myFlix

with Angular

Overview

The myFlix Angular client is a web application that serves as the front-end for an API I've created earlier. It provides users with access to information about movies. The users are able to register and create a list with their favorite movies.

TECH STACK

- TYPESCRIPT
- HTML
- SCSS

Context

myFlix was built from scratch as a part of my 6-month CareerFoundry Full-Stack Web Development course. The goal is to demonstrate my skills in full-stack JavaScript web development.

Objective

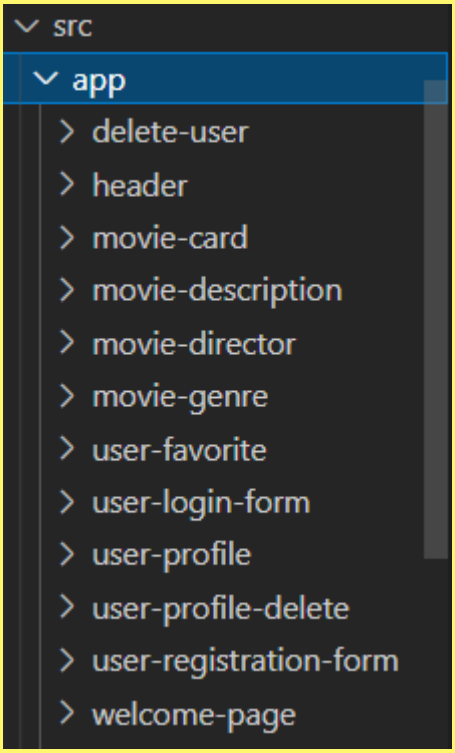
The aim of this app is to become a part of my portfolio as an ambitious full-stack project. The problem that I had to solve is to adapt the same structure of the client-side that I've already created in React to Angular.

THE PROJECT

Structure.

The components for this project were already decided by a previous task in the course. There they were created using **React** and the goal in this project was to re-write them with **Angular.**

List of components.



Step-by-step.

1. Welcome page
2. Movie list
3. Profile view
4. Style

THE WELCOME PAGE

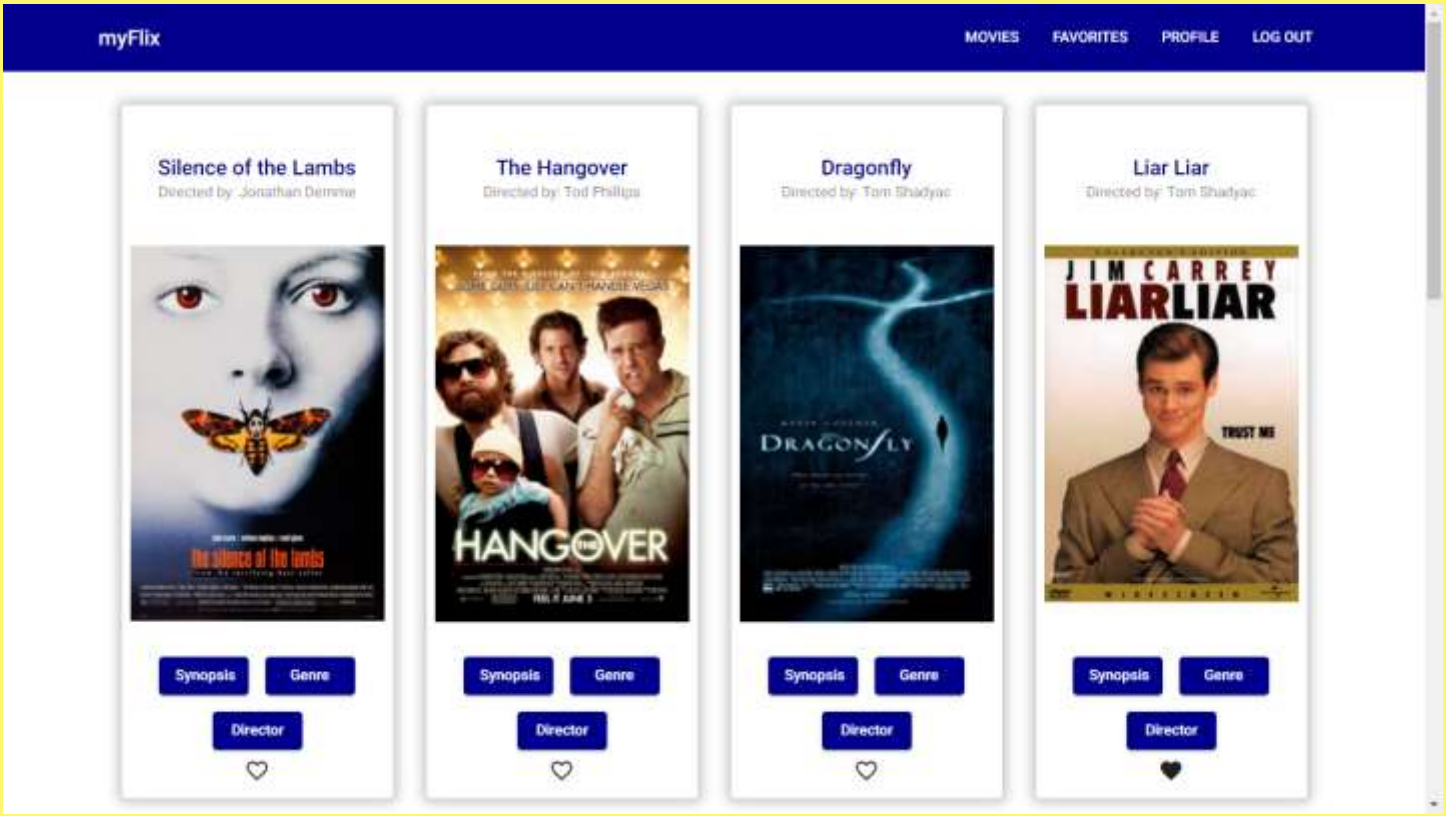
I wanted to create a very simplistic welcome page without unnecessary functions and confusing UI. I had to choose between centered buttons and no background or a photo as a background and the buttons on the top right side. I therefore created a poll on in my Instagram and the first option won by 80%. So when the design was clear I had to familiarize myself with Angular Material and implement it.

The design and functionality of the registration component were already decided by the task. Since, I was already familiar with Angular Design I did the registration form fairly quickly and was able to consolidate what I have learned when creating the welcome page.

```
<mat-card>
    <mat-card-header>
      <mat-card-title>Sign Up!</mat-card-title>
    </mat-card-header>
    <mat-card-content>
      <form>
        <mat-form-field>
          <input
            matInput …
```

THE MOVIE LIST

- The movie list is the first component that a logged in user sees
- It shows a list of all movies
- The list is displayed using movie cards
- The movie cards have a button to add a movie to or remove it from user's favorite movies list



The movie list component is the most important in the app. It consists of all movies from the database. From there the users have the option to add a movie to their favorites list.

About the design.

The idea the movies to be shown as movie cards was inspired by the VHS tapes covers in the 80s. Adding movies to your favorites list resembles a collection of tapes you would buy and store at home if it was 1985.

Did everything go well?

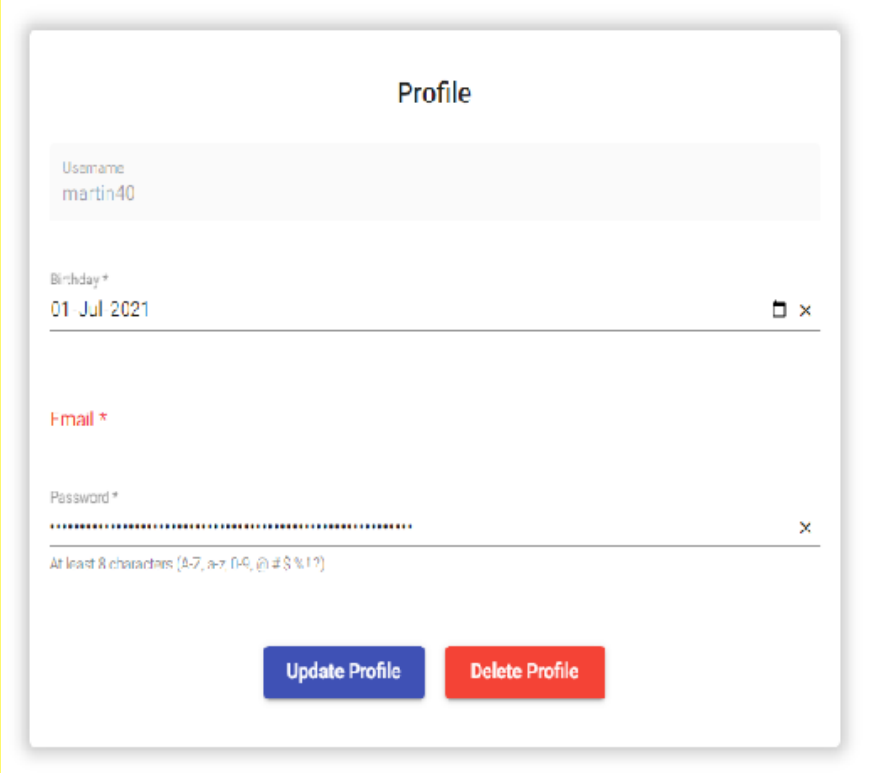
Well, no. Since I was dealing with Angular's structure for the first time, I had to figure out how do the template files and the .ts files work together. This took me some time. I had to take a look at the database and the schemas in order to connect the dots.

After I understood the logic I easily implemented it in the Favorites List component as well. This process highlighted also the main differences between React and Angular.

The most challenging part of this component was to create the function that adds a movie to the favorites list. It threw error 401 Unauthorized. The issue turned out to be in the POST method. It expected three parameters – URL, body and options but instead I had provided only 2. The options were replacing the body and therefore the 401. This got me back to the phase in which I created the API and the endpoints. So in conclusion, with the issues in this Angular project I covered all the steps once again from creating the database with Mongo to the design part.

THE PROFILE VIEW

This is the place to edit or delete a user. It's a compulsory component for apps that provide an option to register. It is meant to be similar to all registration forms online so the users can easily navigate through. It is again built using Angular Material in order the dialogs to look familiar as this page is about user information it had to show the current information that's in the database. So when a user goes to their profile they can see what they've shared.



What was the tricky part?

The endpoint to get a user by username was missing. So showing the information for the current user was not possible. After a bit of wondering I figured it out and again went back to the server-side of the app and simply created an endpoint. An easy solution which took much more time to think of than to write. And that's it the project was almost done. I only had to choose the colors.

STYLE

I wanted to create a responsive app and therefore the looks of all components in this app are done using SCSS. The challenge here is to make it look good on all devices. Following the best practices in web design I have chosen well-contrasting colors that improve readability and make it easier for the customers to navigate through. The design of the app is simple because its main purpose is to demonstrate my mastery of Angular.

```scss
.custom-card-header {
  min-height: 125px;
  display: flex;
  align-items: center;
  justify-content: center;
  text-align: center;
}

.custom-card-title {
  color: darkblue;
}

.custom-card-content {
  height: 350px;
  padding: 0.5rem;
}

.custom-card-actions {
  padding: 0.5rem;
  text-align: center;
}

.button-card {
  width: 85px;
  background-color: darkblue;
  color: white
}
```

SUMMARY

This was the last project of my 6-month Full-Stack Web Development course and it was the most challenging. As I've mentioned the structure was already familiar because I had already worked on it with a previous project which had the same components but it was created by using React. This project helped me to understand the differences between Angular and React – the most popular JavaScript Frameworks.

Because of some specifics of the two approaches I also had to add several endpoints to the API needed for different methods in Angular. This was a good opportunity to exercise previously learned skills and play around with tools like Postman.

In general, I enjoyed the process of creating this app and who knows, maybe this is how IMDB started?

There is a link to the app:

https://martink21.github.io/myFlix-Angular-client/

There is a link to the repository on GitHub:

https://github.com/martink21/myFlix-Angular-client