# myFlix

Case Study

## Project Overview

### Description

Using the tech stack MERN, I built the server-side and client-side of this project from scratch. This app allows users to create a profile and displays details about movies, genres, and directors. Users can add movies to their favorites list and remove them, as well as view their information and delete their profile.

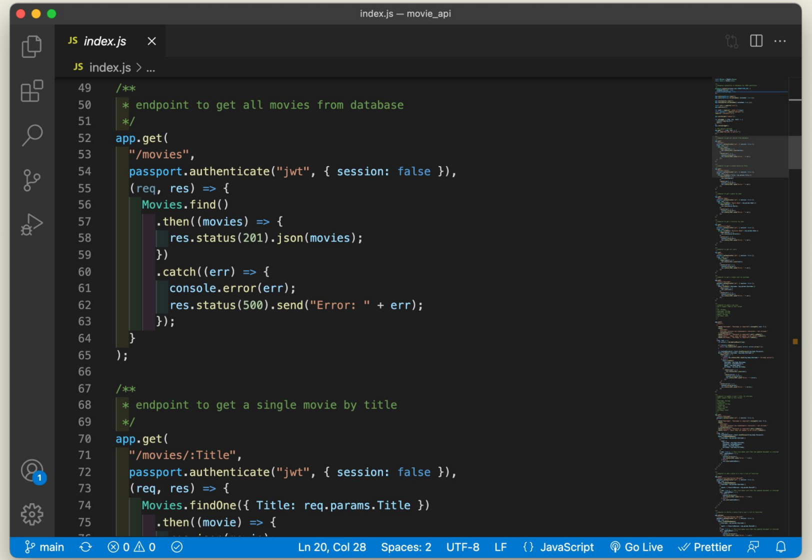
### Context

myFlix was a personal project I built as part of my full-stack web development course at CareerFoundry.

### Objective

This project allowed me to demonstrate my ability to build a full-stack project with MERN and learn the intricacies involved with creating my own API and database.

# Server-Side Approach



```
/**
 * endpoint to get all movies from database
 */
app.get(
  "/movies",
  passport.authenticate("jwt", { session: false }),
  (req, res) => {
    Movies.find()
      .then((movies) => {
        res.status(201).json(movies);
      })
      .catch((err) => {
        console.error(err);
        res.status(500).send("Error: " + err);
      });
  }
);

/**
 * endpoint to get a single movie by title
 */
app.get(
  "/movies/:Title",
  passport.authenticate("jwt", { session: false }),
  (req, res) => {
    Movies.findOne({ Title: req.params.Title })
      .then((movie) => {
```

I created the backend for this full-stack project using Node.js and Express.

I built the REST API that interacts with a non relational database, MongoDB.

The movie and user data stored in this REST API are accessed via HTTP request methods: POST, GET, PUT, DELETE, and translates to CRUD operations, respectively.

The API provides movie and user data in a JSON object, and these endings were tested using Postman.

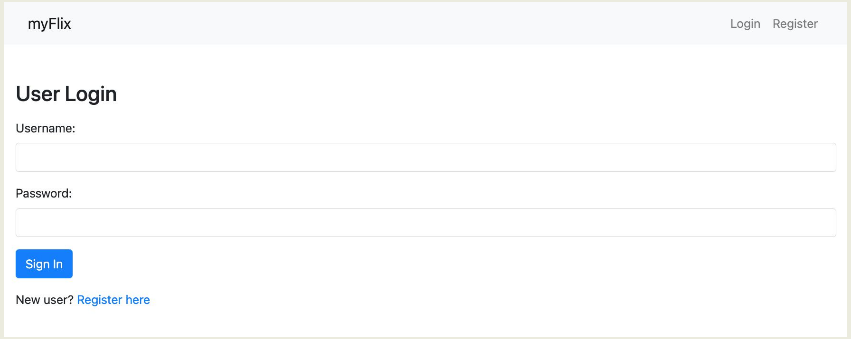
The business logic layer was modeled using Mongoose.

I implemented basic HTTP authentication and JWT/Token-based authentication/authorization using Passport.

# Client-Side Approach

I built the frontend of myFlix using React and React-Redux. This responsive, SPA  is the interface used to make requests to and receive responses from the backend through the previously defined REST API endpoints.

The main view includes the following sub views: a login, registration, a single movie view, genre view, director view, and profile view, each styled with Bootstrap. myFlix allows users to add and remove movies to and from their favorites list, and current users have the option to update their information and delete their profile.



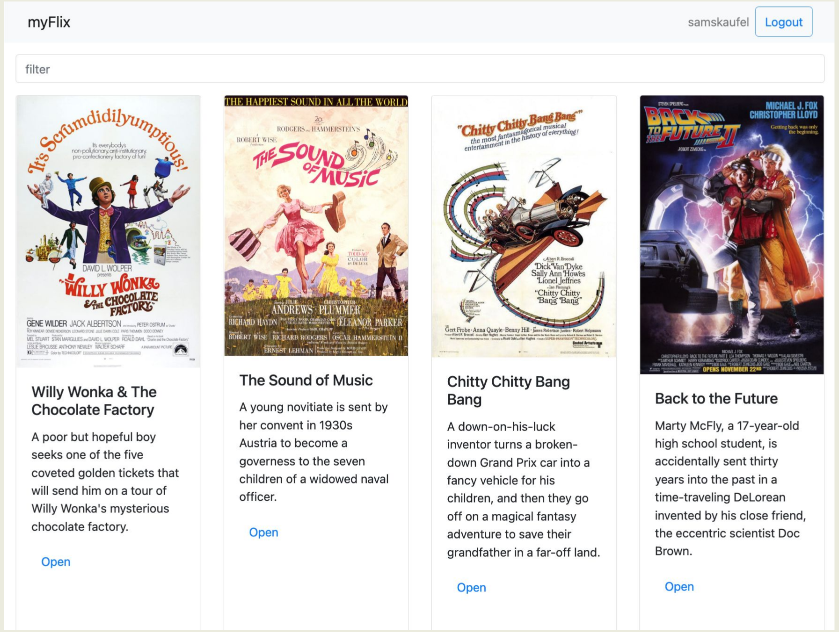
User Login

Username:

Password:

Sign In

New user? Register here



**Willy Wonka & The Chocolate Factory**

A poor but hopeful boy seeks one of the five coveted golden tickets that will send him on a tour of Willy Wonka's mysterious chocolate factory.

Open

**The Sound of Music**

A young novitiate is sent by her convent in 1930s Austria to become a governess to the seven children of a widowed naval officer.

Open

**Chitty Chitty Bang Bang**

A down-on-his-luck inventor turns a broken-down Grand Prix car into a fancy vehicle for his children, and then they go off on a magical fantasy adventure to save their grandfather in a far-off land.

Open

**Back to the Future**

Marty McFly, a 17-year-old high school student, is accidentally sent thirty years into the past in a time-traveling DeLorean invented by his close friend, the eccentric scientist Doc Brown.

Open

## Project Debrief

It was very rewarding to complete my first full-stack project from start to finish.

My main challenges were with implementing React-Redux into my main view and profile view of my application which required the user and movie data currently being stored using state and localStorage, to be stored in the Redux store.

What I would do differently next time is include more comments in my code, and change the “user” object to something else like “userData” because it was easy to mix up the “user” object with the “username”, as well as keep to strict camelCase conventions to help with consistency.

Next steps: I would like to work on the UI design of my application, and add another view with leading actors/actresses.

# Project Repositories

Server-side

Client-side

## Credits

Role:

Lead Developer

Mentor:

Vinicius De Antoni

Tutor:

Francis Kim