

Introduction

Movies is a movie website manager. Here the user can discover new movies through lists as “Popular”, “Upcoming”, “Top rated” or by search. The user can discover more information about the movie by clicking on the movie poster and add the movie to their watchlist. To access the “add to watchlist” and other functions the user must be logged in. Once logged in the user can access their profile, which contains information about their watchlist, search history and stats/information. The user has the option of removing a movie from their watchlist. The user does not have to be logged in to use the search function and filter/sort.

The website utilizes The MovieDatabase API as

Homepage

This page presents the user with top movies, displaying movies in a grid. It has the option of selecting movies based on lists “Popular”, “Upcoming” and “Top rated”, available in the navigation bar to the left. If the user clicks on a movie a modal appears containing more information and actions.

Search

The search bar lets the user search for any movie they want. After searching, one can filter and sort the result. If the user clicks on a movie a modal appears containing more information and actions.

Userpage

The user page offers information about the user, such as username, email, number of searches and movies in watchlist. Below this information there are two tabs “Watchlist” and “Search history”. The user can navigate between these tabs/components, thus displaying different content.

Watchlist

The watchlist is a list containing all the movies the user has added to their watchlist. It is meant as a collection of movies the user wants to watch in the future. It also has the option of removing movies from their list.

Search history

The search history is a list containing all the searches the user has done. Each search has a timestamp (the date of the search) and the search text itself.

Project requirments

The web-application

The project runs on the groups virtual machine (<http://it2810-20.idi.ntnu.no:8084>) and uses node.js on the server-side. The project was started developing the web-application in Angular version 4, but it was updated to version 5. Bootstrap is also used for UI. It is setup using a MEAN Stack (MongoDB, Express.js, Angular, Nodejs).

Backend database

We have chosen to use MongoDB with Mongoose for our database. The website utilizes The Movie Database API and it is used for fetching data. The application uses many reading and writing operations to the database.

List/grid based view

Our list/grid based view is implemented both the homepage and the search page where the user can see the movies they have searched/selected. The user sees the result as movie-posters but has the option to see a more detailed view when clicking on the movie.

Sorting

The list based view can also be sorted on “top rated” and “most popular”. By clicking one of the buttons the user can choose between these sorting methods.

Filtering

The search result can also be filtered on “year” and “genre”. By clicking one of the filters the user is granted with the filtered result in the grid view.

Dynamic data loading

The movies in the grid view is loaded dynamically using lazyload. The group utilizes a npm module called “infinite-scroll”. The page loads more movies if the user scrolls down.

User page functionality

The user page is the users own profile page for when they are logged in. A user can click on the user icon in the navigation bar and be redirected to this page.

Session handling

The application uses JSON Web Tokens as session handling. For more information see: <https://jwt.io/>

Alternative view

Our fancy view is a displayment of user stats. This include the number of searches the user has and number of movies in the watchlist. Also the displayment of movies is kind of fancy :)

Testing

For testing the server code we are using Mocha, while the angular code test are using Karma with Jasmine.