Państwowa Wyższa Szkoła Techniczno-Ekonomiczna

**im. ks. Bronisława Markiewicza**

**w Jarosławiu**

# Instytut Inżynierii Technicznej

# 



**Marcin Piasek, Dawid Piątek, Dawid Jabłoński**

**nr albumu: 37224, 37225, 37200**

**Dokumentacja Techniczna**

**Projekt Zespołowy**

**Tytuł projektu:**

**Aplikacja internetowa poświęcona filmom**

**Data: 03.05.2022**

Jarosław 2022

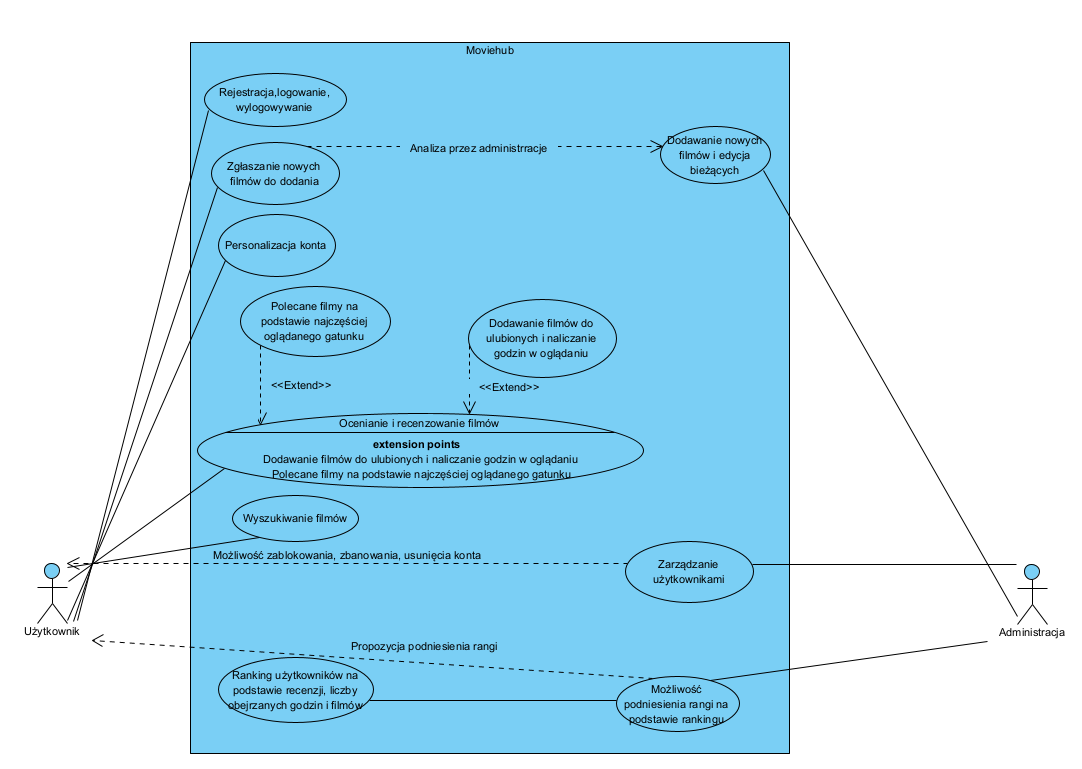
Wymagania funkcjonalne:

* Użytkownik może się zarejestrować, zalogować i wylogować
* Strona posiada panel administracji
* Profil użytkownika z możliwością personalizacji konta
* wyświetlanie na profilu użytkownika ilości obejrzanych filmów, ilości porzuconych i ilość czasu spędzonego na oglądaniu
* Użytkownik może ocenić film i dodać go do obejrzanych
* Filmy posiadają swoje podstrony z informacjami (czas trwania, aktorzy, gatunek, komentarze, ile osób obejrzało, ocena, zwiastun filmu w serwisie YouTube)
* Możliwość skomentowania filmu, podstawowa filtracja wulgaryzmów w komentarzach i możliwość ich usuwania
* Możliwość wyszukiwania filmów
* Ranking filmów i sortowanie po popularności, ocenie, gatunku
* Ranking użytkowników (sortowanie po ilości komentarzy, godzin, recenzji)
* Możliwość dodawania nowych filmów przez administrację
* Zgłaszanie przez użytkowników filmów do dodania
* Rangi użytkowników (Admin, Recenzent, Użytkownik)
* Możliwość podniesienia rangi na podstawie rankingu
* Możliwość zaznaczenia ponownego obejrzenia filmu, naliczenie dodatkowego czasu (rewatch)
* Polecane filmy na podstawie najczęściej oglądanego gatunku

Wymagania niefunkcjonalne:

* Strona ma być minimalistyczna
* przejrzysta
* Intuicyjna w obsłudze

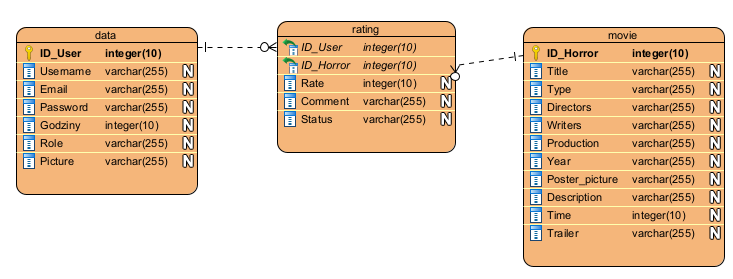
Diagram przypadków użycia:



poglądowy wygląd strony:



Diagram ERD bazy danych



Projekt napisany w środowisku:

Microsoft Visual Studio Code 1.66.2

Aplikacja korzysta z XAMPP w wersji:

**v3.2.4**

Do obsługi bazy danych potrzebny jest program **XAMPP** z uruchomionymi serwerami **Apache** i **MySQL**,aby móc zaimportować bazę danych należy ją utworzyć w **localhost/phpmyadmin**

kwerenda tworząca bazę danych:

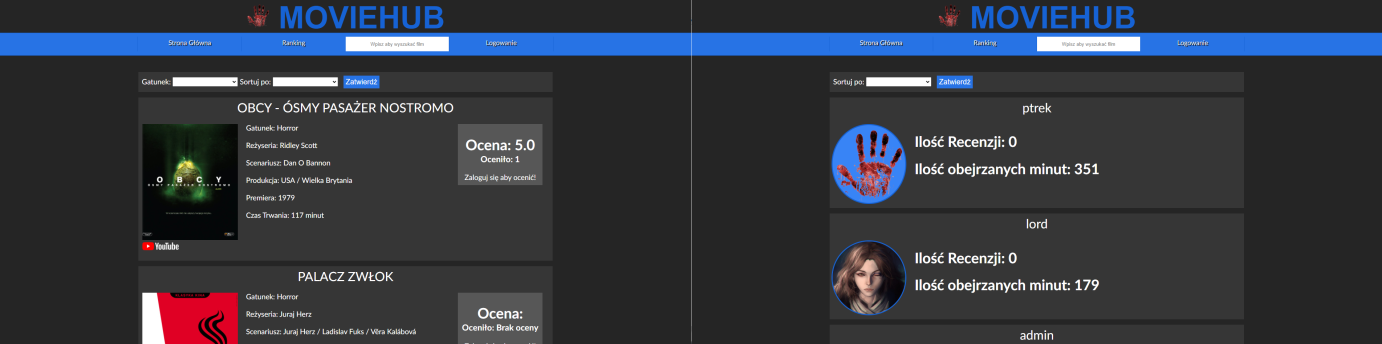
**CREATE DATABASE `moviehub` /\*!40100 DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4\_polish\_ci \*/**

Po utworzeniu bazy należy ją zaimportować, plik bazy znajduje się w projekcie pod nazwą: **moviehub.sql**

Strona internetowa po uruchomieniu:

****

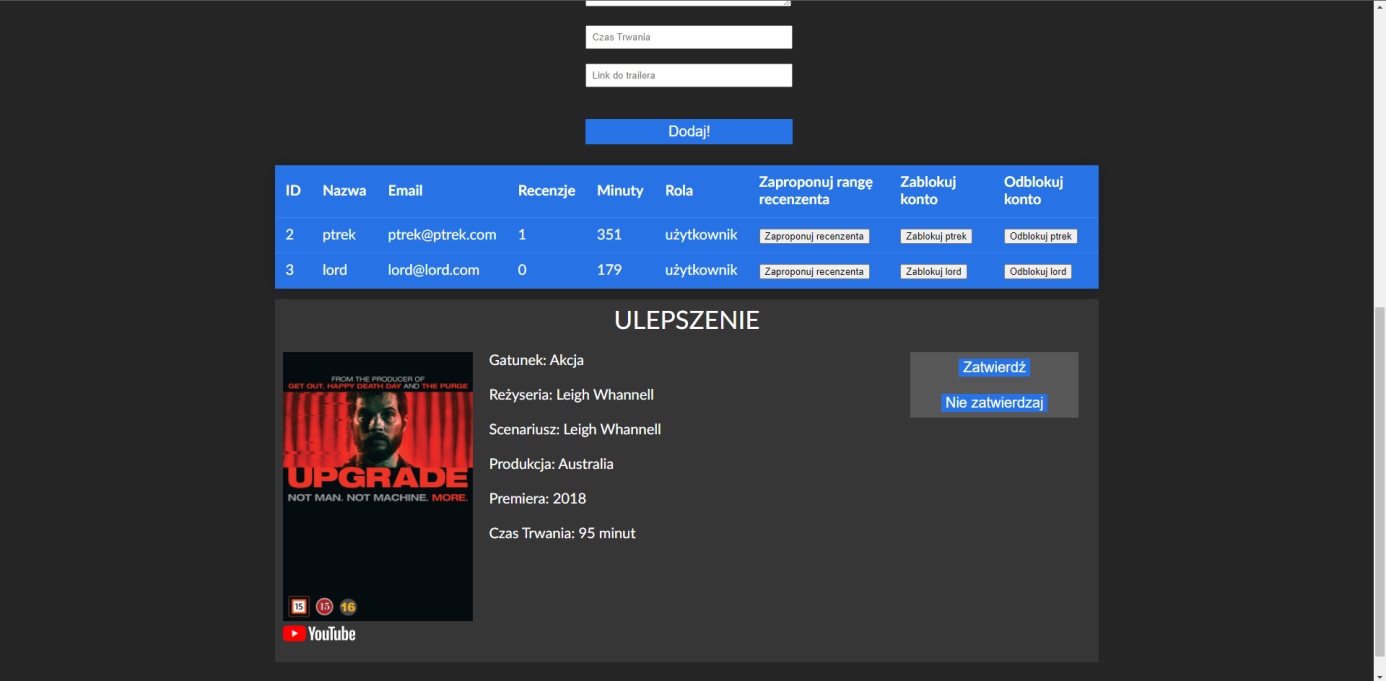
Na stronie głównej wyświetla się losowy film z bazy. Na każdej podstronie znajduję się pole wyszukiwania, w którym można wyszukać film na podstawie tytułu. Filmy można oceniać dopiero po zalogowaniu się.



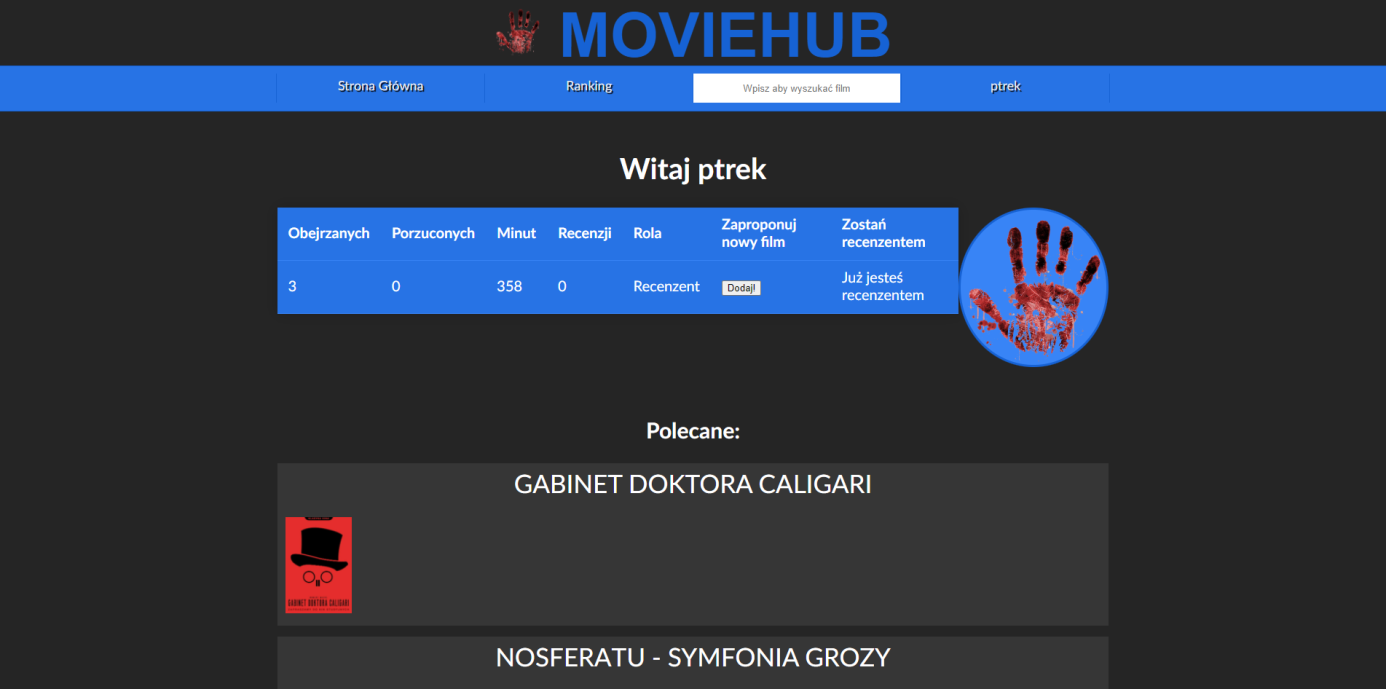
po przejściu na podstronę rankingu filmów jest możliwość ich filtrowania na podstawie gatunku i sortowania ich Alfabetycznie, po ocenie, po popularności. W rankingu użytkowników również można sortować użytkowników tym razem po liczbie minut spędzonych na oglądaniu lub ilości wystawionych recenzji.



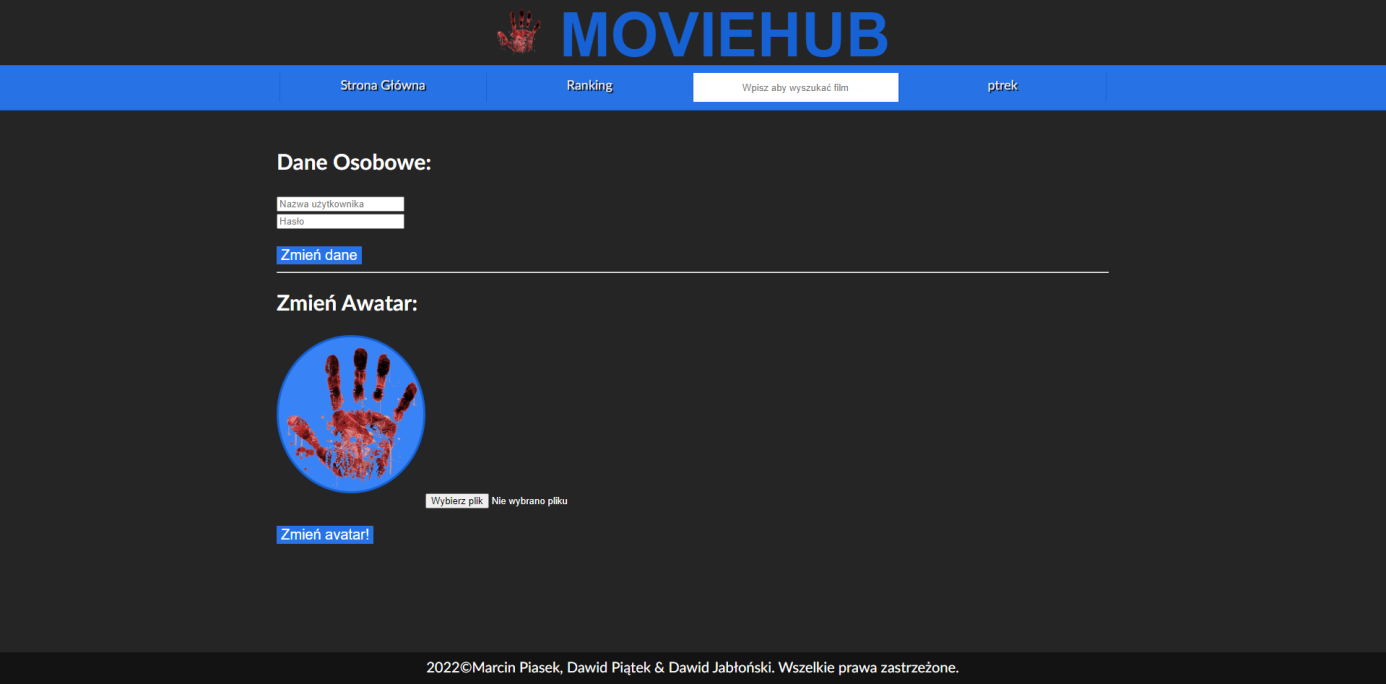
Po przejściu na podstronę filmu można przeczytać, krótki opis filmu oraz jeśli jest się zalogowanym wystawić recenzję, która posiada filtrację słów i nie pozwoli wystawić recenzji, w której znajdzie się słowo z listy zabronionych. administrator może tutaj usuwać recenzje.



Po zalogowaniu na administratora zostajemy przeniesieni do panelu administracji, gdzie można dodawać nowe filmy do bazy danych, poniżej są wyświetlani użytkownicy, którzy mogą zostać zbanowani, odbanowani lub można również wysłać do nich propozycje zmiany ich roli na recenzenta, aby mogli również dodawać filmy bez czekania na akceptację ich przez administracje. Poniżej wyświetlają się filmy które zostały zaproponowane przez użytkowników, administrator może je akceptować lub usunąć.



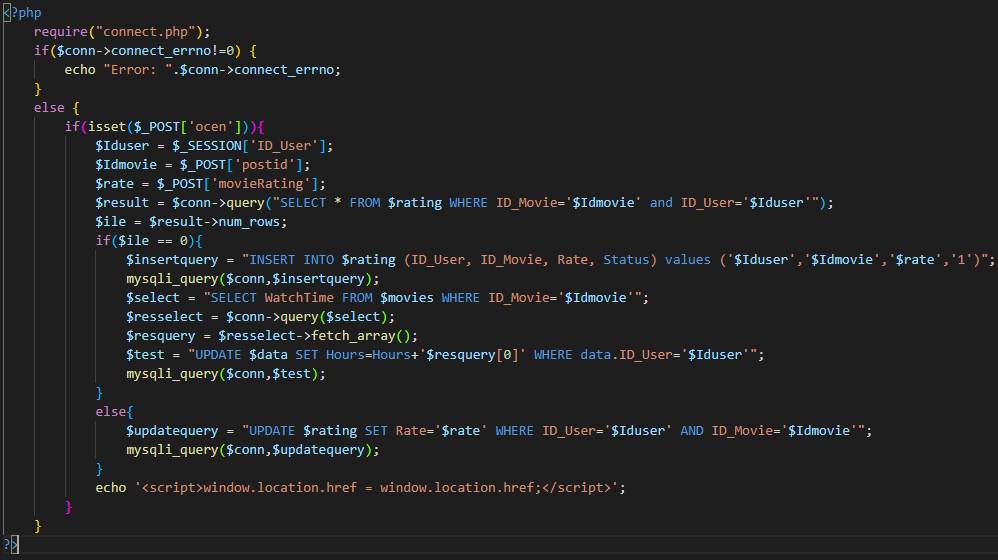
Po zalogowaniu na użytkownika wyświetlane są jego statystyki takie jak ilość obejrzanych filmów ilość porzuconych filmów ilość minut spędzonych na oglądaniu i ilość wystawionych recenzji. Wyświetla się również aktualna rola użytkownika. Obok statystyk wyświetlany jest przycisk do zaproponowania filmu do dodania, aby ten film był wyświetlany np. w rankingu musi on zostać pierwsze zaakceptowany przez administratora. Jeżeli użytkownik został wybrany do otrzymania wyższej rangi wyświetli się tu przycisk do akceptacji. Poniżej wyświetlane są filmy które użytkownik obejrzał i oceny jakie im wystawił. Tutaj też użytkownik może zmienić ilość ile razy dany film obejrzał lub czy dany film porzucił w trakcie jego oglądania.

****

Gdy użytkownik jest zalogowany można również przejść do ustawień gdzie można zmienić wyświetlany awatar lub zmienić login i hasło.

Przykładowy kod PHP

Ocenianie filmów:

****

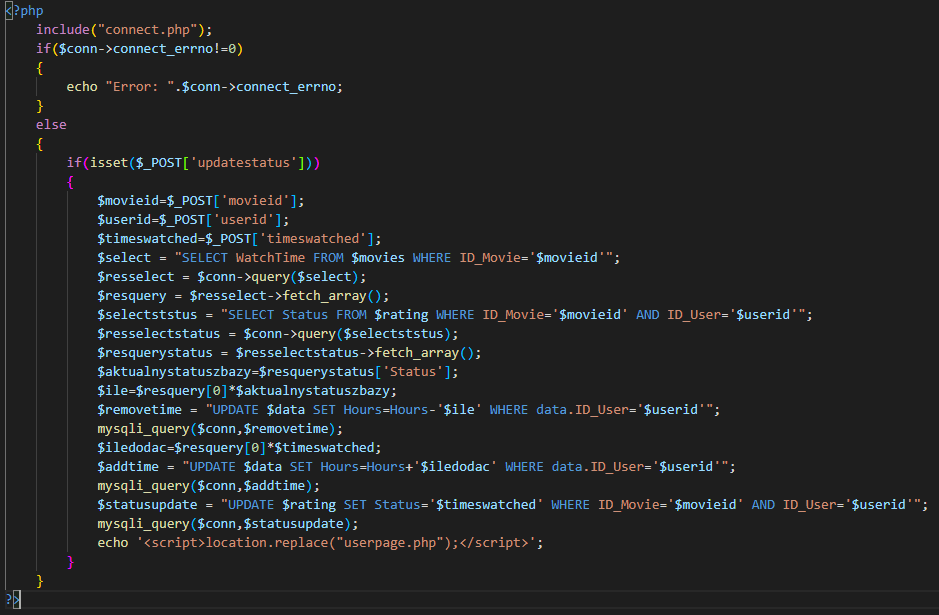
Na początku dołączany jest skrypt *connect.php*, który tworzy połączenie z bazą danych, następnie sprawdzane jest, czy przy łączeniu z bazą danych nie wystąpił błąd, jeżeli wystąpił, zostanie on wypisany. Gdy nie ma błędu, sprawdzane jest, czy zmienna *ocen* przesłana metodą *POST* jest ustawiona, jeżeli tak przypisywana jest do zmiennej *Iduser* zmienna sesyjna *ID\_User* odpowiadająca użytkownikowi, który jest aktualnie zalogowany. Przypisywane do zmiennych *Idmovie* i *rate* są również przesłane za pomocą *POST* *postid* i *movieRating*. Następnie do zmiennej *result* przekazywany jest wynik zapytania SQL umieszczonego w argumencie metody *query()* wywołanej na rzecz obiektu *conn*. W kolejnym kroku dla obiektu *result* wywołujemy metodę *num\_rows*, aby zliczyć ilość wierszy otrzymanych w wyniku i przypisujemy to do zmiennej *ile*. Sprawdzamy, czy zmienna *ile* jest równa 0, jeżeli tak, to przypisujemy do zmiennej *insertquery* zapytanie SQL dodające wpis do bazy, w kolejnej linii wywoływana jest funkcja *mysqli\_query* z argumentami: *conn* odpowiadającym za połączenie z bazą i *insertquery* za zapytanie *SQL*. W kolejnym etapie tworzone jest nowe zapytanie przypisane do zmiennej *select*. Następnie do zmiennej *resselect* przekazywany jest wynik metody *query()* wywołanej z argumentem *select* na rzecz obiektu *conn*. Dla *resselect* wywoływana jest metoda *fetch\_array()* zwracająca tablicę numeryczną, którą przypisujemy do zmiennej *resquery*. Następnie z wykorzystaniem pierwszej wartości z *resquery* tworzone jest zapytanie *SQL* aktualizujące liczbę obejrzanych godzin przez użytkownika, w kolejnej linii wywoływana jest funkcja *mysqli\_query* z argumentami. W przeciwnym wypadku, jeżeli użytkownik już raz ocenił ten film i zmienia swoją ocenę, tworzone jest zapytanie aktualizujące ocenę i wywoływana jest funkcja *mysqli\_query* z odpowiednimi argumentami. Na koniec poleceniem echo wypisywany jest skrypt *js*, który ma za zadanie odświeżyć stronę.

Dodawanie Recenzji:



Po pierwsze rozpoczynamy sesję, dołączamy plik connect.php i sprawdzamy czy połączenie z bazą zostało ustanowione. następnie sprawdzamy czy but\_submit jest ustawiony jeżeli tak dane przesłane przez POST przypisywane są do zmiennych. zmienna z recenzja przypisywana jest do nowej zmiennej, dzielona na poszczególne słowa za pomocą implode(). Następnie tworzona jest tablica z wykluczonymi słowami, w pętli foreach() dla każdego słowa z listy sprawdzane jest czy znajduje sie w recenzji i jeżeli się znajduje zostaje ustawiona zmienna sesyjna z błędem zmienna $working zmienia wartość na false a strona zostaje odświeżona. Jeżeli nie zostanie znalezione żadne wystąpienie słowa z listy, zostanie wywołane zapytanie SQL wyciągające recenzje aktualnie zalogowanego użytkownika dla filmu z danym id, a następnie zostanie policzona ilość zwróconych rekordów. W kolejnym warunku sprawdzane jest czy $working dalej jest ustawiony na true, a ilość komentarzy jest równa 0 jeżeli tak to do bazy wstawiana jest racenzja z id użytkownika i id filmu. Ustawiana jest również zmienna sesyjna z komunikatem potwierdzającym dodanie recenzji i strona zostaje przeładowana. Kolejna instrukcja warunkowa sprawdza czy zmienna working jest ustawiona na true jeżeli tak następuje aktualizacja recenzji

Zmiana ile razy użytkownik obejrzał film:



Na początku dołączany jest skrypt *connect.php* do połączenia z bazą, sprawdzane jest w instrukcji warunkowej czy połączenie przebiegło pomyślnie i jeżeli tak to dodatkowo sprawdzane jest czy została ustawiona zmienna updatestatus. Jeżeli tak to do zmiennych przypisywane są wartości przesłane metodą POST, wykonywane jest zapytanie do bazy pobierające czas trwania dla filmu o danym id i zapytanie pobierające status dla tego filmu i użytkownika o tym id z tabeli rating. Następnie aktualny status przemnażany jest przez czas filmu i zapisywany do zmiennej ile, wykonywana jest aktualizacja ilości obejrzanego czasu przez danego użytkownika, odejmując aktualną ilość czasu spędzonego na oglądaniu o wartość zmiennej ile. Następnie ilość czasu filmu przemnażana jest przez otrzymaną z POST wartość timeswatched i wynik tego zapisywany jest do zmiennej iledodac. Wykonane zostaje zapytanie SQL aktualizujące ilość czasu obejrzanego na profilu użytkownika dodając wartość zmiennej iledodac. Na koniec aktualizowany jest status w tabeli rating na wartość zmiennej timeswatched dla filmu o danym id i id użytkownika i następuj odświeżenie strony