



# MyFlix app, React version

A case study by Challis Regan, Lead Developer

Tutor: Davi Morais Ferreira

Mentor: Akunna Nwosu



Open



Open



Open





# Overview

- [MyFlix](#) is a single-page web application built with React, which provides users with access to information about different movies, directors, and genres. Users are able to sign up, update their personal information, and create a list of their favorite movies.
- MyFlix interacts with [MyFlix API](#), a custom API that contains all data relevant to the MyFlix app, including information about movies, titles, descriptions, genres, directors, images, features, users, and more.
- It is a demo project I built for my web development course at CareerFoundry. I finished this project in April 2024.
- <https://cmr927-myflix.netlify.app/>



# Technical Stack



- React
- React-Bootstrap
- React-Router
- Parcel-Bundler
- HTML5
- SCSS
- JavaScript
- Netlify
- bootstrap
- dayjs
- prop-types
- process



# 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.

## Code sample from main-view.jsx

```
1  import { useState, useEffect } from "react";
2  import { MovieCard } from "../movie-card/movie-card";
3  import { MovieView } from "../movie-view/movie-view";
4  import { LoginView } from "../login-view/login-view";
5  import { SignupView } from "../signup-view/signup-view";
6  import { NavigationBar } from "../navigation-bar/navigation-bar";
7  import { MoviesFilter } from "../movies-filter/movies-filter";
8  import { Col, Row, Container } from "react-bootstrap";
9  import { BrowserRouter, Routes, Route, Navigate } from "react-router-dom";
10 import PropTypes from "prop-types";
11 import { ProfileView } from "../profile-view/profile-view";
12
13 ✓ export const MainView = () => {
14     const storedUser = JSON.parse(localStorage.getItem("user"));
15     const storedToken = localStorage.getItem("token");
16     const [user, setUser] = useState(storedUser ? storedUser : null);
17     const [token, setToken] = useState(storedToken ? storedToken : null);
18     const [movies, setMovies] = useState([]);
19     const [filter, setFilter] = useState("");
20 ✓   const onLoggedOut = () => {
21       setUser(null);
22       setToken(null);
23       localStorage.removeItem("token");
24       localStorage.removeItem("user");
25     }
26
27     useEffect(() => {
28       if (!token || token === null) {
29         return;
30       }

```




# Building process

- Distinguish between states and props of React components
- Create React components using JSX
- Create connected React views for a single-page application (SPA)
- Integrate child and parent components effectively
- Compare how to use class components and function components throughout
- Describe the React component lifecycle
- Use Hooks in function components to interact with the React component lifecycle
- Render data from an API in React components
- Create forms with React
- Execute frontend form validation with React
- Apply authentication logic to a React form





# Building process continued

- 
- Apply the principles of good design using UI frameworks
  - Style an app using React Bootstrap features and components
  - Implement navigation between views
  - Implement state-based routing to route views in a React SPA
  - Describe the data lifecycle in a Redux app
  - Design a Redux app
  - Deploy the app using Netlify





# Retrospective



- This was my first project connecting my own API with the front-end and designing how the user interacts and finds their movies. In retrospect, I wish I would have picked a specific theme or sub-genre for my movies (time travel, sailing, etc.). I could re-design this project in the future.

Thank you! Feel  
free to connect  
with me. 😊

[Challis.regan1@gmail.com](mailto:Challis.regan1@gmail.com)

[GitHub](#)

[Portfolio](#)

