COMP2011

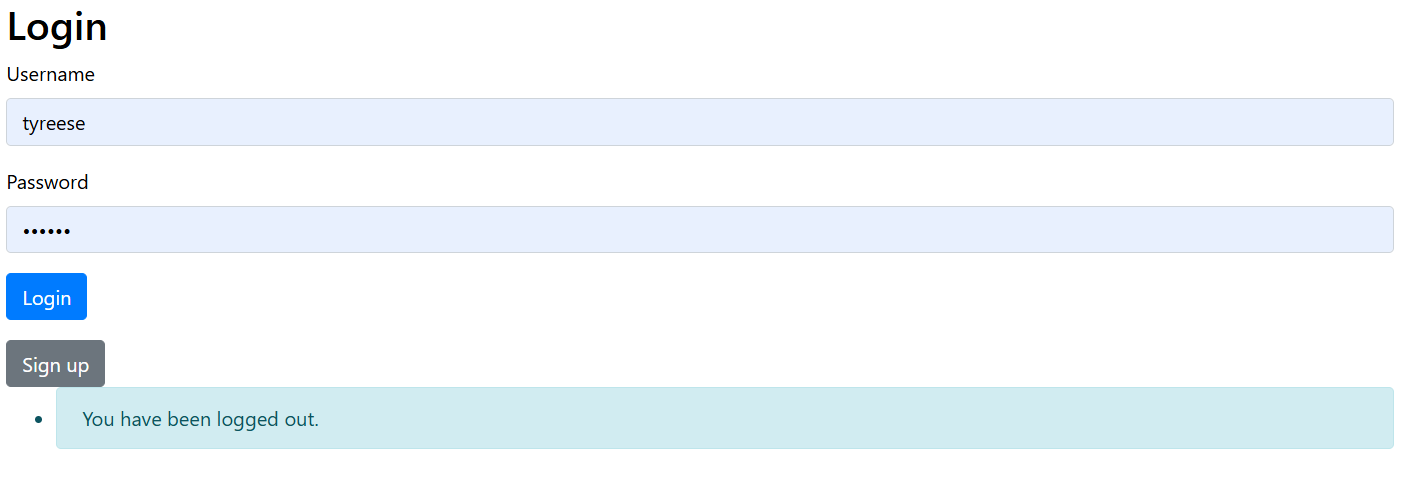
Tyreese Green

# My Website

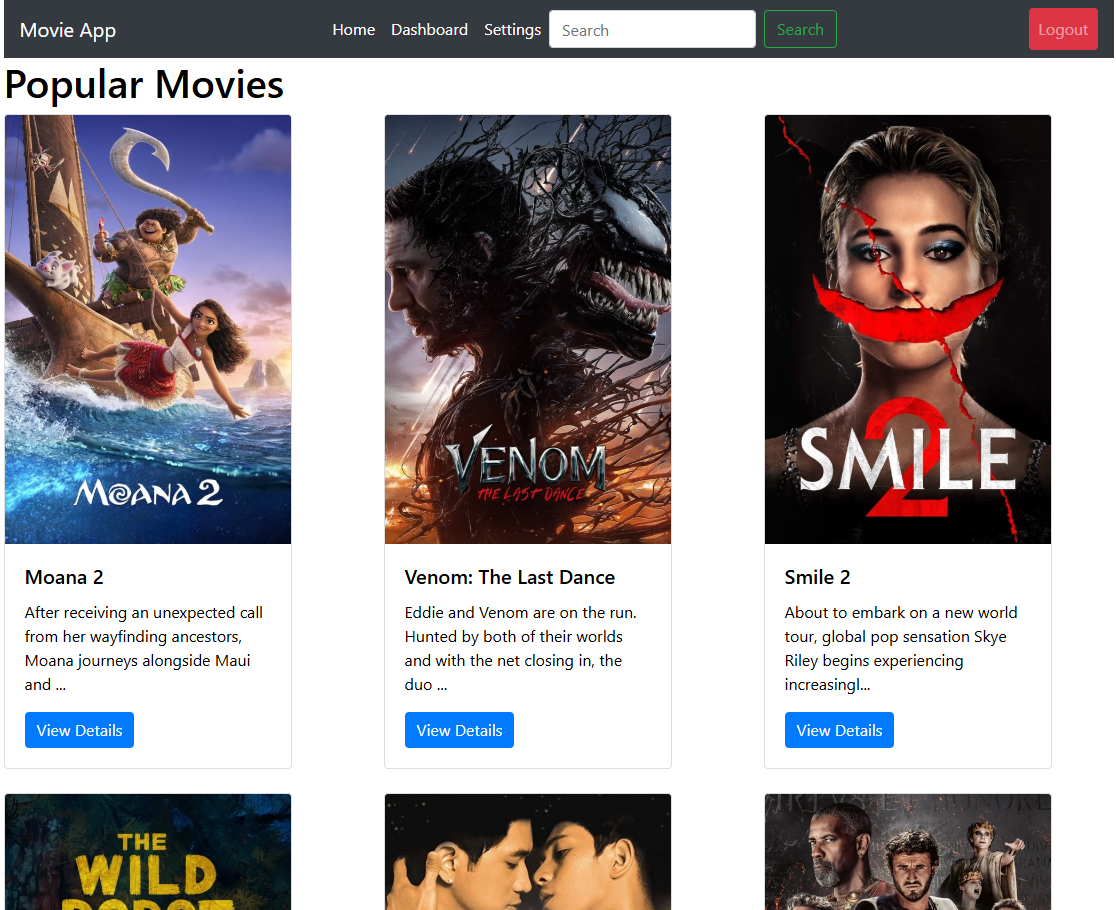
This website is a movie reviewing app. The user can log in to their account, where they can view, like and review movies. Users can also view the reviews and ratings of other people when on the movie page. On the dashboard page, the user can see their liked movies.

## User Path

When opening the site, the user will be met with a login screen. Here they can login or create a new account. This page also displays relevant error messages or information such as entering an incorrect password.



Once the user has logged in, they will be sent to the homepage, where they can see a list of popular movies. Each movie card can be pressed, which takes them to a page which has more details on the movie and the user can view or create ratings for that movie.



Movie page:

A screenshot of a movie

Description automatically generated

When the user goes to their dashboard, they can view their liked movies. This is intended for users to use this as a watch list as well as movies they enjoy.

A screenshot of a computer

Description automatically generated

Also, the user has the option to search for specific movies that they want to view. Here is an example using Star Wars as the search query.

A screenshot of a movie

Description automatically generated

Finally, the settings page allows the user to change their username and password as well as a toggle dark mode button.

A screenshot of a computer

Description automatically generated

## Design

I chose this design because it displays the necessary information to the user and it is all easy to access. While developing this, I encountered many issues such as learning how to use the API to gather movie data. To get past this issue, I read the documentation on how to get the data using the API, then I used this for my solution as seen below.

A computer screen shot of text

Description automatically generated

For the movie cards on the homepage, I used the compact design so the users could see a small preview of the movie description as well as the poster. This should be enough information for the user to know about the movie so they can make a choice on whether they want to view more detail on the movie.

Furthermore, for the movie details page, I chose this design so that the rest of the movie detail was displayed and the review section was at the bottom of the page. It was necessary to put this at the bottom of the page as the amount of reviews can only increase overtime so it would be the best location for the reviews.

## Database

This database has multiple tables:

* User – stores user details such as name and password
* Movie – store movie details such as genre, release date and overview
* Review – Stores a user’s ratings and review, that links to the user and the movie they have reviewed
* Like – Creates a many-to-many relationship between users and movies. This allows users to like a movie

The design of this database adds to the user experience as it provides a system for users to like movies and review them. Users can use the like feature to create their own small feed of movies that they particularly enjoy. Furthermore, the review feature allows for users to interact with other users by leaving reviews and ratings.

The many-to-many relationship between users and movies is implemented through the like table in the database as seen below. A user can have many likes, and a movie can have many likes

A screen shot of a computer program

Description automatically generated

## Accessibility

Firstly, I implemented a dark mode feature so users can choose a more appealing contrast. Moreover, I also used the Wave web accessibility evaluation tool to ensure that I was meeting industry standards. In the screenshot below, it shows that no errors were detected, so I was compliant with standards. This screenshot also shows an example of the site with the dark mode feature enabled.

A screenshot of a computer

Description automatically generated

Also, when using a smaller screen, the navbar shrinks down so that it appears in a toggle navbar menu as seen below. This is good for devices with smaller screens

A screenshot of a movie

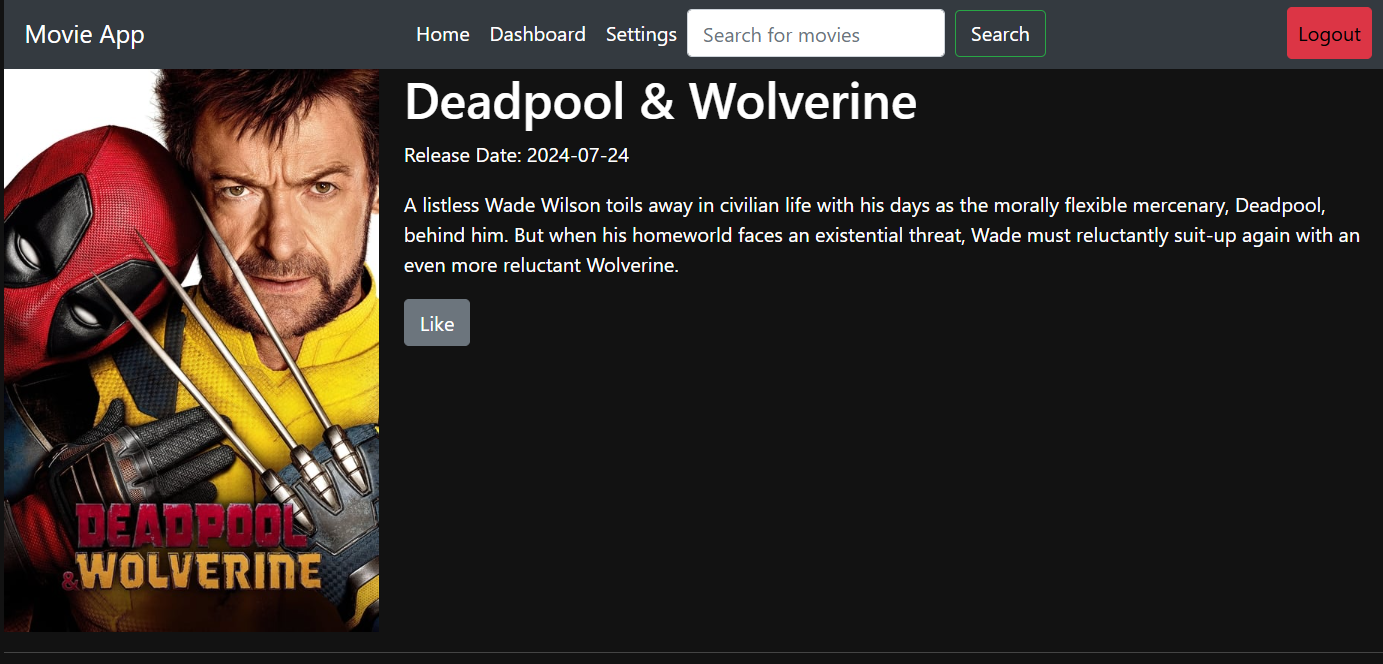
Description automatically generated

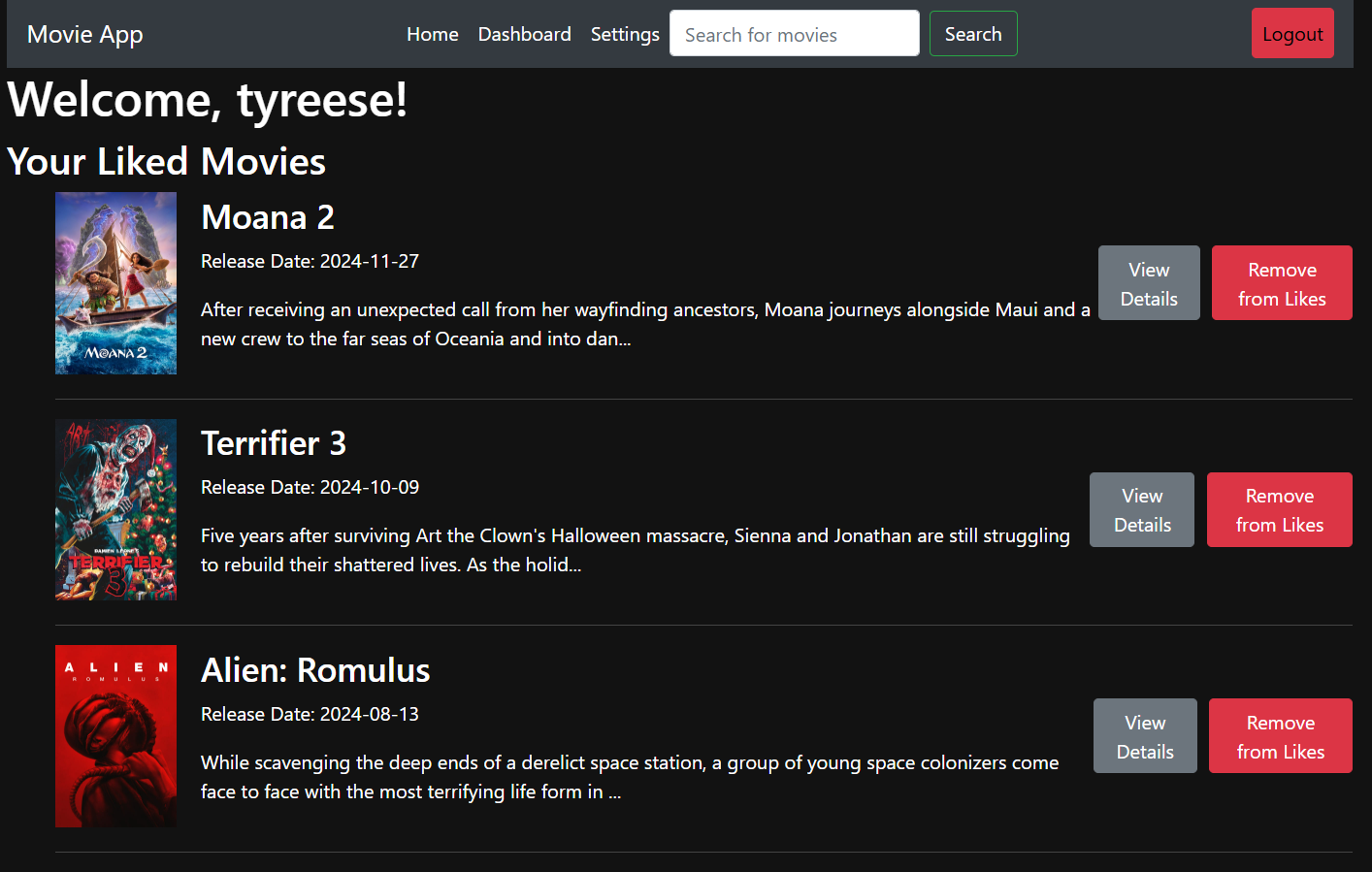
## Advanced Feature

My advanced features are the liking system and the reviewing system. I used ajax to implement these features.

### Liking System

Users have the option to like a movie when they are on the detailed page. This adds a movie to the users own personal list of movies.





A screen shot of a computer program

Description automatically generated

### Reviewing feature

The reviewing feature allows users to submit a rating and a review, which is visible to other users. This allows other users to know if the movie is good or bad as well as providing any other details about the movie.

A black and white rectangle with white lines

Description automatically generated

Code for submitting a review:

A screen shot of a computer program

Description automatically generated

Code for getting all of the review for a movie:

A screen shot of a computer program

Description automatically generated

# Testing & Critical Analysis

## Testing

### Functionality

Every time I added a new feature, I would thoroughly test that feature to ensure that it worked and to make sure that it did not effect any other features. An example of this is when I was implementing the search function, I realized that some unpopular movies would sometimes appear, so they would have missing data. Once I discovered this issue, I implemented a check to see if that data is there before adding attempting to add it to the database.

### Layout & responsiveness

I used chrome to test different device sizes on each page. The app would maintain a similar layout with a more layered structure. The navigation menu would also become a toggle menu, which is easier to use on mobile devices.

A screenshot of a computer

Description automatically generated

### Accessibility

To make sure that my site was accessible, I used the wave accessibility chrome extension on the site to see if there were any compliance issues. When using this, I noticed many issues involving contrasting colours on buttons. To address these issues, I changed the colour of some text and the colours to ensure that they met the contrasting colours requirements. Once I was finished addressing these issues, the pages that I checked were compliant with the accessibility requirements.

## Analysis

### What Went Well

Discuss the things which you think you did really well- what are you proud of in this work?

I believe that my implementation of the like and review system went well. I found that the system is easy to use and works well with the function of the website. Also, this allows users to interact with the website and my implementation of how reviews are displayed was also good. A screenshot of how reviews are displayed can be seen below.

A screenshot of a computer

Description automatically generated

Furthermore, another thing that went well was the layout of the page. Many key buttons are always easy to access and key parts of the page such as the movie card are clearly displayed to the user as each movie card shows the image, a description and an option to view more details.

### What Went Badly

What did you find particularly challenging, and why? What specific problems did you have?

When working with the API, I encountered multiple issues such as odd data formats. For example, when I implemented the search feature, some very old movies with limited data would appear, which did not have a release date. This caused the web page to crash multiple times while testing. I had to ensure that there were many error checking systems in place to avoid issues like these.

Also, initially implementing the login system was difficult as personal details such as passwords had to be securely stored. I had to do some research on how to implement the password system with some security. Eventually, I found how to implement hashing for the passwords, so the raw password would not be stored in the database.

### Improvements

Initially, I wanted this to be a movie recommendation app, however I did not have enough experience with the API to implement that within the time I completed the app. I would implement this. Also, I would give the option to view other user’ profiles so they can see their liked movies.

# Bibliography

# 

Website API Movie data taken from:

https://developer.themoviedb.org/docs/getting-started