PROJECT TITLE: MYFLIX CLIENT

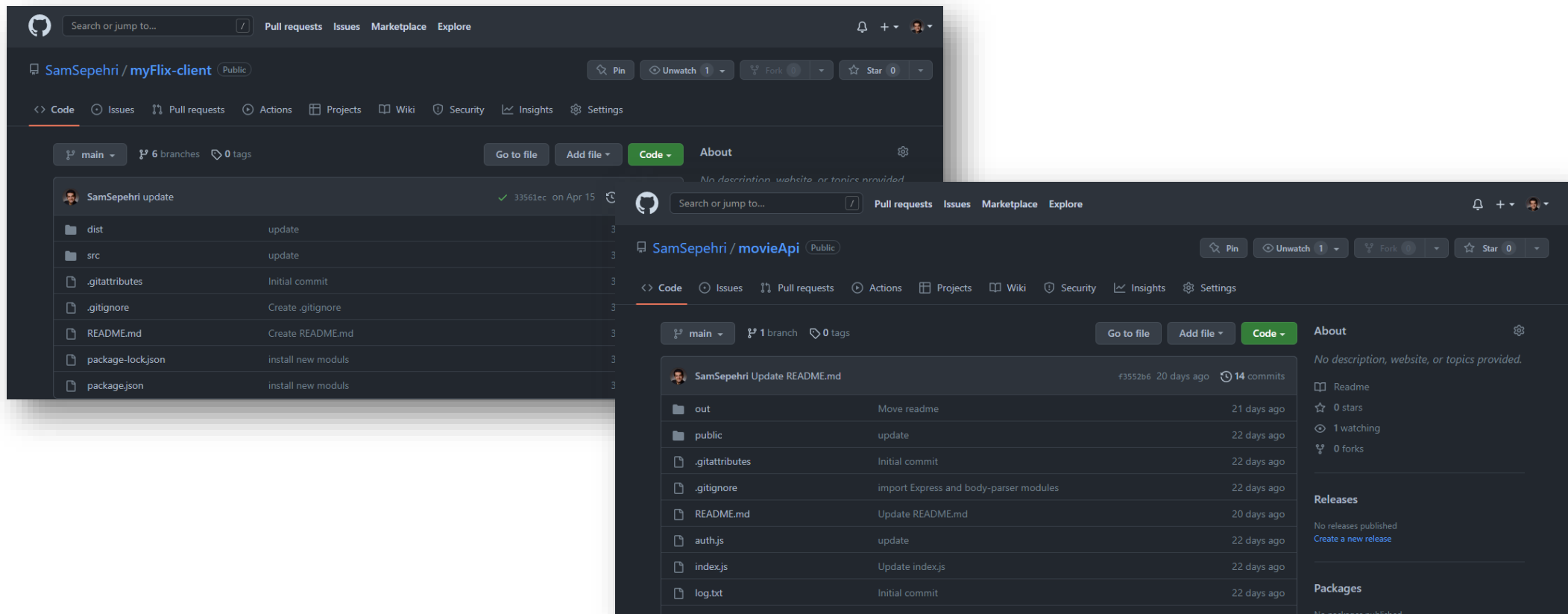Sam Sepehri

Overview :

MyFlix is a web-base movie database application, developed using the MERN stack, that provides users with access to information about movies, directors, and genres. Users are able to create a user-account, update their profile, and create a list of favorite movies.

Purpose & Context :

This project was a my personal project that I built as one of my web development courses at CareerFoundry to show my knowledge of full-stack JavaScript development.
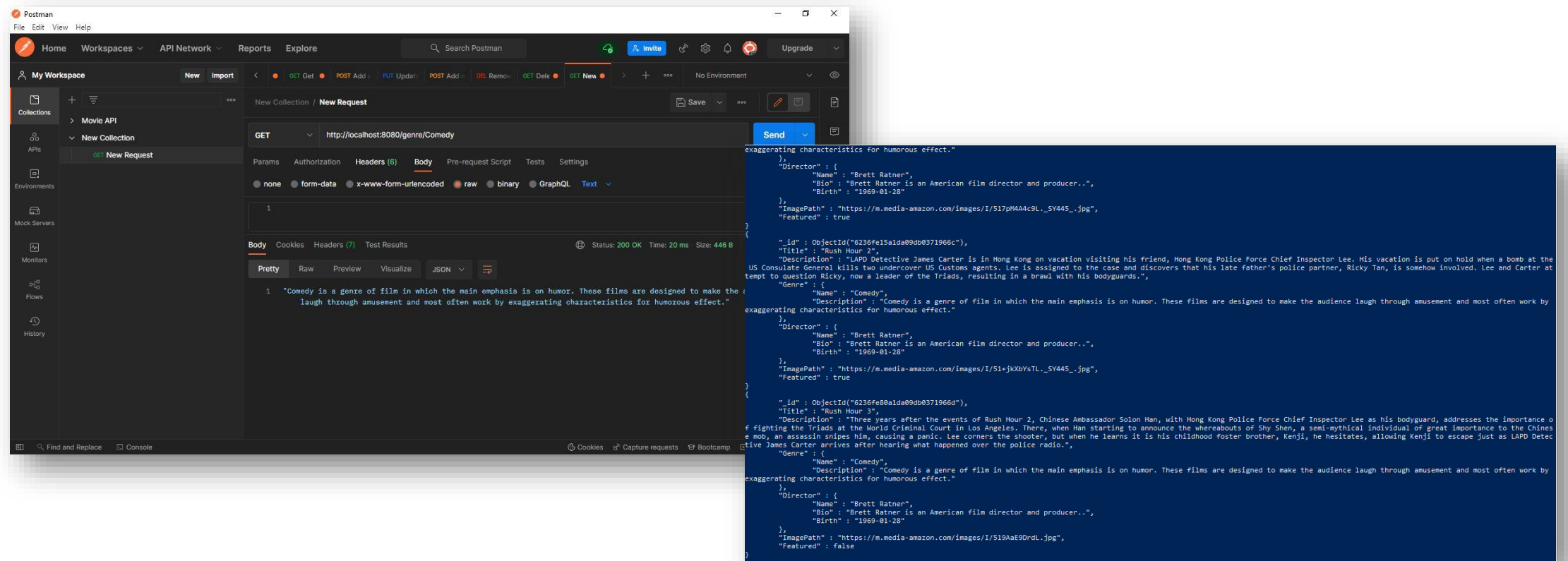
Objective :

The goal of the project was to create a web application using React to build the client-side for an application called myFlix, based on its existing server-side code (REST API and database. The problem I wanted to solve is to build the server-side and client-side for the application from scratch. this application is created to enable users to create an account and perform a certain number of actions, apart from searching movie titles, namely:

- Allows users to log in with a username and password

- Returns data (description, genre, director, image)

- Sorting and filtering

- Add a movie to a favorites list

- Modify their account information (personal information, favorite movies)

- Allows existing users to deregister

Approach :

Server-Side: I`ve created a RESTful API using Node.js and Express, that interacts with a non-relational database (MongoDB). The API can be accessed via commonly used HTTP methods like GET or POST. To retrieve data from and store data in the database, CRUD methods are used. The API provides movie information in JSON format.
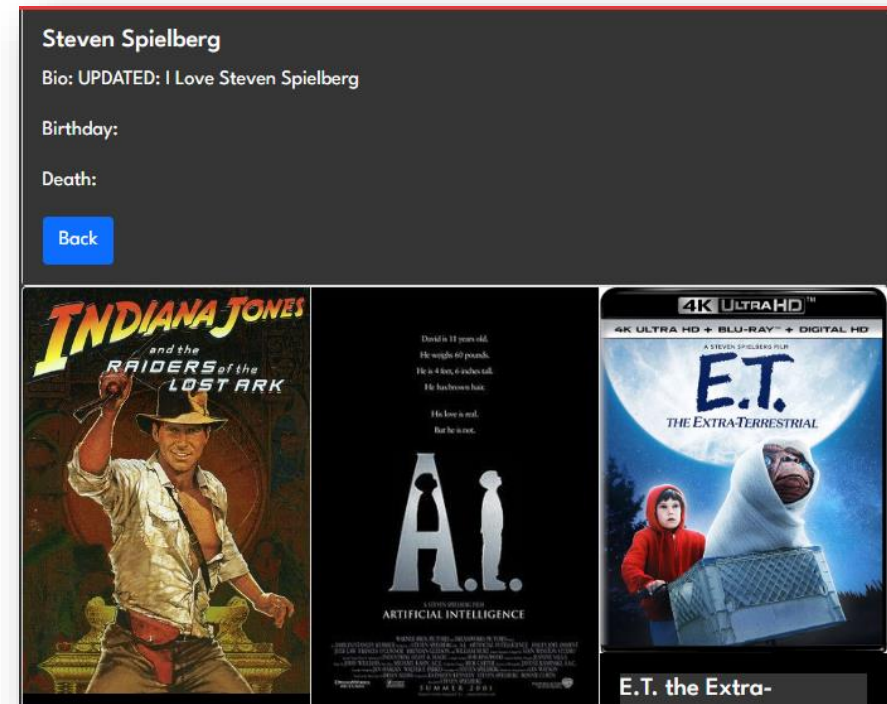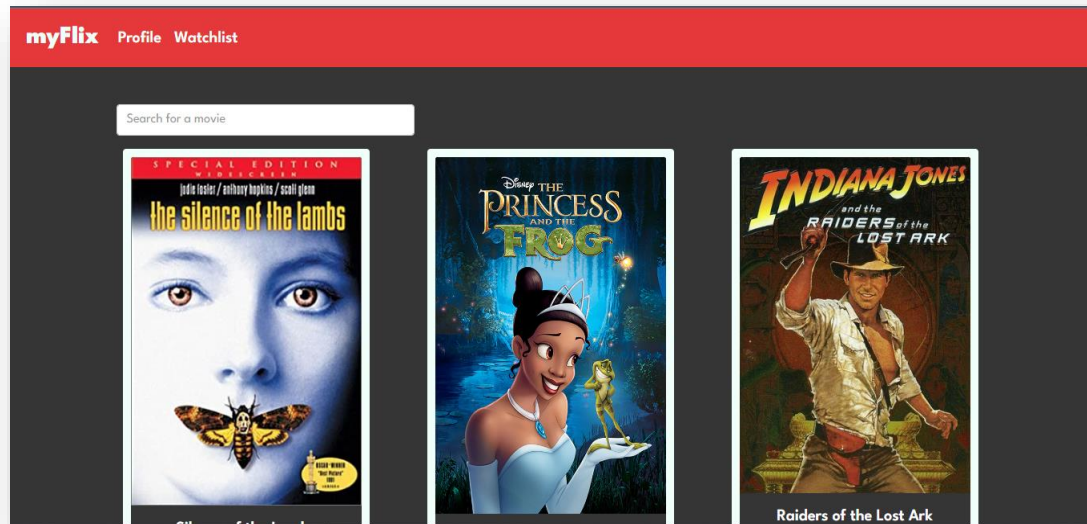
Client-Side

After completing the API, I started to build the interface. users would need when making requests to, and receiving responses from, the server-side. It is a single-page, responsive application, developed with React and React-Redux. It provides several interface views, including, but not limited to, a main view (shows a list of all movies), single movie view (shows data about a single movie and allows users to add the movie to their list of favorites), a login view, a registration view and a profile view (where users can update their user data and list of favorites).

Challenges

This was both my favorite and most challenging project. I have enjoyed building the API and working with   +
database structures.   •

The biggest challenge I faced was linking my end-points in front-end and back-end together. while implementing and routing the end-points, the capitalization and lowercase is very important point and it needs to pay attention to right spelling on both side. To have a better understanding about this coupling between back-end and front-end, I wrote my movieApi from scratch.

Besides that, I also did a lot of research in Google or Stackoverflow website to find more information about Client-side app routing.

Duration

The development of the client-side took me twice as long as it normally took to complete a CareerFoundry achievement. It was because I needed time to wrap my head around how React works.