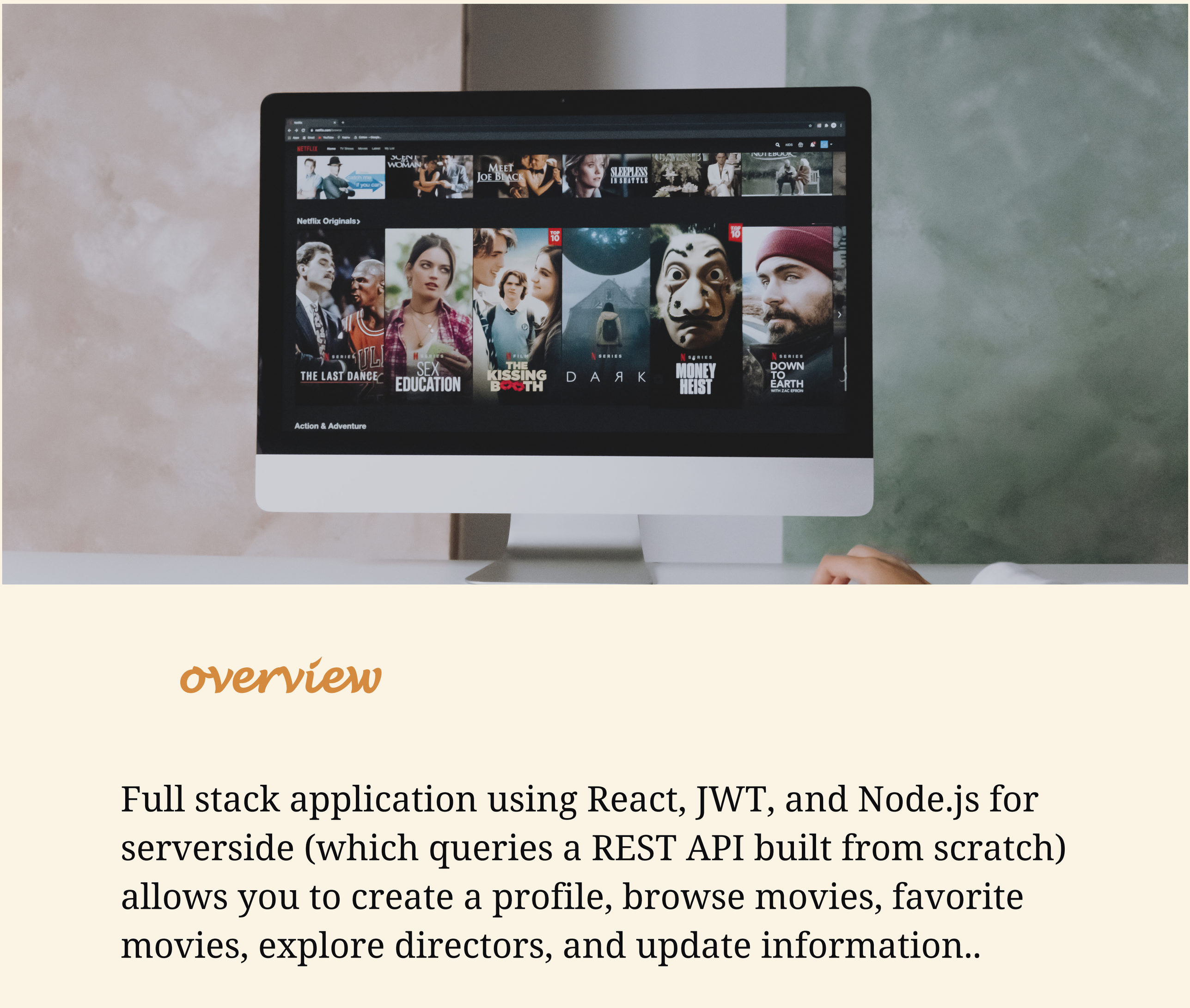
ReactJS Web App - MyFlix

CASE STUDY



overview

Full stack application using React, JWT, and Node.js for serverside (which queries a REST API built from scratch) allows you to create a profile, browse movies, favorite movies, explore directors, and update information..

Purpose and Context

myFlix is a portfolio project built as part the Full-Stack Immersion course at CareerFoundry. This two part project covered front and backend development

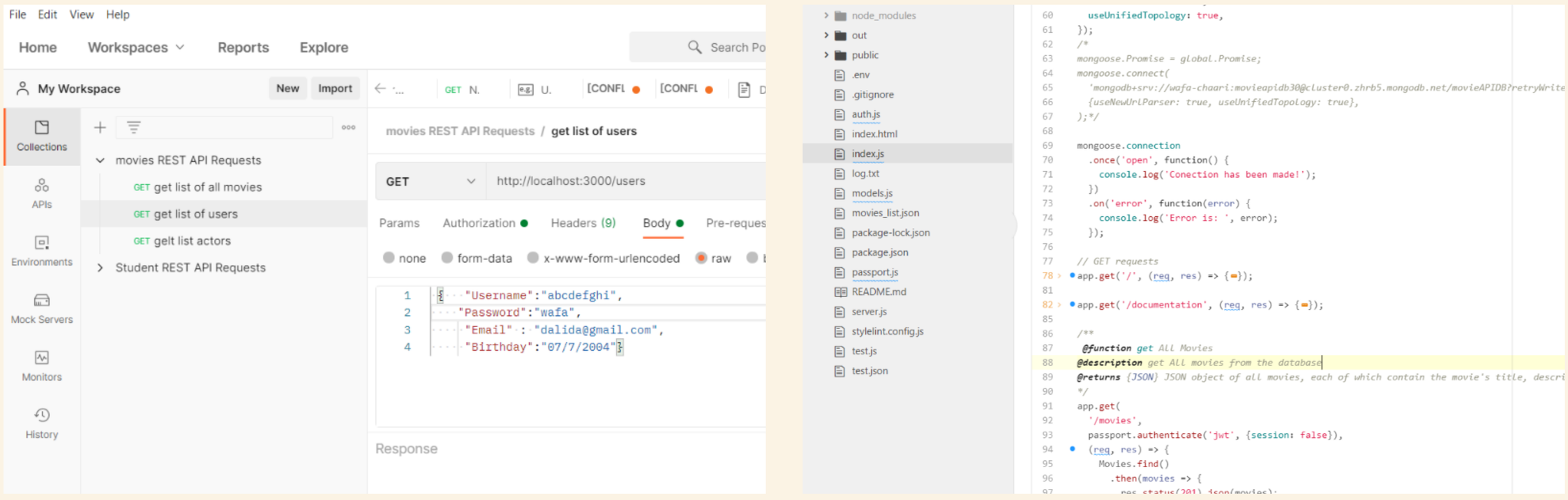
Objective

The web application will provide users with access to information about different movies, directors, and genres. Users will be able to sign up, update their personal information, and create a list of their favorite movies.

Approach

Server-Side

Throughout this Achievement, I've added an entirely new repertoire to my bank of coding knowledge, learning how to go beyond the frontend, or client-side, of your application to create databases and server-side scripts in the backend. I now have a complete REST API I can call my own, along with a non-relational database in which to store the data for my app.



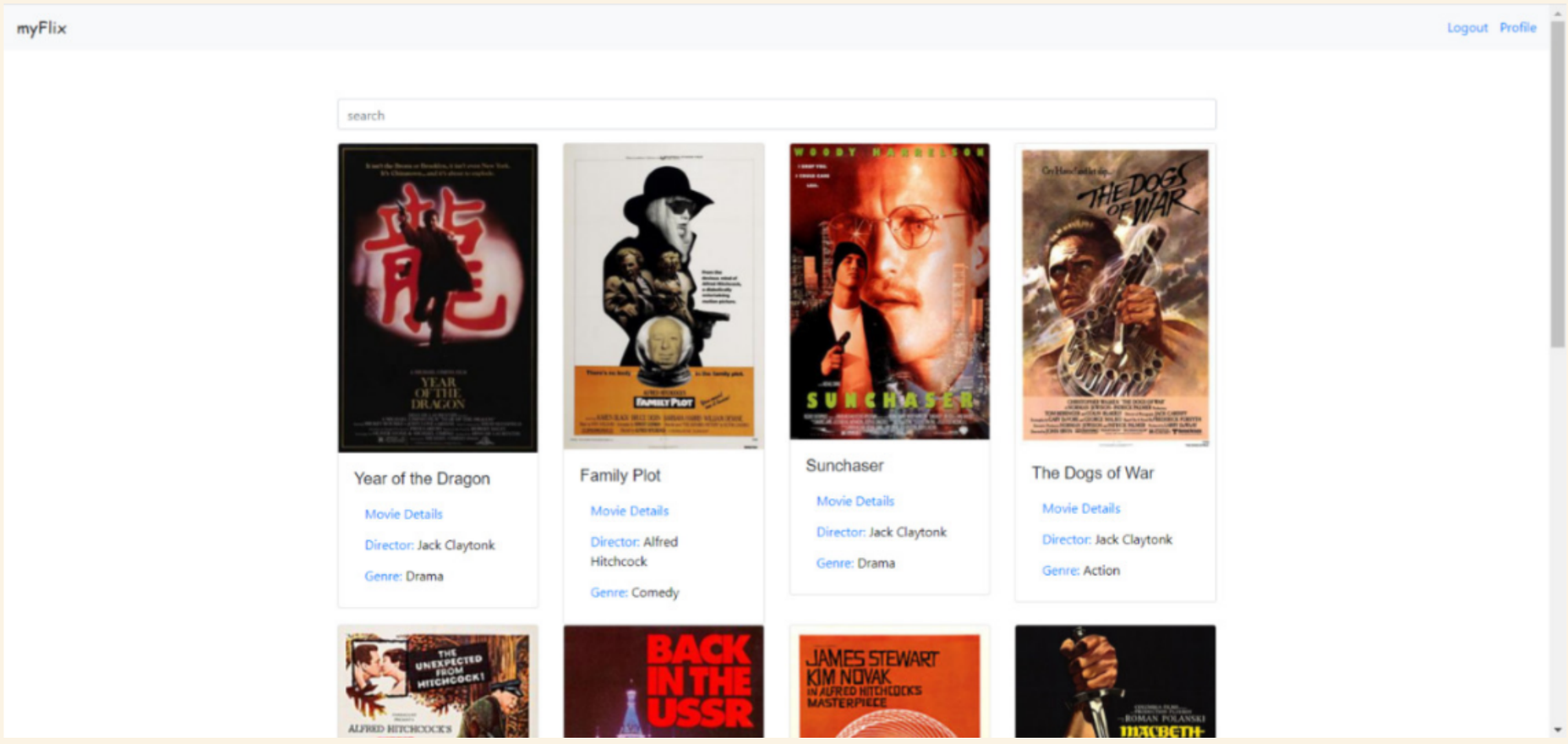
- At first I created a RESTful API by using NodeJS and Express.
- Then, I created a relational (SQL) database as a non-relational (NoSQL) database using MongoDB
- The REST API will be accessed via commonly used HTTP methods like GET and POST. Similar methods (CRUD) was used to retrieve data from the database and store that data in a non-relational way
- In order to test the API, I used Postman

Client-Side

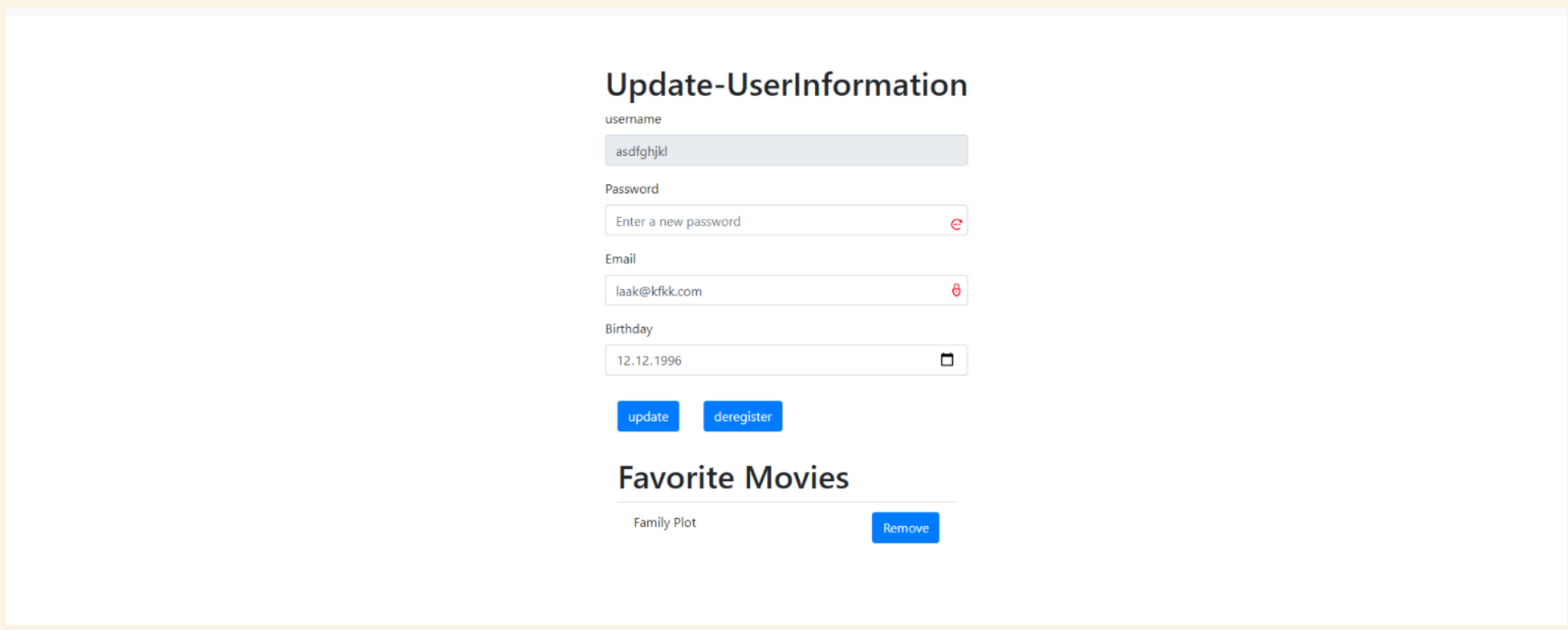
After building the server-side ,I moved from the back to the front, and build the client-side of my MyFlix app. First, I developed the interface views built using the React library that users will interact with to make their requests, such as searching for movies and updating their details. Responses from the server will be rendered into the interface: movie synopses, director biographies, genres, and more.

Login View

Registration View



Profile View



Profile View

- The application used state routing to navigate between views and share URLs
- The application used Parcel as its build tool
- The application must be written with React Redux
- The application must use Bootstrap as a UI library for styling and responsiveness
- The application must contain a mix of class components and function components

Challenges

This was my most challenging project so far. It took me quite a while, until understanding how React and Redux work.

when I coplieted this project I was quite happy and proud. For the future I am planning to add some more movies and further information about the directors and actors.

credits

Role: Lead Developer
Tutor: Itua Akhator
Mentor: Vinh Tuong Mai