

# MyFlix

## A case Study

Bryant Ortega for CareerFoundry





# Overview

MyFLix is a CRUD web application that gives users access to movie details like the directors and plot summary. Users have the option to register for an account, edit their personal data, store a list of their favorite movies, and remove their account.

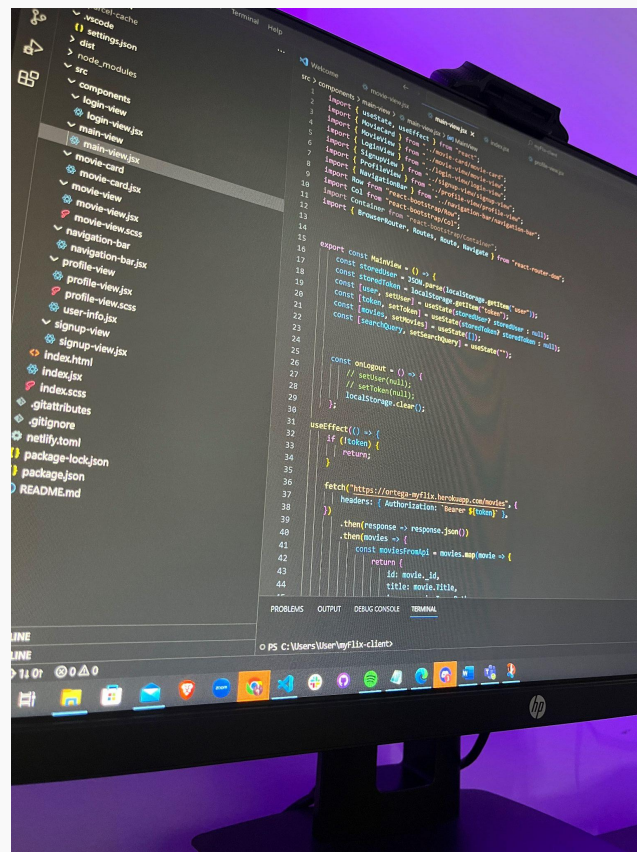


# Purpose and Context:

MyFlix is a personal project I built as part of my web development course at CareerFoundry to demonstrate my mastery of full-stack JavaScript development.

## Objective:

The aim of the project was to have an ambitious full-stack project for my professional portfolio. The problem I wanted to solve is to build the complete server-side and client-side for the application from scratch.





## Duration

70 hours, over 5 weeks, to familiarize myself with MERN stack and complete the app. It took me longer than I would have liked due to simultaneously working part-time and raising two children.



## Credits

Developer: Bryant Ortega

Tutor: Jorge Armando Cortes Montiel

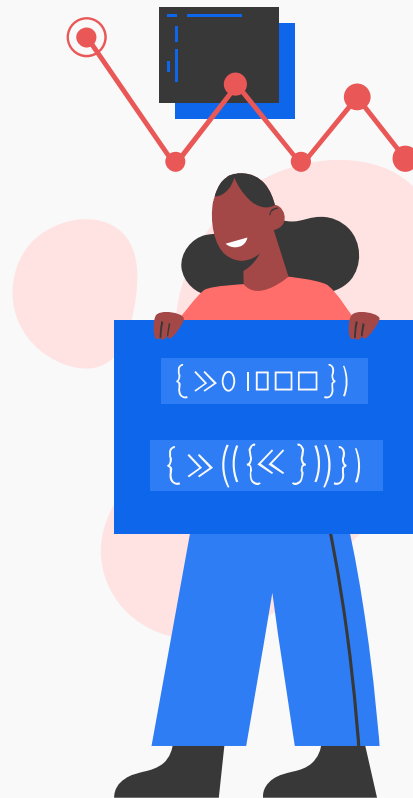
Mentor: Sebastian Kolman

Case Study Template: [Slidesgo](#)



## Tools & Methodologies

MongoDB, Express.js, React, Bootstrap, Node.js, Heroku, Postman, Github, JavaScript



# Server Side

I built a REST API that uses CRUD methods to manage movie data using JSON format.

## Developer environment

Prepared with **Node.JS** and **Express**.

## Business logic

I configured my schema and data models with **Mongoose** for uniformity.

## Authentication & Authorization

I implemented basic HTTP authentication for initial login requests, JWT authentication for future requests **CORS** password hashing for added security.

## Deployment

- Endpoints tested with **Postman**
- **MongoDB** used to host my database
- Deployed my api with **Heroku**.



# Client Side

I designed a clean uniform interface with React so users could easily manipulate the database.

{ }

## Architecture

Component based for organization. I decided not to implement **Redux** as I do not have plans to greatly scale MyFlix.

## App routing

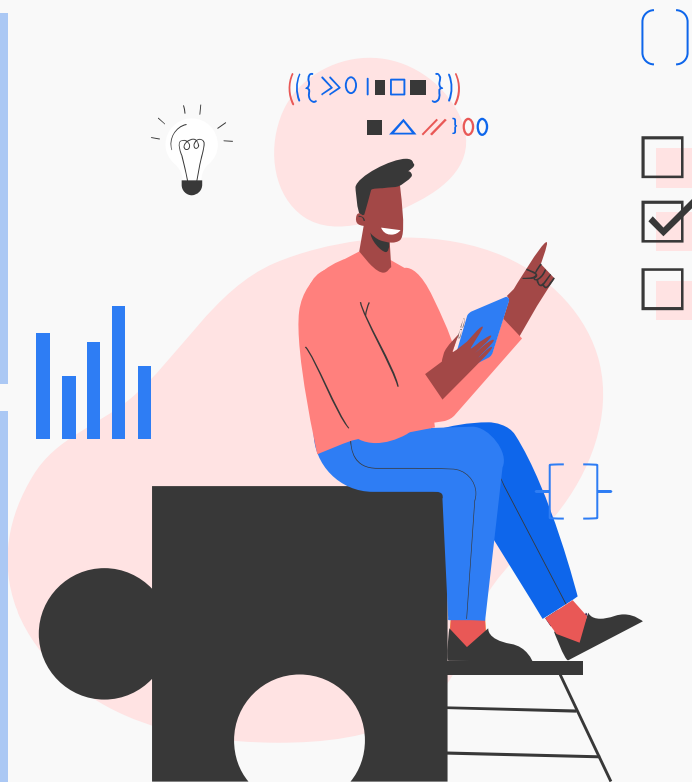
I implement state-based routing to route views in a **React** SPA for better search engine optimization.

## Design

I used **React Bootstrap** to quickly create a clean uniform aesthetic by using forms, buttons, and cards.


## Deployment

I used **Netlify** to host MyFlix's front end to familiarize myself with an alternative to Heroku.

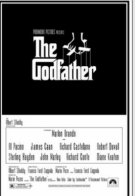




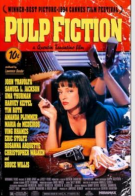
MyFlix
Movies
Profile
Logout




**Snatch**  
Guy Ritchie



**The Godfather**  
Francis Ford Coppola



**Pulp Fiction**  
Quentin Tarantino



**The Lord of the Rings: The Fellowship of the Ring**  
Peter Jackson




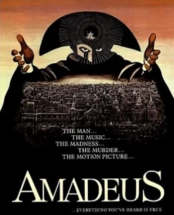




MyFlix
Movies
Profile
Logout


**Profile**  
Name: finaluser10  
Email: testuser@fakemail.com  
Birthday: Jul 06, 1987  
[Delete account](#)

**Update your profile information.**  
Username:  
  
Password:  
  
Email:  
  
Birthday:

#### Favorite movies:

MyFlix
Movies
Profile
Logout



Title: The Lord of the Rings: The Two Towers

Genre: Fantasy Fiction

Director: Peter Jackson

Description: The surviving members of the Fellowship have split into three groups. Frodo and Sam face many perils on their continuing quest to save Middle-earth by destroying the One Ring in the fires of Mount Doom. Merry and Pippin escape from the Orcs and must convince the Ents to join the battle against evil.

MyFlix Login Signup

**Login:**  
Username:  
  
Password:

[I don't](#)  
[have an](#)  
[account.](#)

# Page Views

Once a user logs in, they are able to search movies, get more details, add/remove favorites, and view and update their personal information.



# Summary & Final Reflections



I was able to complete the application to demonstrate that I could build completely from scratch both the client side and server side.



The most challenging part of the project was understanding the workflow amongst the various working components.



Once I got to the authentication, it all clicked for me and I am now confident in my understanding of building both server and client sides of a web application.



Now that I understand how a React application is built, I will make use of CRA so that I can make more efficient use of time.