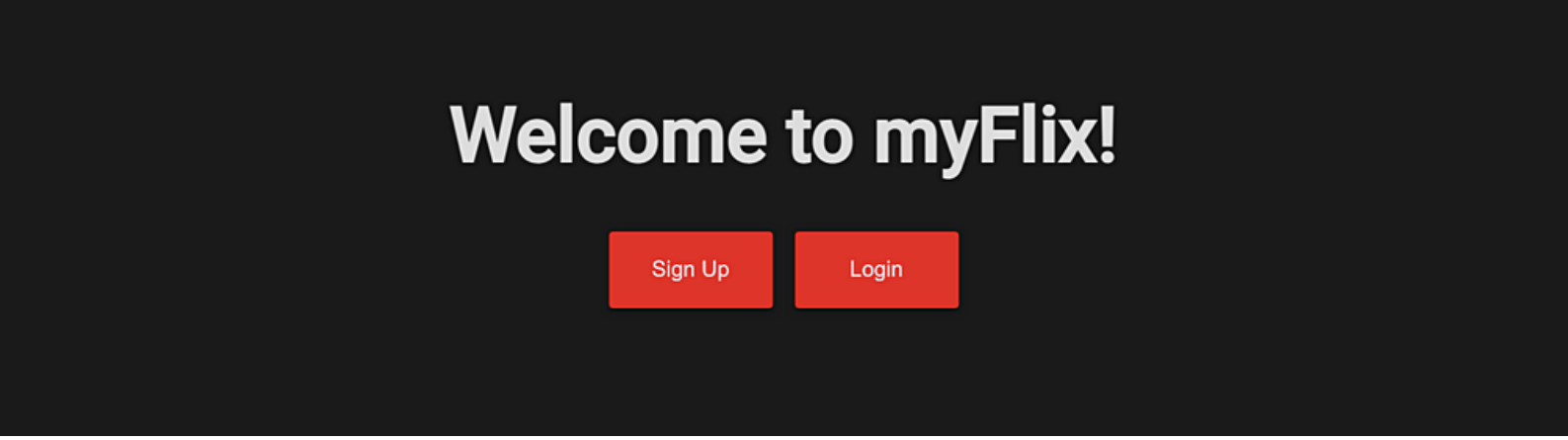
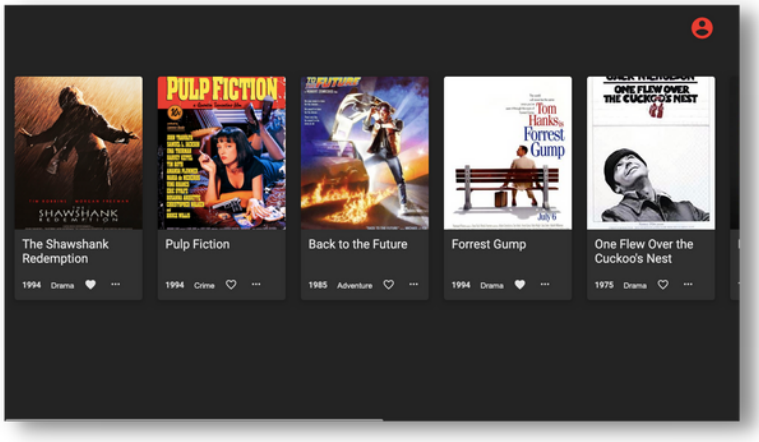
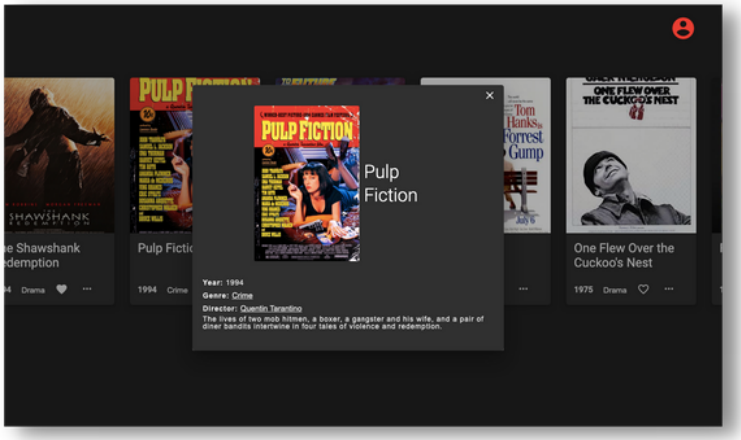


Case study for myFlix Angular App

Overview

MyFlix is a single-page web application that enables users to register and log in to view a collection of movie titles with detailed information, add movies to their favorites, and view, edit, or delete their profile. The app was developed using Angular.js and built on top of a previously developed RESTful API.



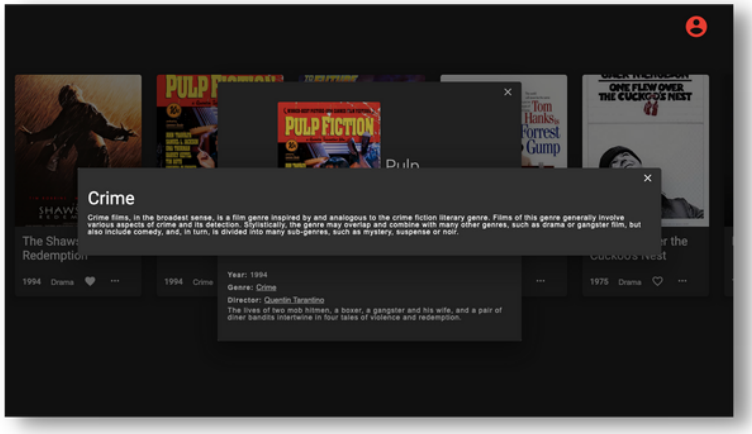


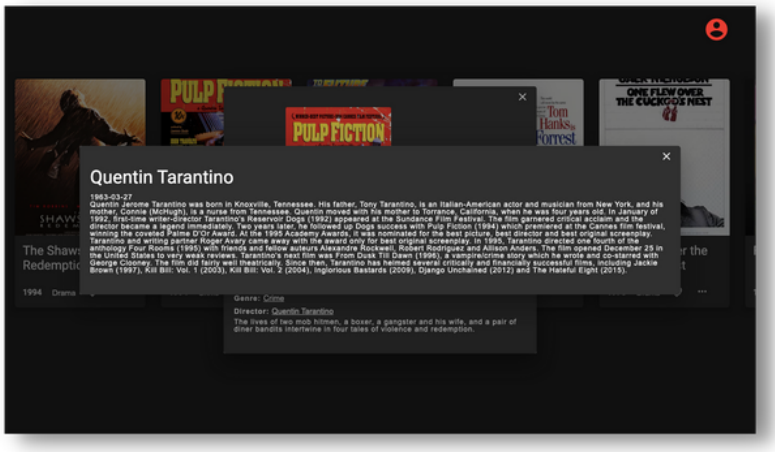
Objective

To demonstrate my proficiency in web development by utilizing Angular to build the client-side of an application called MyFlix based on its existing server-side code and showcase the ability to effectively utilize various technologies such as Node.js, Express, and MongoDB.

Context

Angular is an extremely popular framework that can be used for a wide range of mobile and desktop applications. It's particularly well-suited to large projects, as well as projects requiring a complex user interface because it comes with a variety of built-in modules and services. In order to be able to develop the project, first I had to understand the basics of TypeScript - the preferred language of Angular.





Features

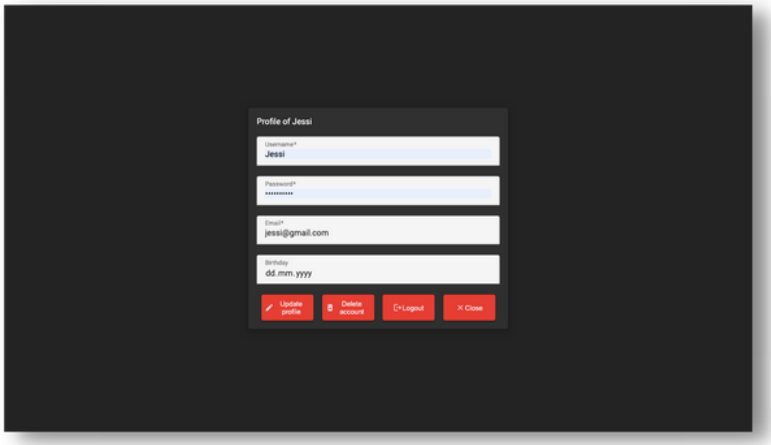
When a user enters the website he is greeted by a welcome message and options to log in or sign up. The user then is redirected to the main page which has a list of movies. The user can access the description, genre, and director information by clicking on the info button, and add any movie to the list of his favorite movies by clicking on the heart button. The user can also see, edit or delete his profile by accessing the profile page in the top right corner.

Approach

- To begin, I set up the Angular app using the Angular CLI and integrated a previously developed movie API to fetch all necessary data. I then created the welcome page and forms for login and user registration.
- The next stage of development involved creating views for movie lists and user profiles, utilizing Angular's router to implement routes for the welcome view, movie card view, and user profile view. I also added functionality such as links on movie cards to open movie information modals and a "like" button to add movies to a favorites list.
- One of the most time-consuming tasks was creating modals for general movie information, as well as genre and director information modals.
- Once the application functionality was complete, I moved on to the fun part: designing the user interface. I utilized SCSS and Angular Material, which streamlined the process.
- Finally, I wrote comprehensive documentation and hosted the app on GitHub pages

Challenges

This project presented a few challenges for me. Navigating Angular's steep learning curve and adjusting to its differences from the React framework I am used to was a hurdle. Additionally, working with TypeScript, which has a distinct syntax from the JavaScript I am familiar with, was another challenge. However, despite these obstacles, I found the development process to be enjoyable and I am now well-versed in the basics of Angular and TypeScript, and have a valuable addition to my web development portfolio.



View my project's hosted version and GitHub repo:

Live **GitHub**

The project was created by Vitaliy Porsnev as a part of CareerFoundy Full-Stack Web Development Course.