myFlix React App

Project Overview:

Role: Lead Developer **Mentors: Dean Gilewicz & Neal Peters**

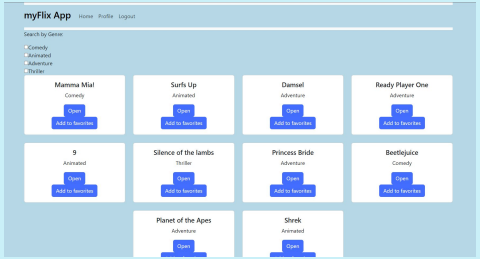
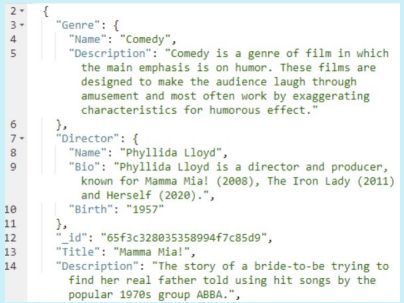
myFlix project is a single-page, responsive React application that facilitates user requests and renders the response from an existing server-side REST API and database via a number of different interface views. The goal of this application is to allow movie enthusiasts to access and save information about different movies they want to watch or recommend to their peers.

Key Features:

- Search feature that filters a list of movies by genre
- Profile view that displays user details and allows user to update their info
- Main view returns a list of movies where each button has a "More Details" button that takes the user to a page with additional details of the movie they selected

Tools, Skills, & Methodologies:

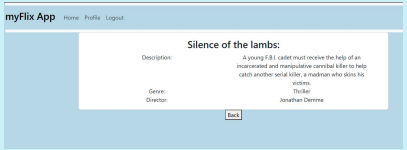
- Parcel
- Node.js
- Bootstrap
- MERN (MongoDB, Express, React, and Node.js) stack

Purpose & Context:	Objective:	Duration:
This was a personal project I built as part of my web development course at CareerFoundry to demonstrate my understanding of full-stack JavaScript development and technologies.  	• To create a full-stack project that I could add to my professional portfolio • Problem to be solved: ○ to build the client-side of the myFlix application and complete the full-stack project, while utilizing the server-side (API and database) I previously built and tools like React DOM	As a whole, this project took me the longest to complete out of all the projects throughout the CareerFoundry course. I built the complete server-side and client-side for this application from scratch. This is the project through which I gained my understanding of JavaScript and React. 

STEPS

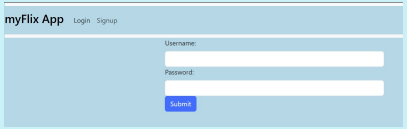
The approach started with learning the uses of different frameworks and libraries in a variety of real-world scenarios to better equip me with the ability to make decisions regarding which framework or library to use for future projects. Additionally, I learned the basics of React and Parcel, like React's sustainability and how Parcel can help expedite the build process. With this information, I used Parcel to complete the necessary build operations for the project and created the basic project structure.

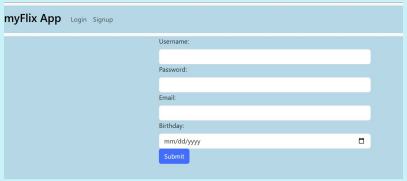
```
myFlix-client
├── .gitignore
├── package.json
├── src
│   ├── index.html
│   ├── index.scss
│   └── index.jsx
```

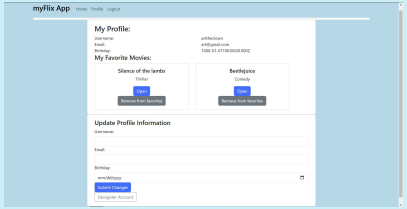


I was given a project brief with a set of technical requirements and application features. Using the project brief as a reference, I used JSX to create the MainView, MovieCard, and MovieView views and components for my app. Once the basic views and components were created, I hooked up my React client-side application with my existing server-side API using React Hooked that allowed me to render the data from my API throughout my new React components.

Next, I created login and signup forms and executed frontend form validation and enabled token-based authentication. Once the signup and login views were created, I integrated them with the MainView so that they were displayed as the first screen for non-authenticated users. With the signup and login views complete, I added styling to the application with React Bootstrap using the project brief to help guide my Bootstrap styling decisions.







The final features I added to the application were a navigation bar for users to switch between different views, a movie filter to allow users to view movies of a genre they specify, and profile view where users can alter their information and see their favorite movies. In order to implement these features, I implemented state-based routing for my application, so that users can navigate between the different views and obtain unique URLs to share views with others. Finally, I hosted my React myFlix application on Netlify.

FINAL THOUGHTS

Takeaways

In the future, one of the improvements I would like to make to my application is to add movie posters/images to each movies and display them on the MainView, MovieCard, and MovieView. I feel like this feature would advance my application to make it more user-friendly and provides more information about the movies to the users.

What didn't do well?

As a whole, this project was the largest project I completed throughout my CareerFoundry course. This resulted in dealing with a lot of small errors, like syntax errors or simple misunderstandings, as I was still getting the hang of JavaScript while completing this project. I think that it was due to these factors that caused the amount of errors I experienced.

What went well?

Like I previously stated, in combination with the server-side of this project, this project was the largest that I've completed. While this project did take me the longest to complete, it didn't take me as long to complete as I thought it would. This project helped me hone my debugging skills, which is a crucial skill in my future career in web development.