CAREERFOUNDRY

Achievement 3 Project: myFlix React app

CF

Objective

Using React, build the client-side for an app called myFlix based on its existing server-side code (REST API and database).

Context

Client-side development hasn't always been so prominent. In the past, pages would be generated on the server-side and sent to the browser, resulting in a poor user experience. Thanks to modern browsers and libraries such as React, the client-side of an app is today considered to be just as important as the server-side. Full-stack developers need to be well-versed in both server-side and client-side development.

In the previous Achievement, you built the server-side for a movie app called myFlix. The API and database that you built meet the information needs of myFlix users. Now, you need to create the interface they'll use when making requests to—and receiving responses from—the server-side. The client-side of your myFlix app will include several interface views (built using the React library) that will handle data through the (previously defined) REST API endpoints.

The code you write impacts both your users and your fellow developers. As you work through this Achievement, you'll need to consider, among other things, the readability and maintenance of your codebase, and the design and usability of your app.

By the end of the Achievement, you'll have a complete web app (client-side and server-side) built using full-stack JavaScript technologies, which you can then showcase in your portfolio. This project will demonstrate your mastery of full-stack JavaScript development. The complete tech stack you'll master is known as the MERN (MongoDB, Express, React, and Node.js) stack.

The 5 Ws

1. **Who**: The users of your myFlix app—movie enthusiasts who enjoy reading information about different movies.
2. **What**: A single-page, responsive app with routing, rich interactions, several interface views, and a polished user experience. The client-side developed in this Achievement will support the existing server-side (from Achievement 2) by facilitating user requests and rendering the response from the server-side via a number of different interface views.
3. **When**: myFlix users will be able to use it whenever they want to read and save information about different movies.
4. **Where**: The app will be hosted online. The myFlix app itself is responsive and can therefore be used anywhere and on any device, giving all users the same experience.
5. **Why**: Movie enthusiasts like to be able to access information about different movies, whenever they want to. Having the ability to save a list of their favorite movies will ensure users always have access to the films they want to watch or recommend to their peers.

Design Criteria

User Stories

- As a user, I want to be able to access information about movies so that I can learn more about movies I've watched or am interested in.
- As a user, I want to be able to create a profile so I can save data about my favorite movies.

Features & Requirements

The following feature requirements were extracted from the user stories just listed. **Please note**, your project will only be approved if the following essential feature requirements are implemented in your Achievement project.

Essential Views & Features:

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

Single Movie view (optional features)

- Allow users to see which actors star in which movies
- Allow users to view more information about different movies, such as the release date and the movie rating
- Allow users to access different movie information, such as genre description and director bio, without leaving the view (e.g., tooltips)
- Allow users to share a movie
- Display a list of related or similar movies

Main view (optional features)

- Allow users to sort movies based on different criteria

Profile, Single Movie, and Main views (optional features)

- Allow users to create a “To Watch” list in addition to their “Favorite Movies” list

Wireframes

You can download wireframes for each of the views for your project here:

- [MYFLIX PROJECT WIREFRAMES (.zip)]

Technical Requirements

- The application *must* be a single-page application (SPA)
- The application *must* use state routing to navigate between views and share URLs
- The application *must* give users the option to filter movies using a “search” feature
- The application *must* use Parcel as its build tool
- The application *must* be written using the React library and in ES2015+
- The application *must* use Bootstrap as a UI library for styling and responsiveness
- The application *must* contain function components
- The application *must* be hosted online
- The application *may* use React Redux for state management of at least one feature (i.e., filtering movies)