

MYFLIX



CASE STUDY

ALICJA NOTOWSKA

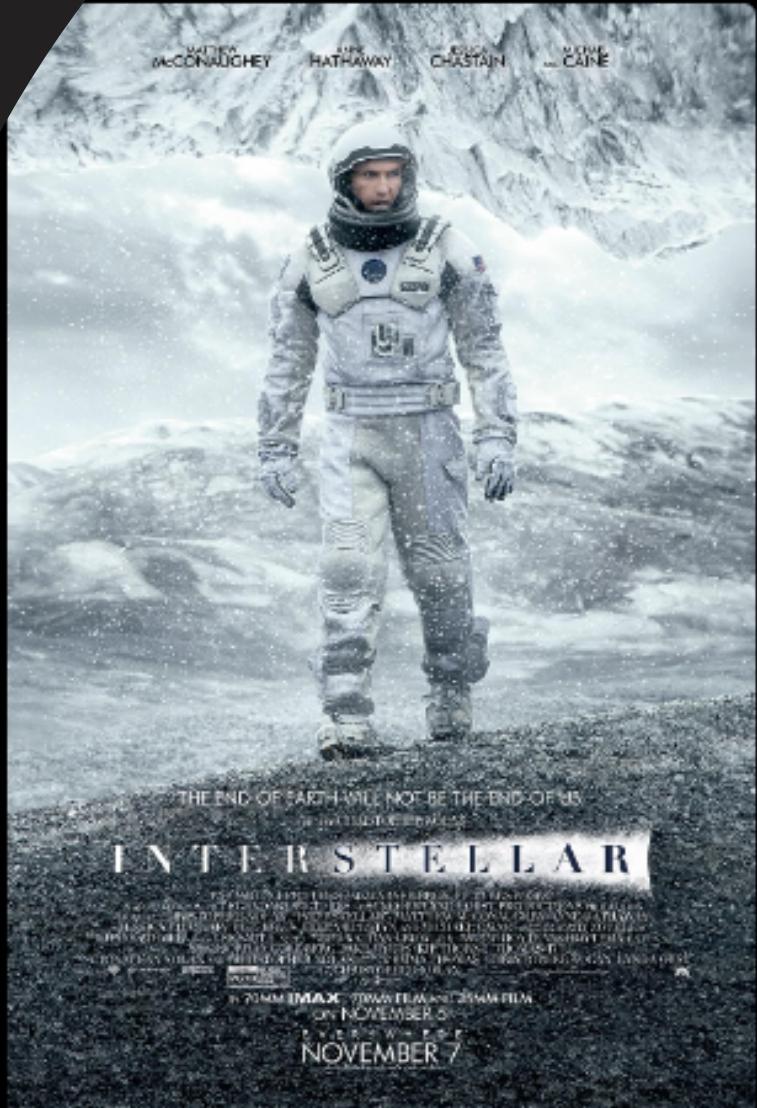
 [My Portfolio Website](#)

 [Look up the code on Github](#)



OVERVIEW

MyFlix is a web app, developed using the MERN stack, that provides users with access to information about movies, directors, and genres. Users are able to create an account, update their personal data, and create a list of favourite movies.



Interstellar

Christopher Nolan

Open



The Silence of the Lambs

Jonathan Demme

Open

PURPOSE AND CONTEXT

Purpose

The purpose was to learn and apply gained skills to demonstrate my mastery of full-stack JavaScript development.

Context

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

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

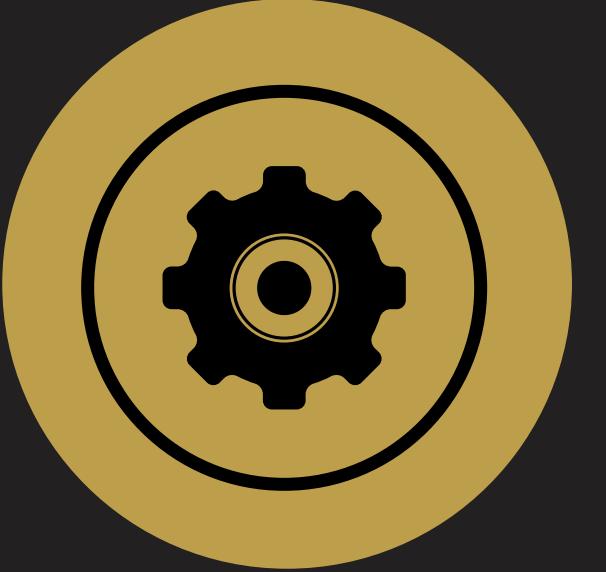
OBJECTIVE





Duration

It took about a month to build the API from scratch and two months to create the front-end with React.



Tools & Methods

- React
- Redux
- React Bootstrap
- MongoDB
- Express
- Heroku
- Node.js
- Postman
- Parcel



Credits

I was the developer on this project. Of course, I had help of my tutor and mentor.



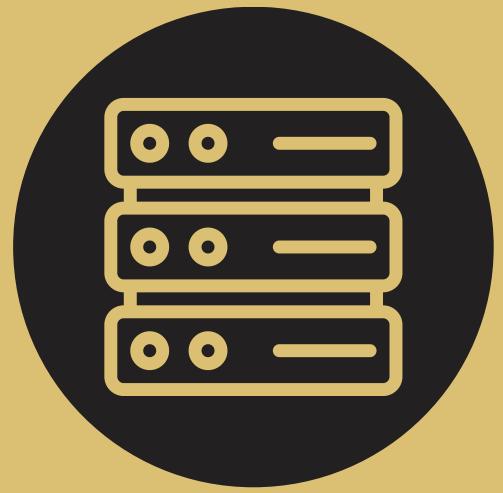
Server-side

I created a RESTful API from scratch using Node.js and Express. The API interacts with a non-relational database and it can be accessed via commonly used HTTP methods like GET or POST. CRUD methods are used for data retrieval and storage. The API provides movie information in JSON format.

To test the API, I used Postman. I also included user authentication and authorization code in the form of basic HTTP authentication and JWT authentication.

API is deployed to a cloud platform (Heroku), while the database is hosted on MongoDB Atlas.

ARCHITECTURE



Web Server

Express was used to route requests/responses and interact with request data. I defined and tested the API endpoints and their corresponding HTTP methods.



Business Logic

I used Mongoose to configure schema and models for interactions with the database.



Data Layer

I chose MongoDB to build the database that stores movie and user information.

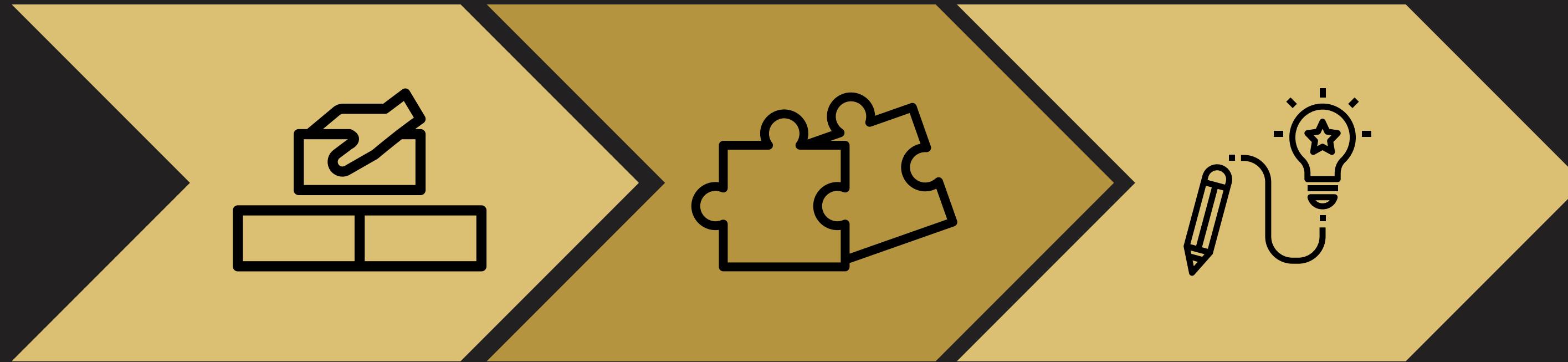
Client-side

After completing the API, I started to build the interface users would need when making requests to, and receiving responses from, the server-side. It is a single-page, responsive application, developed with React and React-Redux. It provides several interface views, including, but not limited to, a main view (shows a list of all movies), single movie view (shows data about a single movie and allows users to add the movie to their list of favourites), a login view, a registration view and a profile view (where users can update their user data and list of favourites).

Once completed, I hosted the app on Netlify.



FRONT-END DEVELOPMENT



The Build

One goal of this project was to understand how to use MVC architecture as a design pattern. To facilitate this, I used React and then, Parcel to complete the build operations.

Components

Once the initial build was done, I created components for the different views and hooks to control the state.

Design

I used React Bootstrap to design the layout of the pages and cards and to ensure consistent styling. I also used client side app-routing to add authentication to access views.

FINAL PRODUCT

MyFlix Home My Profile Log Out

Username : BumbleBee
Email: bumbleb@bmail.com
Date of birth: 1985-05-05T00:00:00.000Z

Update Your Profile

Username:

Email:

Birthday:

Password:

[Update](#)

[Delete Your Account](#)

Your Favourite Movies



Interstellar

MyFlix Home My Profile Log Out

Search by title

Search



Interstellar
Christopher Nolan
[Open](#)

Silence of the Lambs
Jonathan Demme
[Open](#)

The Dark Knight
Christopher Nolan
[Open](#)

The Shawshank Redemption
Frank Darabont
[Open](#)

Challenges

I got stuck a few times during the front-end development process.

Firstly, during the build process I encountered an error with Parcel that took me down the rabbit hole of StackOverflow answers, and none of which provided the solution.

Secondly, it took me a while to understand client-side routing. But with the help of both, my tutor and a mentor session I was able to detangle the knots.



Get In Touch

 [My Website](#)

 alicjaa.in.wonderland@gmail.com

 Berlin, Germany

