Coursera
The Hong Kong University of Science and Technology

Full Stack Web Development Specialization

Capstone Project

MyMovies

Author:
Tiago Justino

Supervisor:
Jogesh Muppala

May 15, 2017

1 Introduction

MyMovies is a social network for cinema fans. It keeps track of movies you've watched, want to watch and don't want to watch. By leveraging the IMDB database it allows you to add custom pieces of information, such as, rating and comments. Also, the social network aspect brings you the possibility of discovering new movies you want to watch and new points of view on movies you've watched at the same time that it allows you to share good moments with people you love.

As a bonus, by keeping track of people's movie watching habits MyMovies is able to help the cinema industry on targeting their customer more precisely.

1.1 Expected List of Features

Inside the scope of this project we plan on achieving the following features:

- The user will be able to create an account and login to the system with or without Facebook. The user will also be able to link or unlink their MyMovies account to their Facebook account;
- The user will be able to invite other users to be their friends on MyMovie. The invited user will be able to accept or decline the invite;
- The user will be able to invite their Facebook friends to join MyMovies.
- The user will be able to follow other users. Users added as friends are followed by default. The user will also be able unfollow friend without unfriending them;
- The user will be able to mark a movie as watching, watched, want to watch or don't want to watch;
- For movies previously marked as watched, the user will be able to favorite and to add rating and comments;
- The user will be able to *fan* names, such as actors, actresses, directors and producers;
- Movies marked as watching are automatically marked as watched after the movie duration (known from imdb api);
- When marking a movie as watching or watched the user will be able to mark a friend as ”watching with”;

- The user will be able to add a date (day, month or year) for movies watched in the past. Date is not mandatory;
- The user will be able to mark a movie as watched more than once;
- The user will be able to access a timeline page where they can see the activities of whom they follow. The timeline will show activities such as watching and watched movies, and comments;
- The user will be able to like or comment on friend's activities,
- The user will be able to mark their comment (or part of it) as spoiler,
- Comments marked as spoiler will appear with a warning message on a user timeline if they haven't watched that movie,
- The user will be able to create and join groups (private and public), such as "Movie Club at Work" or "Stanley Kubrick Fans",
- The user will be able to recommend a movie to a friend.

2 Design and Implementation

Initially only three of the planned features were actually implemented: Login, search for movie title, and add and remove to/from watched list. The three screens can be seen at Figures 1, 2 and 3;

The application scaffolding was done using a github tutorial [2] which serves both the front-end and back-end together.

2.1 Front-end

The front-end was implemented using AngularJS version 1.4.5 together with Bootstrap 3.3.5 and Angular-Route. Also, a client service for accessing the Omdb API was implemented here (see Section 2.3 for more details).

The routes are:

home: Shows the search page (Figure 2) if the user is logged in. If the user is not logged in, it redirects to the login/signup page (Figure 1).

login: The login/signup page.

profile: The profile page. Shows the watched movies list.

2.2 Back-end

The back-end was implemented using Express 4.13, Mongoose 4.3 and Passport 0.3.

The exposed endpoints are:

login (post): Checks the user credentials and authenticate them.

logout (post): Destroys the user session.

loggedin (get): Returns the user info if they're loggedin.

signup (post): Registers a new user.

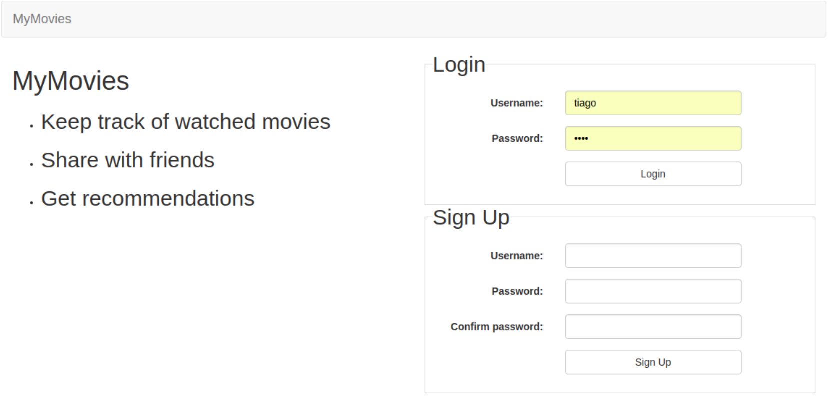
watched (post): Adds a movie to the watched list.

watched/:id (delete): Removes a movie from the watched list.

2.3 Omdb API

The initial idea was using Imdb to retrieve the movies information. Unfortunately, Imdb doesn't provide any official API. There is, however, an unofficial API called Omdb (The Open Movie Database, [1]). This API provides not just information from IMDB, but from other movie websites such as rottentomatoes.com.

Here we opted out for keeping the API access only on the client side, the reason being bandwidth saving.



MyMovies

MyMovies

- Keep track of watched movies
- Share with friends
- Get recommendations

Login

Username: tiago

Password: ••••

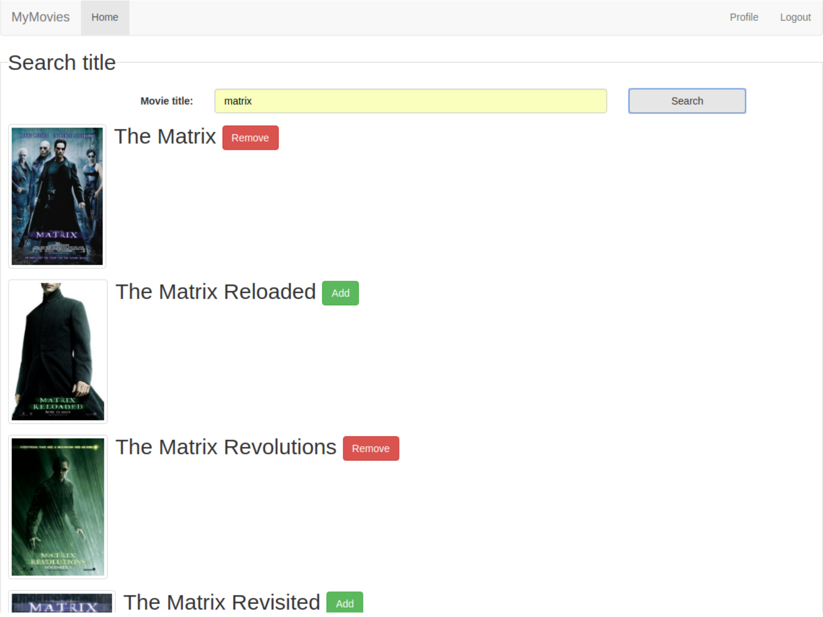
Sign Up

Username:

Password:

Confirm password:

Figure 1: Sign up / Login page.



Search title

Movie title: matrix

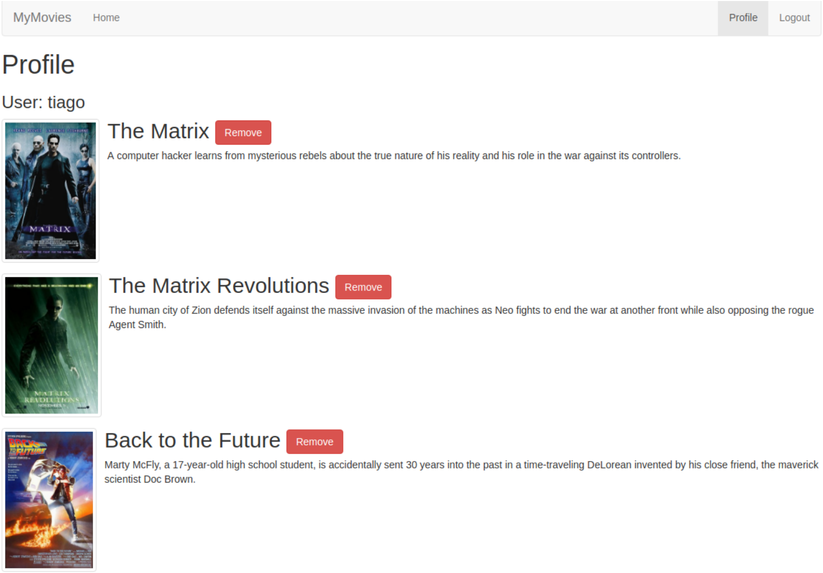
The Matrix

The Matrix Reloaded

The Matrix Revolutions

The Matrix Revisited

Figure 2: Search for titles.



Profile

User: tiago

The Matrix

A computer hacker learns from mysterious rebels about the true nature of his reality and his role in the war against its controllers.

The Matrix Revolutions

The human city of Zion defends itself against the massive invasion of the machines as Neo fights to end the war at another front while also opposing the rogue Agent Smith.

Back to the Future

Marty McFly, a 17-year-old high school student, is accidentally sent 30 years into the past in a time-traveling DeLorean invented by his close friend, the maverick scientist Doc Brown.

Figure 3: Profile page.

3 Conclusion

The results were much less than expected due mainly to two reasons: 1. I believe I planed too many features to be done in two weeks. and 2. I underestimated the difficulty of using the Imdb unofficial API.

References

[1] The open movie database. http://www.omdbapi.com/. Accessed: 2017-05-14.

[2] Passport authentication with express and angular. https://github.com/nax3t/angular-express-passport-tutorial. Accessed: 2017-05-14.