



# Case Study: myFlix App

## Overview

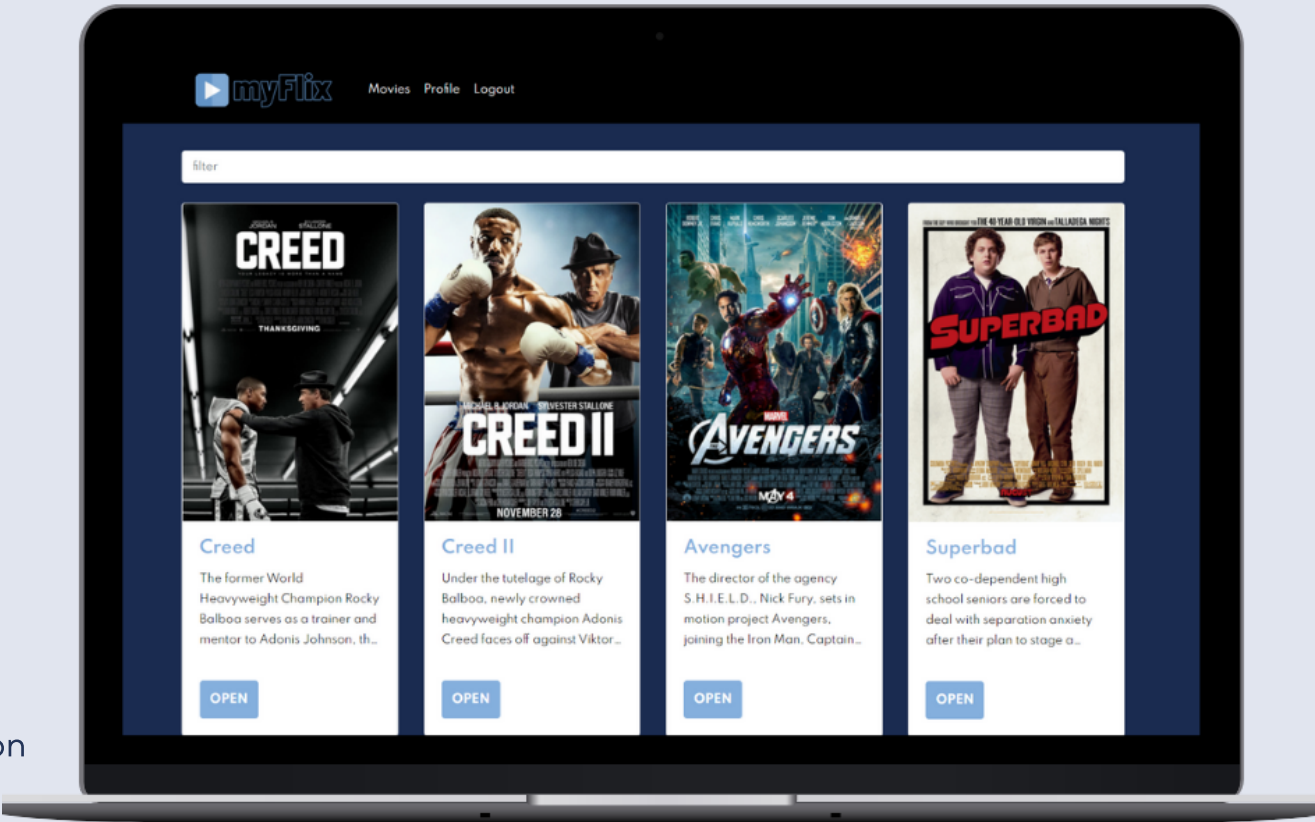
myFlix is a web application that provides users with access to information about different movies, genres, and directors. Users have the ability to create their own account, update their user information, and personalize their experience by creating a list of favorite movies.

## Process & Context

myFlix was developed using the MERN stack, starting with the development of the server-side application and then completing the project by creating the client-side application. myFlix was a personal project built as part of the curriculum through CareerFoundry, a full stack web development program.

## Objective

The goal of the myFlix application was to further develop my front and backend skills, as well as have a MERN project for my professional portfolio. I also had a personal goal of improving my design skills and implementing personal touches outside of the CareerFoundry program requirements. The problem I wanted to solve was building such an ambitious project from the ground up, including building everything for React from scratch.



## Approach

### The Server-Side

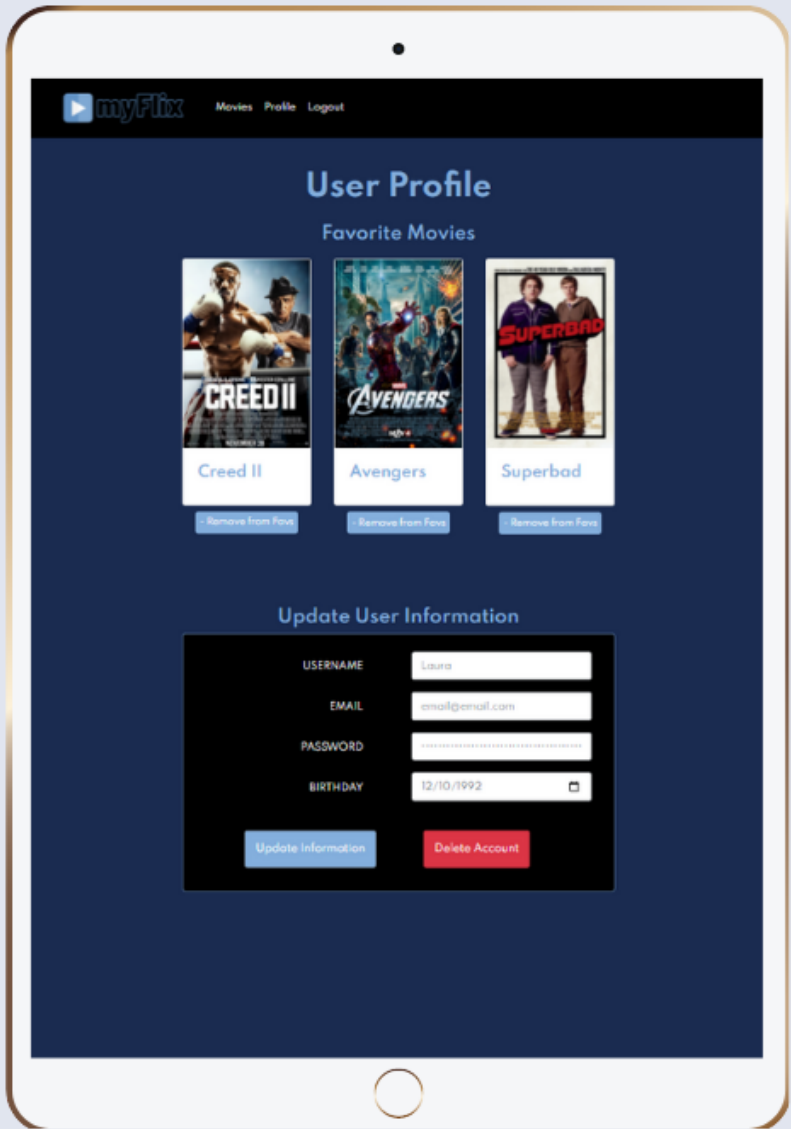
I created a server-side application that utilizes Node.js, Express, and RESTful API architecture. First I set up the project and it’s dependencies, then routing the HTTP requests using Express and defining the URL endpoints. The next step was creating a NoSQL Database using MongoDB. To retrieve data from and store data in the database, CRUD methods are used.

[View endpoint documentation here](#)

Throughout the development process, the API was tested through Postman. The final steps were incorporating data validation logic into my API in the form of basic HTTP and JWT authentication, and then hosting the project through Heroku.

### The Client-Side

Once the server-side of the application was created, it was time to develop the client-side, where users can make requests to and receive responses from the server-side. The SPA (single-page application) was made using React and Redux. This allowed for the implementation of multiple views, including the Registration & Login Views, Main View, Single Movie View, and Profile View, each providing essential features to the app.



## Challenges

Being a new developer, this project was a huge feat! It provided many learning and problem-solving opportunities, and overall made me a better developer. In particular, I found creating all of the features of the Profile View to be the most challenging and time-consuming piece of the project as a whole. It was a fun challenge determining the best way and place to implement each essential feature and learn how to. I found the most challenging part of the server-side process to be implementing and fully understanding the JWT tokens.