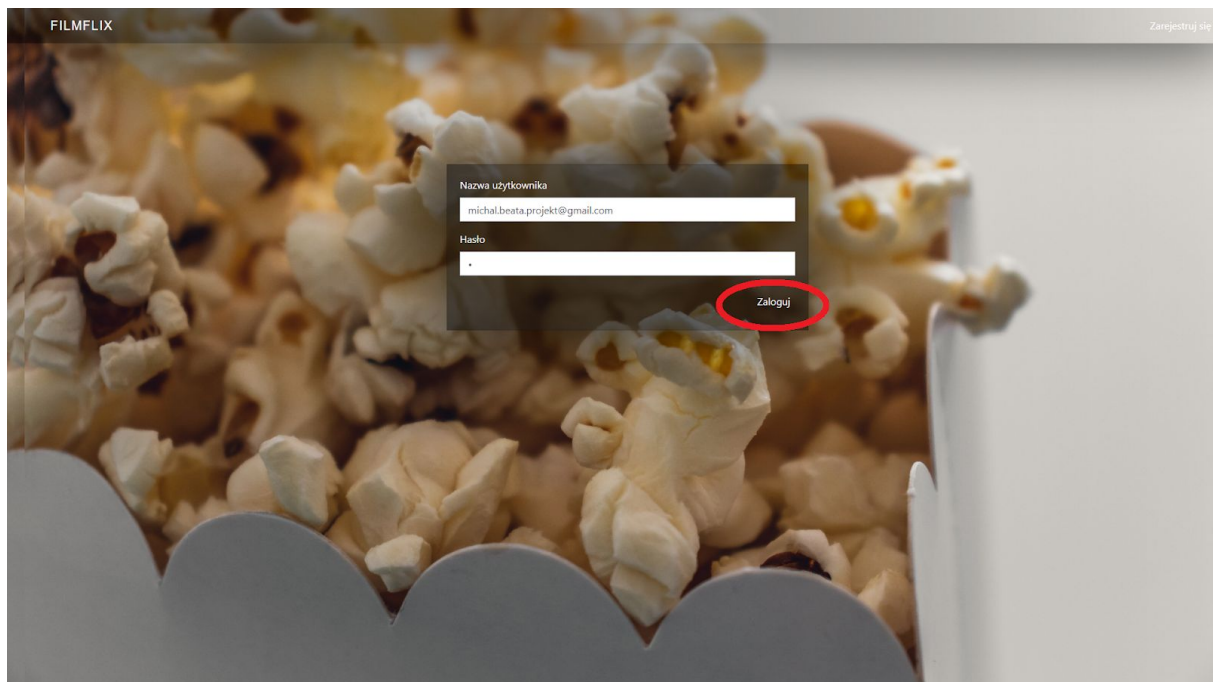
Na stronie z rejestracją należy podać swój A) adres e-mail oraz B) podać hasło, którymi to chcemy logować się do serwisu:



Podane dane zatwierdzamy przyciskiem “Enter” lub przyciskiem “Zarejestruj” znajdującym się w prawym dolnym rogu okna z podanymi danymi. Jeśli dane będą nieprawidłowe, to znaczy jeśli adres e-mail nie zawiera znaku “@” lub hasło nie zgadza się z hasłem, z pola “Potwierdź hasło”, lub podany adres e-mail jest już zajęty to system powinien zwrócić komunikat oznaczony na czerwono “Nieprawidłowe dane”, jeśli jednak wszystko uda się poprawnie wprowadzić to użytkownik zostanie przekierowany na kolejną stronę rejestracji - pola płatności. Etap wprowadzania metod płatności można pominąć i od razu przejść do logowania się, ponieważ na potrzeby prezentacji status “zapłacono” jest przechowywany na serwerze (w bazie danych), ale brak opłacenia konta nie blokuje możliwości oglądania filmów czy logowania się.

## 2. Logowanie

Aby zalogować się na swoje konto użytkownik musi przejść na stronę główną (/login) i w okienku na środku ekranu podać swoje dane z rejestracji:



i musi potwierdzić je enterem lub oznaczonym wyżej przyciskiem “Zaloguj”. Po zalogowaniu powinien wyświetlić się komunikat obok przycisku - “Zalogowano pomyślnie” jako zielony napis, a użytkownik powinien zostać przekierowany na stronę główną /home.

## 3. Strona główna /home

Po zalogowaniu, na górze strony widnieją dwie zakładki: Filmy i Seriale. Na potrzeby prezentacji przygotowaliśmy tylko filmy, ponieważ zakładka “Seriale” byłaby po prostu jej kopią zawierającą inne plakaty i odnośniki do plików na serwerze. W prawym górnym rogu zaś znajduje się przycisk “Profil”, a gdy najedziemy na niego myszą powinny wyświetlić się “Konto” oraz “Wyloguj się” z czego przycisk konto i profil przechodzą na stronę do zarządzania swoim profilem czyli metodami płatności oraz zmianą hasła. Można tam również znaleźć swój adres e-mail i status płatności. Przycisk “Wyloguj się” wylogowuje użytkownika i wraca na stronę logowania.



Poniżej nagłówka znajduje się tytuł i opis serwisu, oraz nieaktywny przycisk "Więcej informacji".

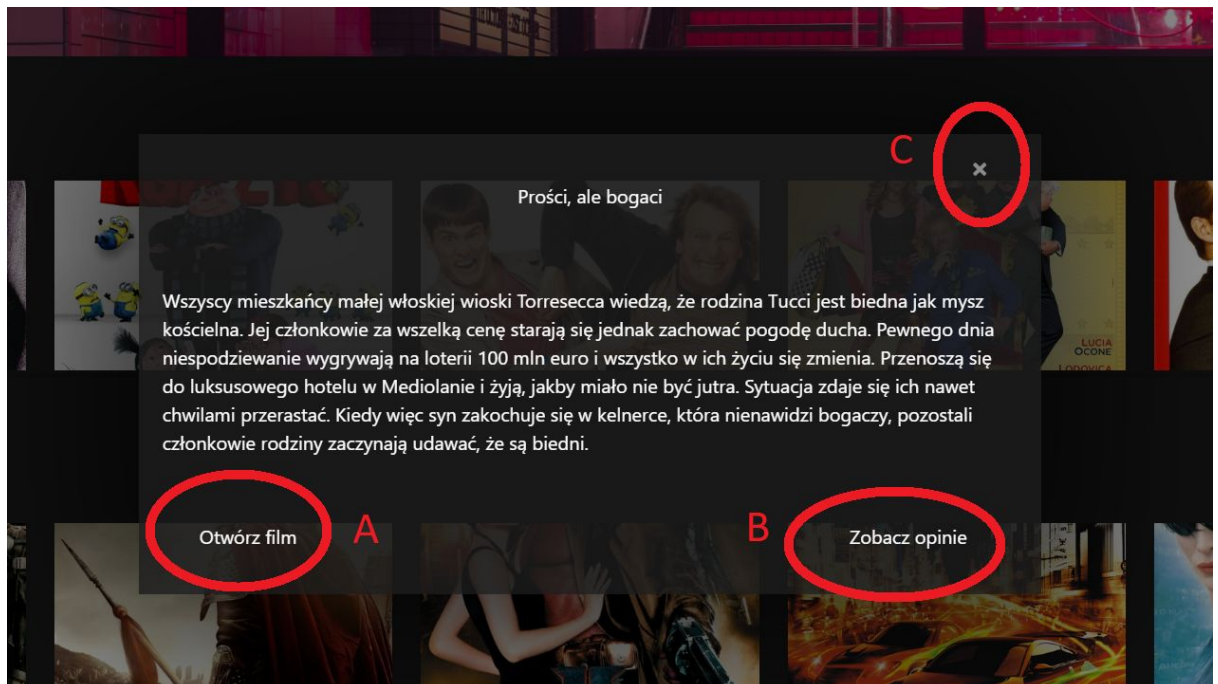
Zjeżdżając jeszcze niżej widnieją już slidery podpisane kategoriami filmy. Slidery takie możemy przesuwac na komputerze myszą klikając jej lewy przycisk i przesuwając w lewo lub w prawo, albo za pomocą białych strzałek po lewej i prawej stronie ekranu:



Na telefonie można przesuwac palcem.

#### 4. Oglądanie filmów i wystawianie opinii

Aby obejrzeć film należy kliknąć jego plakat, który powinien powiększyć się po najechaniu myszą. Na okienku, na środku ekranu widnieje tytuł i opis filmu oraz trzy przyciski w jego rogach:



A) Otwiera film - przechodzi do playera,

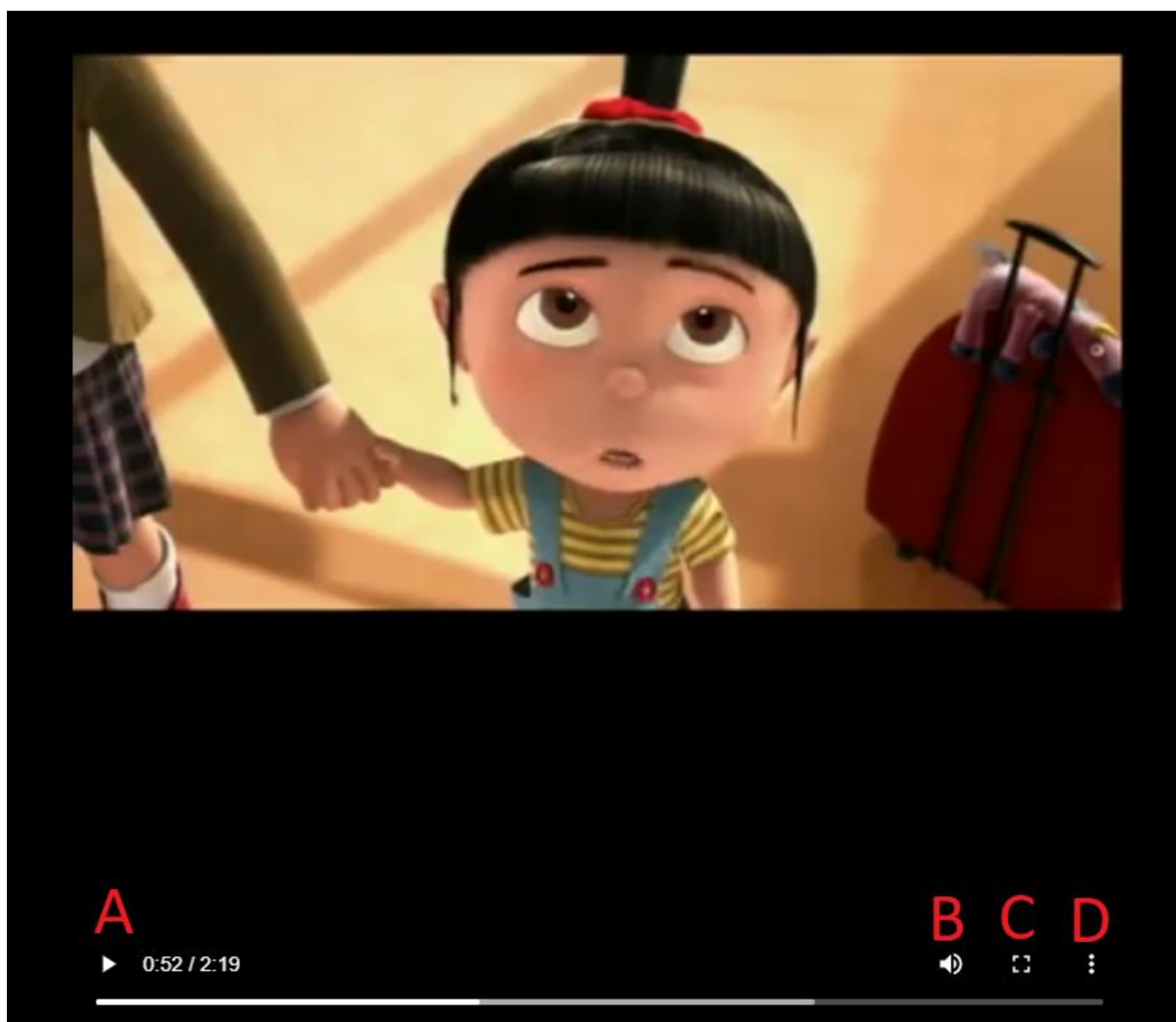
B) Pozwala zobaczyć lub wystawić opinię do filmu,

C) Wychodzi z okna, co również można uzyskać klawiszem "Esc" na klawiaturze.

#### 5. Oglądanie filmów

Aby obejrzeć film można posłużyć się punktem 4. Ogólnie każdy film jest przypisany do adresu /play/[tytuł\_filmu]. Odtwarzacz może różnić się w zależności od przeglądarki, ale powinien zawierać podstawowe funkcje jak:





- A) Odtwórz/zatrzymaj,
- B) Głośność,
- C) Pełny ekran,
- D) Ustawienia (w chrome jest w nich “obraz w obrazie” i “pobierz”).

Dodatkowo odtwarzacz automatycznie zawiera autoplay, czyli po uruchomieniu strony sam uruchamia film oraz w zależności czy obsługuje to przeglądarka powinno dać się sterować filmem za pomocą klawiatury np. przewijać strzałkami lewo i prawo, a góra i dół podgłaszać i przyciszać.