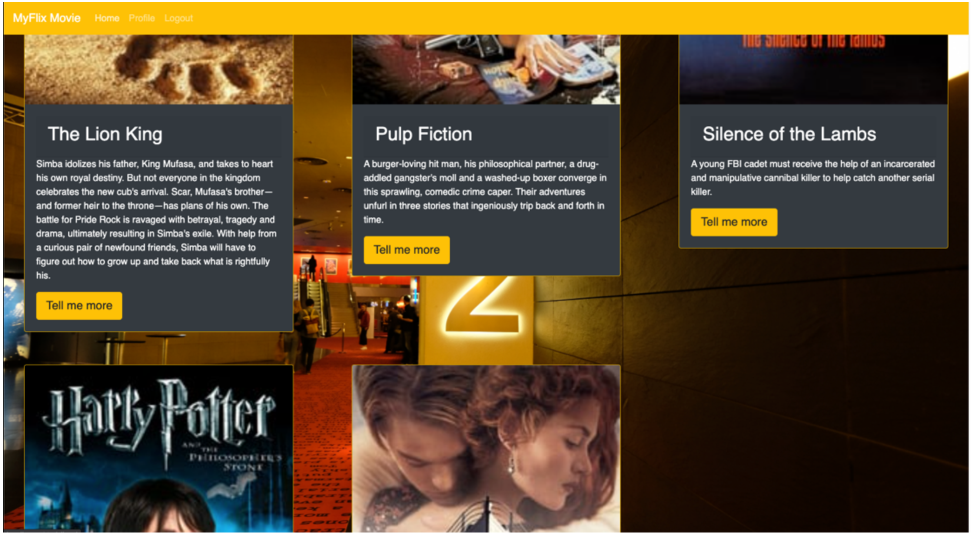
Case study for MyFlix-Movie-app

Image 1: screenshot mainpage MyFlix Movie



The Lion King

Simba idolizes his father, King Mufasa, and takes to heart his own royal destiny. But not everyone in the kingdom celebrates the new cub's arrival. Scar, Mufasa's brother—and former heir to the throne—has plans of his own. The battle for Pride Rock is ravaged with betrayal, tragedy and drama, ultimately resulting in Simba's exile. With help from a curious pair of newfound friends, Simba will have to figure out how to grow up and take back what is rightfully his.

Pulp Fiction

A burger-loving hit man, his philosophical partner, a drug-addled gangster's moll and a washed-up boxer converge in this sprawling, comedic crime caper. Their adventures unfurl in three stories that ingeniously trip back and forth in time.

Silence of the Lambs

A young FBI cadet must receive the help of an incarcerated and manipulative cannibal killer to help catch another serial killer.

Overview

MyFlix is a Movie web application consisting of a database with various movies and a client-side providing users with several features the access to information about different movies, directors and genres. Users are able to sign in and create a list of their favorite movies.

Role: Lead Developer

Link: https://myflix-movie.herokuapp.com/

Technologies used: MERN Stack (MongoDB, React, and NodeJS), Redux and Flux.

Duration: 3 month

Credits: Ali El Zoheiry (Tutor),
Akunna Nwosu (Mentor)

Purpose & Context

This project was part of the Career Foundry Web Development course. It is a perfect example to demonstrate my mastery of full-stack JavaScript development.

Objective

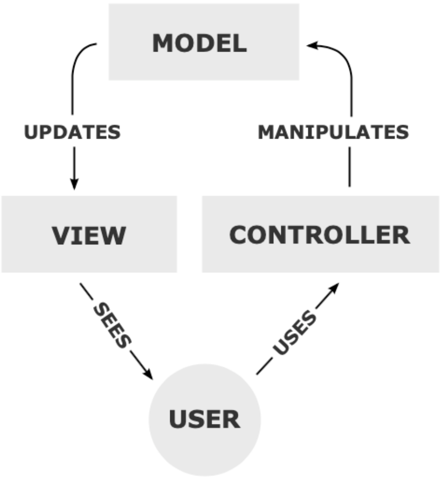
The aim of the project was to master a complete web application with a frontend and backend, using full-stack JavaScript technologies which I can showcase in my portfolio. Also, I had a personal objective to learn more about the tech architecture and be better equipped to make decisions regarding which framework or library to use for my future projects.

Process

- The first step of my **development process** was to built the database and write the back-end logic of how to access the application.
- The next step was to develop the interactive front-end that enables users to experience the app's features.
- Use version control along the way to track changes and host the code on GitHub.

Image 2: Using MVC architecture to scructure the code that makes up a client layer

Source: Wikipedia Commons

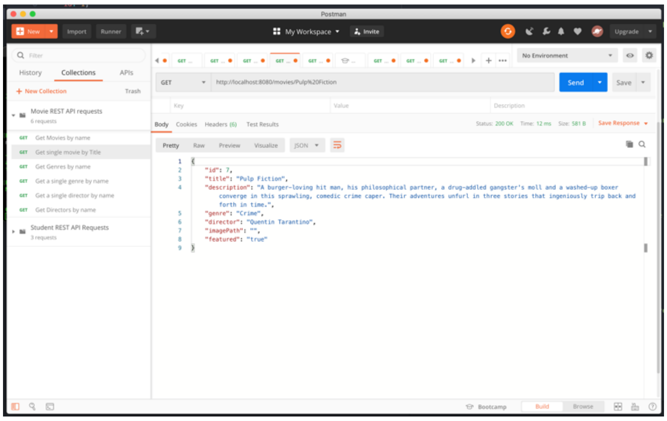


The **server side** of the web application consists of a REST API and a database built with JavaScript, Node.js, Express, and MongoDB. The API can be accessed via commonly used HTTP methods like GET, PUT, POST and DELETE. For retrieving data from the database and store in a non-relational way, I used the CRUD operations “create”, “read”, “update” and “delete”. The API provides movie and user information in JSON.files

I used **Postman** for endpoint testing and Mongoose for the business layer logic. The database was built with MongoDB; the final project is hosted on Heroku.

For more information about each *endpoint* of the REST API, see *documentation.hmtl*

Image 3: Testing Endpoints with Postman
In this example a GET request for a single movie by title.

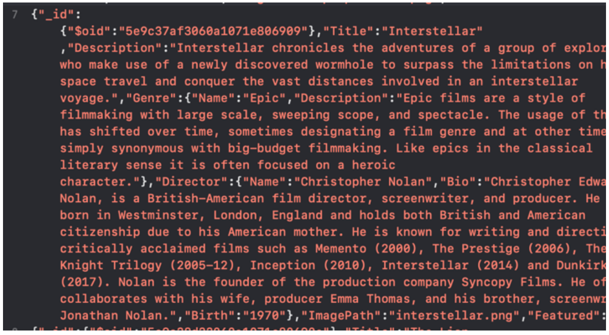


GET http://localhost:8080/movies/Pulp%20Fiction

Status: 200 OK Time: 12 ms Size: 581 B

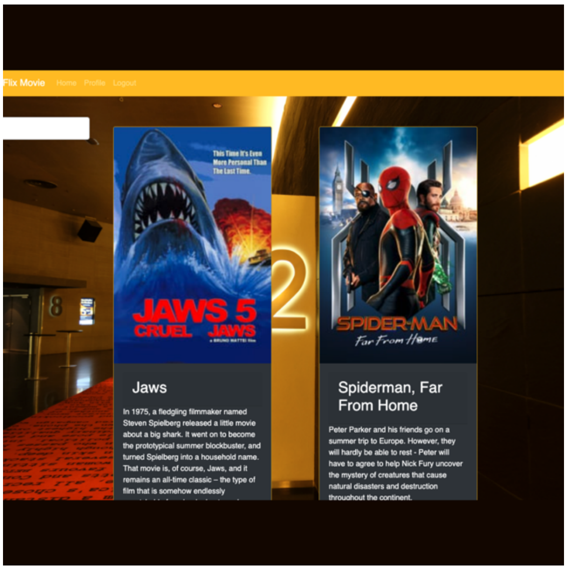
```
{
    "id": 7,
    "title": "Pulp Fiction",
    "description": "A burger-loving hit man, his philosophical partner, a drug-addled gangster's moll and a washed-up boxer
        converge in this sprawling, comedic crime caper. Their adventures unfurl in three stories that ingeniously trip back and
        forth in time.",
    "genre": "Crime",
    "director": "Quentin Tarantino",
    "imagePath": "",
    "featured": "true"
}
```

Image 4: Example of a JSON.file



```
{"_id":
    {"$oid":"5e9c37af3060a1071e806909"},"Title":"Interstellar"
    ,"Description":"Interstellar chronicles the adventures of a group of explor
    who make use of a newly discovered wormhole to surpass the limitations on h
    space travel and conquer the vast distances involved in an interstellar
    voyage.","Genre":{"Name":"Epic","Description":"Epic films are a style of
    filmmaking with large scale, sweeping scope, and spectacle. The usage of th
    has shifted over time, sometimes designating a film genre and at other time
    simply synonymous with big-budget filmmaking. Like epics in the classical
    literary sense it is often focused on a heroic
    character."},"Director":{"Name":"Christopher Nolan","Bio":"Christopher Edwa
    Nolan, is a British-American film director, screenwriter, and producer. He
    born in Westminster, London, England and holds both British and American
    citizenship due to his American mother. He is known for writing and directi
    critically acclaimed films such as Memento (2000), The Prestige (2006), The
    Knight Trilogy (2005-12), Inception (2010), Interstellar (2014) and Dunkirk
    (2017). Nolan is the founder of the production company Syncopy Films. He of
    collaborates with his wife, producer Emma Thomas, and his brother, screenwr
    Jonathan Nolan.","Birth":"1970"},"ImagePath":"interstellar.png","Featured":
```

Image 5: Example of MyFlix Movie, added movied to favorites



Jaws

In 1975, a fledgling filmmaker named Steven Spielberg released a little movie about a big shark. It went on to become the prototypical summer blockbuster, and turned Spielberg into a household name. That movie is, of course, Jaws, and it remains an all-time classic – the type of film that is somehow endlessly

Spiderman, Far From Home

Peter Parker and his friends go on a summer trip to Europe. However, they will hardly be able to rest - Peter will have to agree to help Nick Fury uncover the mystery of creatures that cause natural disasters and destruction throughout the continent.

The **client side** was built with React. The goal was to develop a single-page, responsive application that uses routing, different interactions and has several interface views.

React is a JavaScript-based library that does not have the ability to make HTTP requests. This is a technical requirement, in that we want to allow the users to make GET, POST, PUT, and DELETE REST API calls in React App.

For this reason I imported a third party library Axios in my React Application. Axios makes it easy to send asynchronous HTTP requests to REST endpoints and perform CRUD operations. So the solution was very simple by implementing this as a dependency.

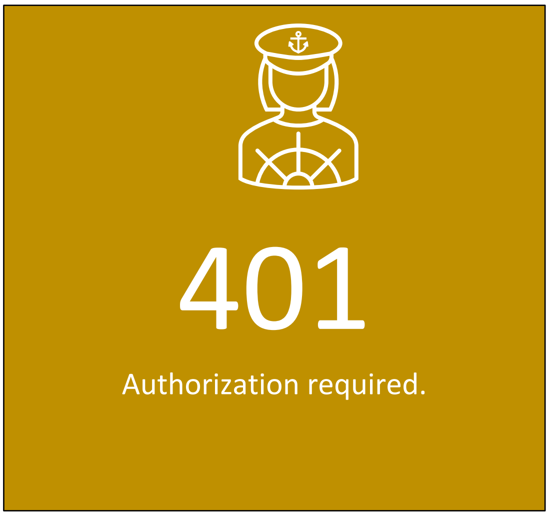
see https://github.com/Joolanda/MyFlix-Movie-app for more on Features and Dependencies.

Challenges

Developing the MyFlix Movie app has been an enriching learning experience in my web development education. Some of the features I implemented required me to gain a deeper understanding of React and Redux.

Overcoming Obstacles
A small problem encountered in development was the update profile feature. I was keep getting a 401 unauthorized response from the API request, and this is because I was not setting the Authorization header correctly. It's a nice example of how Axios makes a PUT call to React App.

Image 6: Example of the 401 Error



401

Authorization required.

Make experience with MyFlix Movie app here.
You can make an account, and try it out!

You will not receive any emails from me and your information will not be used in anyway.