

## Backend – Achievement 3 – Client-Side Programming & React

### 1) Description of the project

- What was my role for this project and what tasks did I face?
  - building a client-side frontend for my RESTful API, built with React
  - building the interface users use when making requests to—and receiving responses from—the server-side
  - including several interface views (built using the React library) that will handle data through the (previously defined) REST API endpoints

### Requirements

#### Main view

- Returns ALL movies to the user (each movie item with an image, title, and description)
- Filtering the list of movies with a “search” feature
- Ability to select a movie for more details
- Ability to log out
- Ability to navigate to Profile view

#### Single Movie view

- Returns data (description, genre, director, image) about a single movie to the user
- Allows users to add a movie to their list of favorites
- Login view
- Allows users to log in with a username and password
- Signup view
- Allows new users to register (username, password, email, date of birth)
- Profile view
- Displays user registration details
- Allows users to update their info (username, password, email, date of birth)
- Displays favorite movies
- Allows users to remove a movie from their list of favorites
- Allows existing users to deregister

### Optional Views & Features

#### Actors view

- Allows users to view information about different actors

#### Genre view

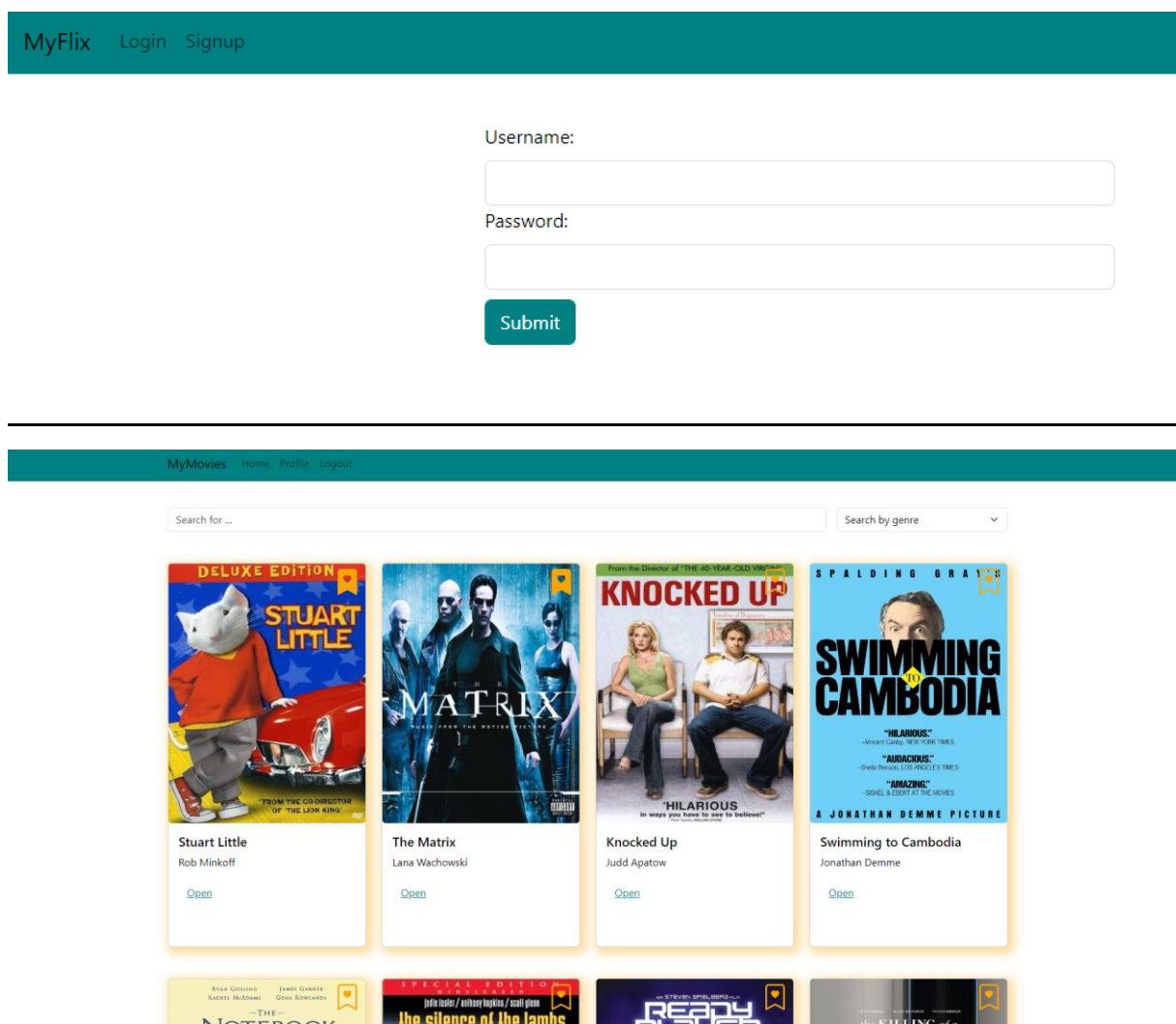
- Returns data about a genre, with a name and description
- Displays example movies

#### Director view

- Returns data about a director (name, bio, birth year, death year)
- Displays example movies from the director

- Lessons I learned / decisions I made during this project
  - introduction to Frameworks and Libraries and deciding when to use them for my project,
  - getting to know and using React as a framework for my client-side RESTful API,
  - React components such as state, props, the virtual DOM
  - developing components, using Parcel,
  - using Hooks in function components to interact with the React component lifecycle
  - handling flows for signup, login, user authentication, form validation
  - using bootstrap in React
  - client-side app routing, navigation between different views
  - design patterns like redux
  - deployment and hosting
  - readability and maintenance of my codebase and the design and usability of my app

## 2) A screenshot to represent the project



3) A link to the project's GitHub repository

- [https://github.com/Luisa-Inc/myFlix\\_client](https://github.com/Luisa-Inc/myFlix_client)

4) A link to the live, hosted version of my app

- <https://myluisaflix.netlify.app/>

5) A list of the technologies used for each project

- React
  - ReactDOM
  - React-Router-Dom
  - Bootstrap
  - React-Bootstrap
  - React-Bootstrap-Icons
  - Prop-Types
  - Moment
  - Parcel/Transformer-Sass (v.2.10.2)
  - Parcel (v.2.10.2)
  - Process