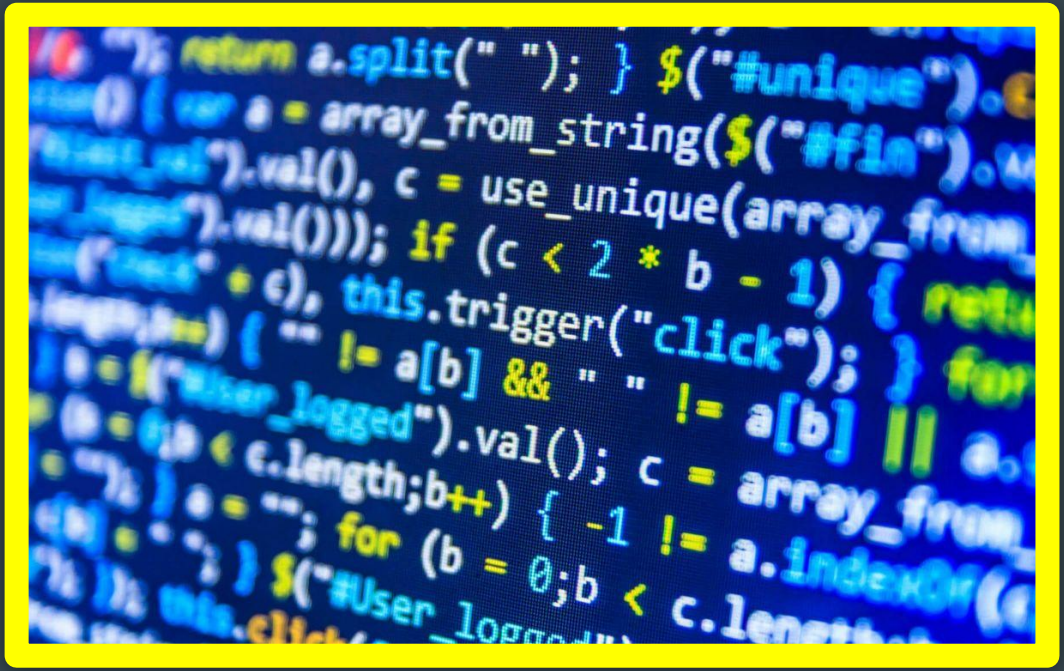
Full Stack Web Development Case Study

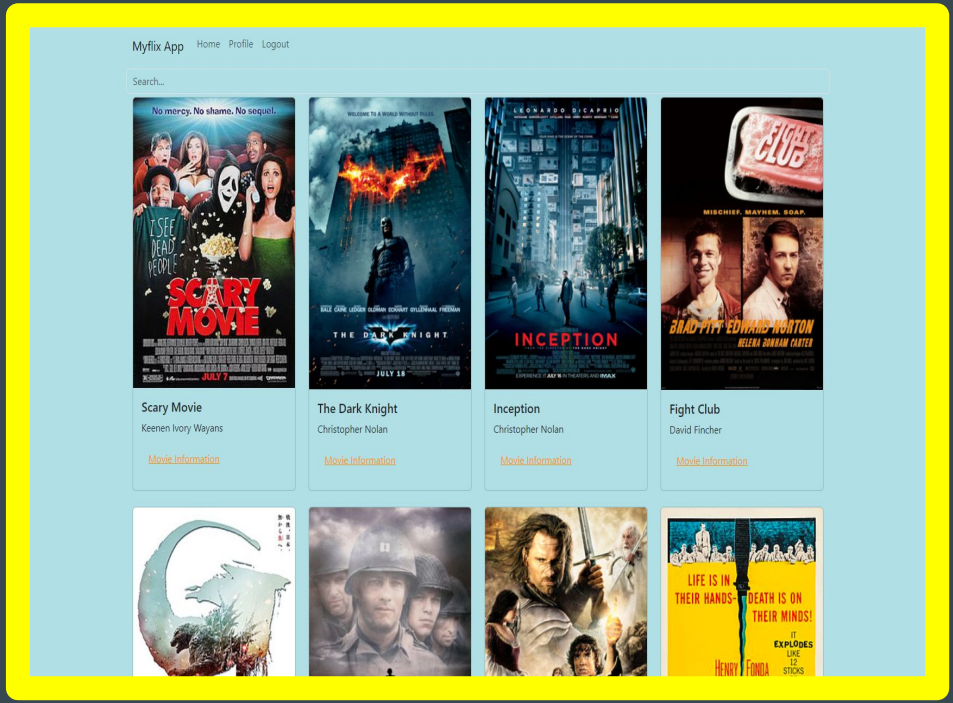
MyFlix

Jesse Sams



Overview

MyFlix is a web application that enables users to browse and save a curated list of their favorite movies. The platform provides detailed movie information, including summaries, directors, and genres. To access these features, users must register for an account. Once logged in, they can also update and customize their user profiles.



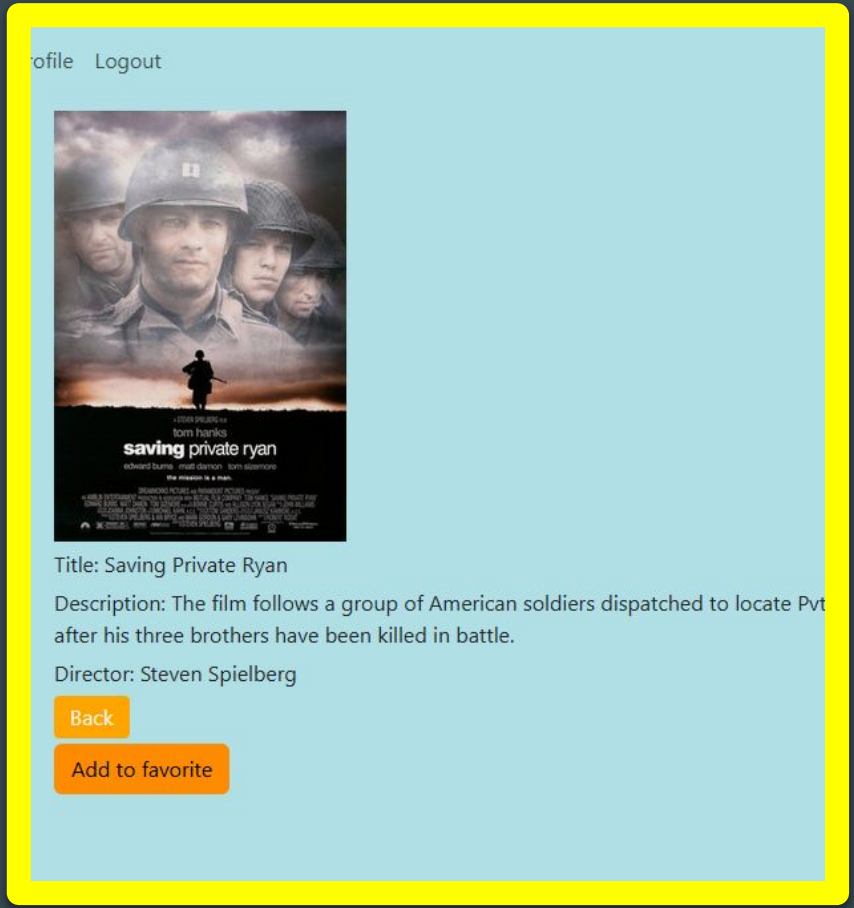
Purpose

MyFlix was a web application developed for an exercise during my tutelage at Careerfoundry.

CAREERFOUNDRY

Objective

The aim was to build a full-stack web application with a robust backend (database and API) and a responsive frontend (client-side).



Backend API

I created my own API using express and passport-jwt. This API designates all the routes and provides Security that requires users to login.

```
/**
* Get all movies.
* @route GET /movies
* @returns {object[]} List of movies.
* @async
*/
app.get('/movies',passport.authenticate('jwt', { session: false }), async (req, res) => {
  await Movies.find()
    .then((movies) => {
      res.status(201).json(movies);
    })
    .catch((err) => {
      console.error(err);
      res.status(500).send('Error: ' + err);
    });
});


/**
* Get a movie by title.
* @route GET /movies/:Title
* @param {string} Title - Movie title.
* @returns {object} Movie details.
* @async
*/
app.get('/movies/:Title', passport.authenticate('jwt', { session: false }), async (req, res) => {
  await Movies.findOne({ Title: req.params.Title })
    .then((movie) => {
      res.json(movie);
    })
    .catch((err) => {
      console.error(err);
      res.status(500).send('Error: ' + err);
    });
});
```

Backend Database

I created a Database using MongoDB to store user account information as well as all related movie information


```
_id: ObjectId('66d7f05c96575d71932710bd')
Title : "Inception"
Description : "A thief and his crew with the rare ability to enter the dreams of othe…"
▸ Genre : Object
▸ Director : Object
ImagePath : "https://upload.wikimedia.org/wikipedia/en/2/2e/Inception_%282010%29_th…"
Featured : true

_id: ObjectId('66d6a3553748d6532c2710c5')
Title : "Fight Club"
Description : "An alienated office worker and a charismatic nihilist start an undergr…"
▸ Genre : Object
▸ Director : Object
ImagePath : "https://upload.wikimedia.org/wikipedia/en/f/fc/Fight_Club_poster.jpg"
Featured : true

_id: ObjectId('66d69b813748d6532c2710bd')
Title : "Godzilla Minus One"
Description : "The movie focuses on Koichi a kamikaze pilot who abandoned his duty, h…"
▸ Genre : Object
▸ Director : Object
ImagePath : "https://upload.wikimedia.org/wikipedia/en/thumb/e/e0/Godzilla_Minus_On…"
Featured : true
```

Frontend

The Client Side of my application is designed to be functional and easy to navigate. I have a movie page where users can view all available movies and a profile page for users to edit their profile.

```
                return {
                    _id: movie._id,
                    Title: movie.Title,
                    Description: movie.Description,
                    ImagePath: movie.ImagePath,
                    Director: movie.Director
                };
            });
            dispatch(setMovies(moviesApi));
        }).catch((e) => {
            console.log(e);
        });
}, [token]);

const onLoggedOut = () => {
  setUser(null);
  setToken(null);
  localStorage.clear();
}

const onLoggedIn = (user, token) => {
    setUser(user);
    setToken(token);
    localStorage.setItem("user", JSON.stringify(user));
    localStorage.setItem("token", token);
}

const updatedUser = user => {
    setUser(user);
    localStorage.setItem('user', JSON.stringify(user));
}


return (
    <BrowserRouter>
    <NavigationBar user={user} onLoggedOut={onLoggedOut} />
```

MyFlix Information

MyFlix API Specs

Express
Passport-jwt
Bcrypt
Cors
Mongoose

MyFlix Database Specs

MongoDB
Heroku

MyFlix Client Specs

React
Bootstrap
Axios
Redux

Reflection

What I struggled with

This was still around the start of my tutelage so I really struggled to connect the client side of my application with the backend of my application

What I would do now

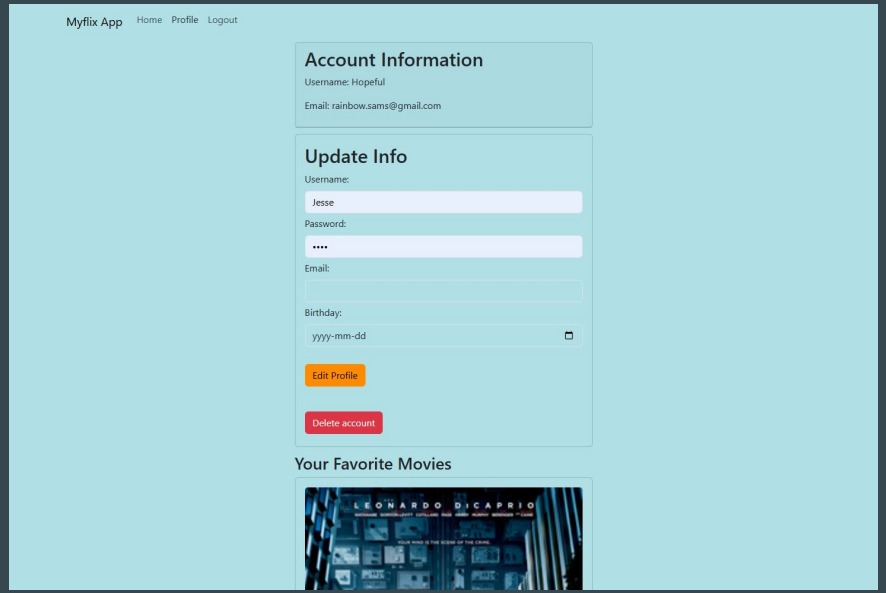
I would make the application faster then when I did before. I would also try to make a more refined and user friendly interface.

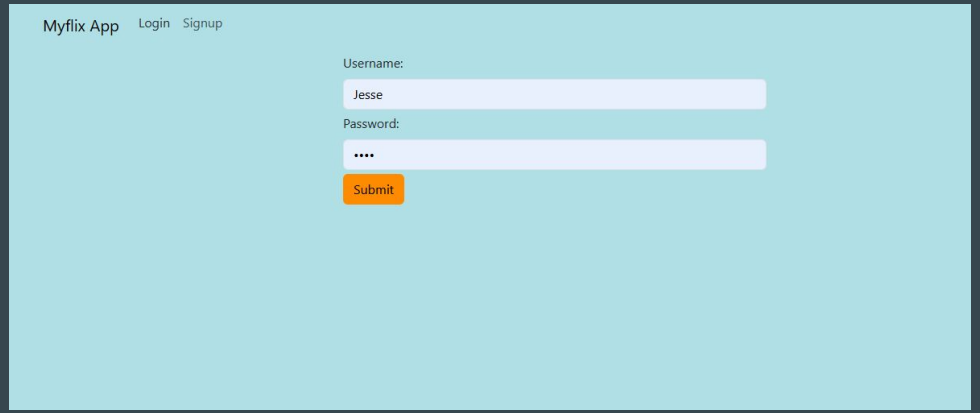
What I Excelled at

I did a good job at keeping the code short and well formatted with proper documentation.

Thank You For Reading

More Website Images





Website Link:
https://myflixfm.netlify.app/login