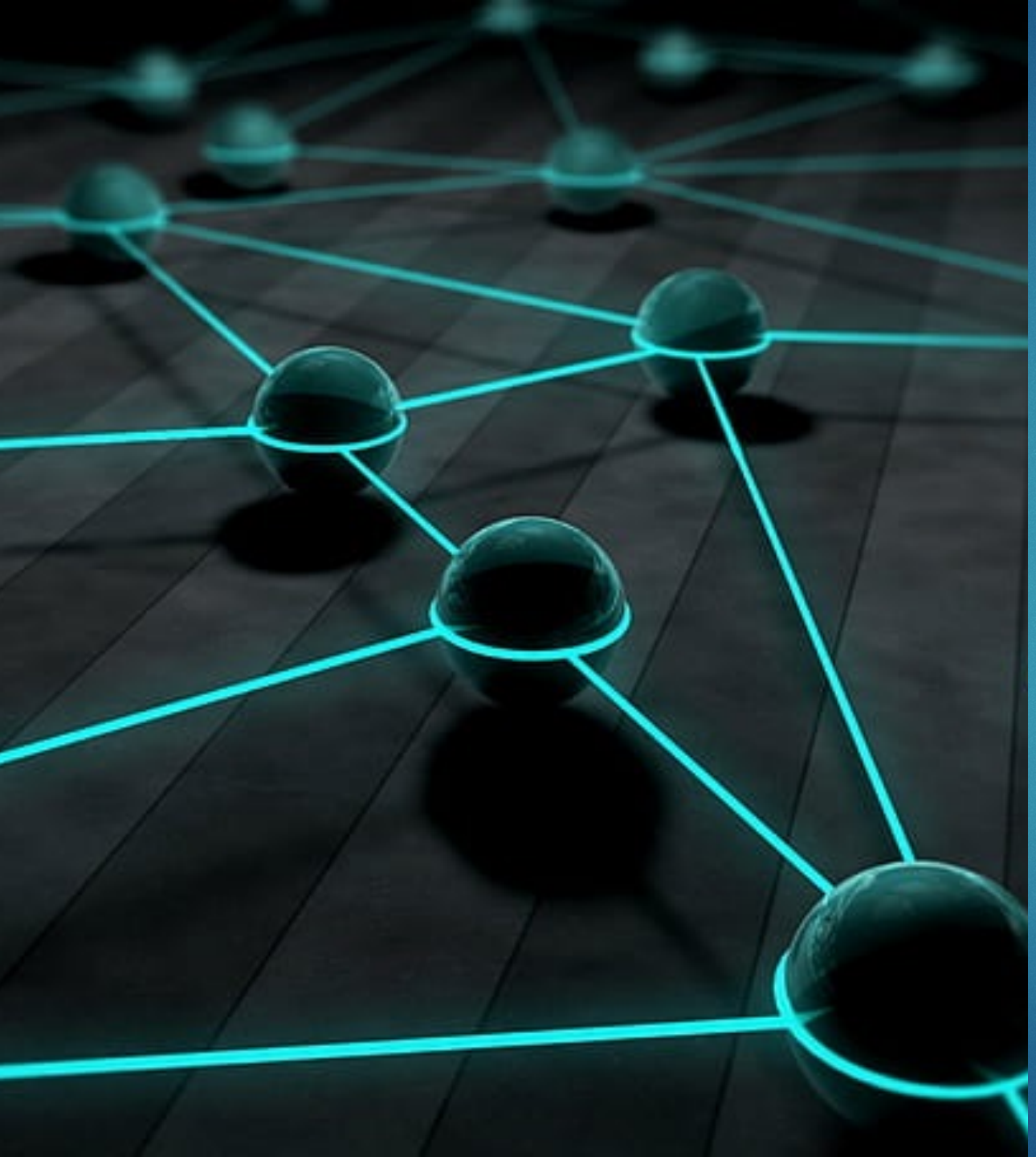


myFlix: Full Stack Development Case Study

SAPNA BOLIKAL

Overview

- This is myFlix app, an app I created using MERN stack. This web app provides information about movies, directors, and genres of movies. Users can create a profile where they can add and remove favorite movies, update user information, and delete their account to their choosing.

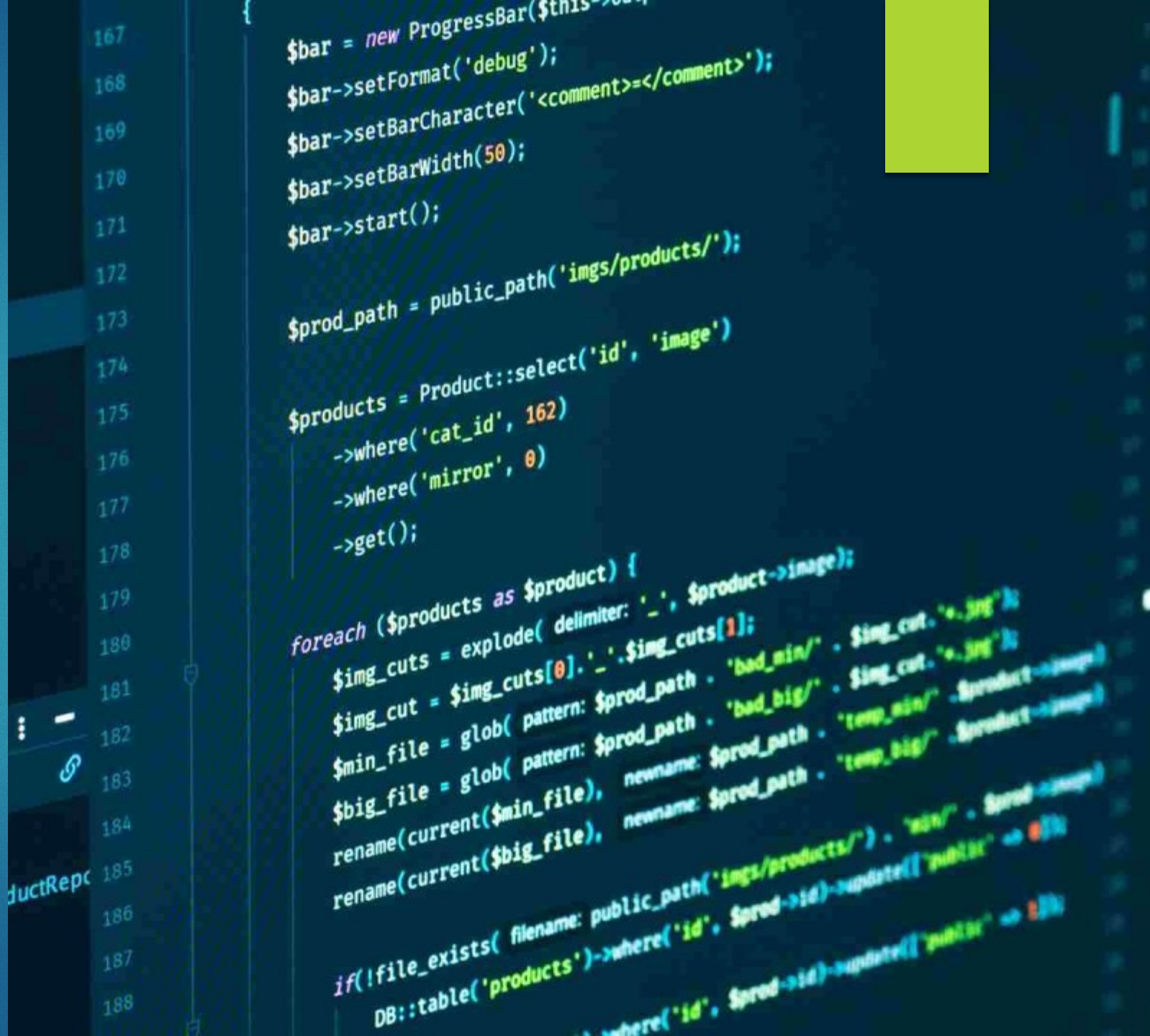
- See my site

Purpose

- The purpose of this app was to display my skills in full-stack JavaScript development.

Objective

The objective of the project was to create a challenging web app in which I built a complete server side and client-side app.



e Logout

Batman Begins

After witnessing his parents death, Bruce learns the art of fighting to confront injustice. When he returns to Gotham as Batman, he must

Bridesmaids

Competition between honor and a bridesm is the brides best frie to upend the life of a pastry chef.

Open

Duration

- The front-end of the app took longer than the back end, due to figuring out the code and integrating the functions

Credits

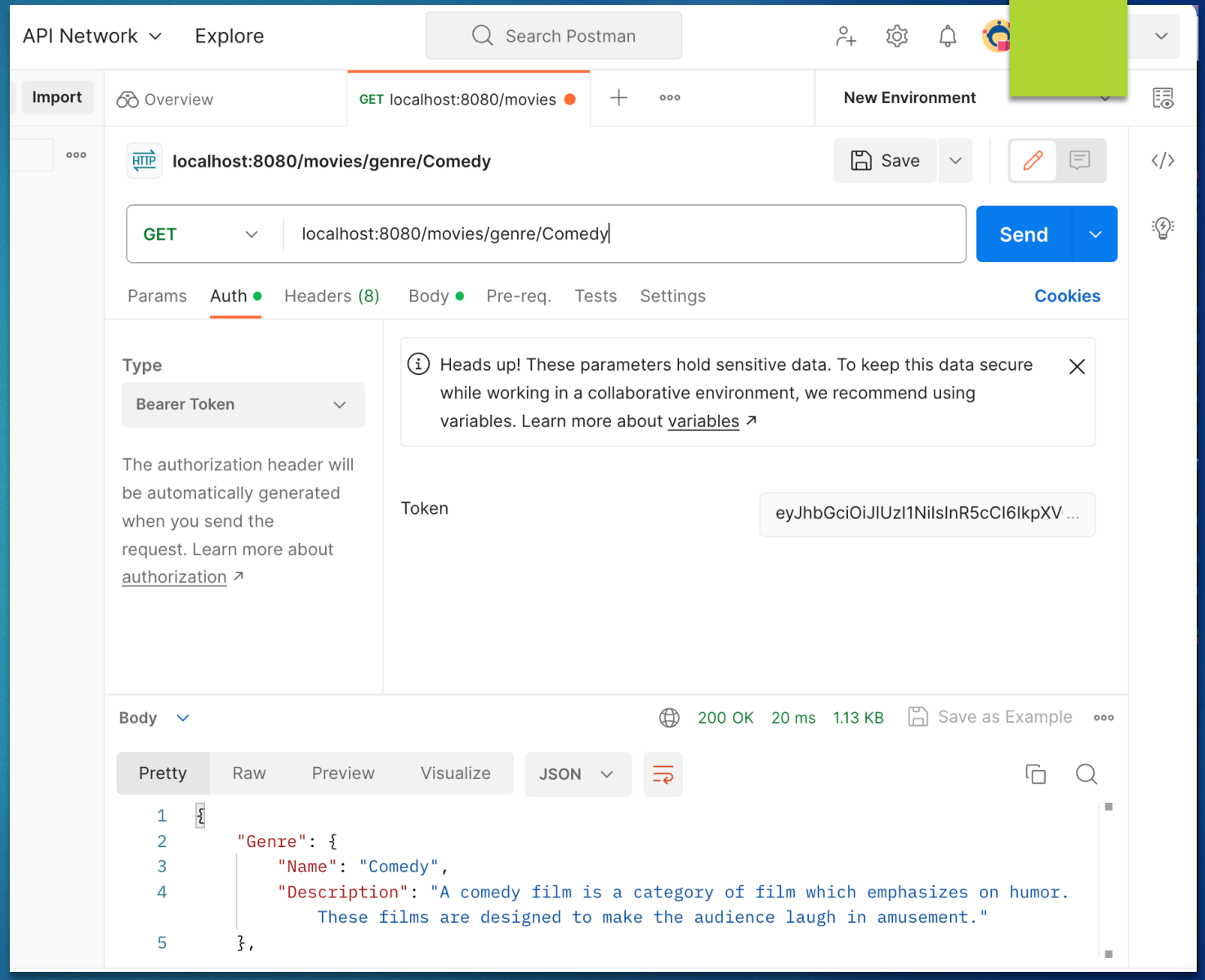
Lead Developer: Sapna Bolikal
Mentor: Akunna Nwosu
Tutor: Moises Serrano

Methodologies

MERN stack Postman
Heroku React Bootstrap

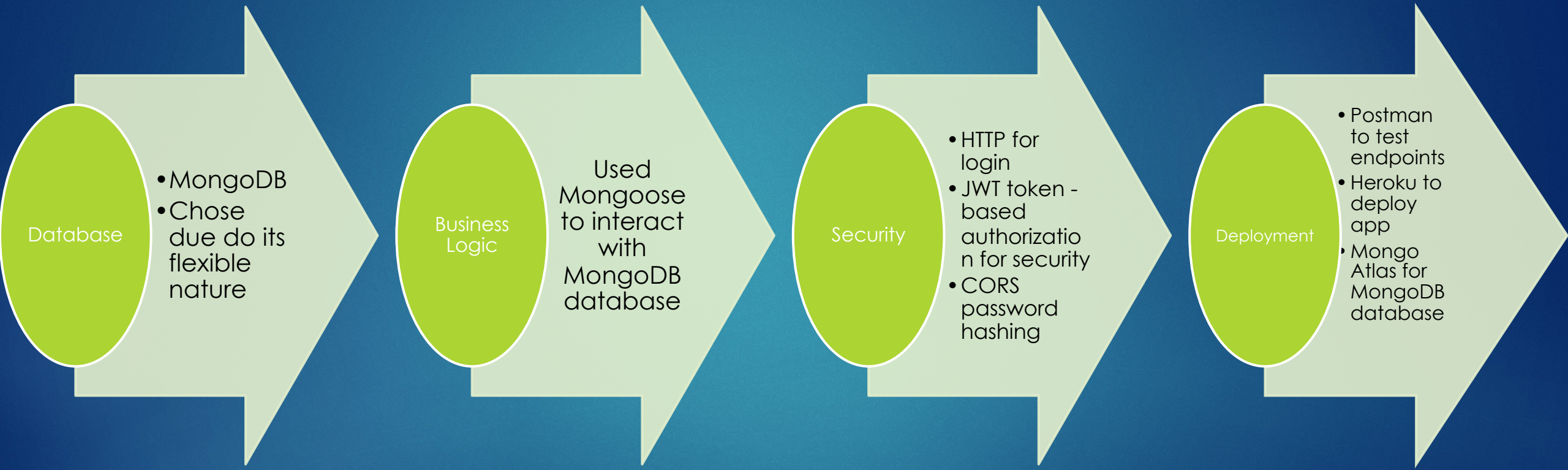
Approach

- Server-Side
- For the server-side I created a RESTful API using Node.js and Express, which interacted with MongoDB, the non-relational database. HTTP methods were used to retrieve the API. CRUD methods were used to store and retrieve data from MongoDB. API provides movie in JSON format.



Tested HTTP endpoints using Postman

Server-Side


Database
•MongoDB
•Chose due do its flexible nature
Business Logic
Used Mongoose to interact with MongoDB database
Security
• HTTP for login
• JWT token - based authorizatio n for security
• CORS password hashing
Deployment
• Postman to test endpoints
• Heroku to deploy app
• Mongo Atlas for MongoDB database

Client Side

USED REACT TO BUILD THE PROJECT BECAUSE IT IS WELL DOCUMENTED AND RELIABLE

I CREATED DIFFERENT COMPONENTS FOR EACH VIEW AND USED FETCH TO PULL THE API

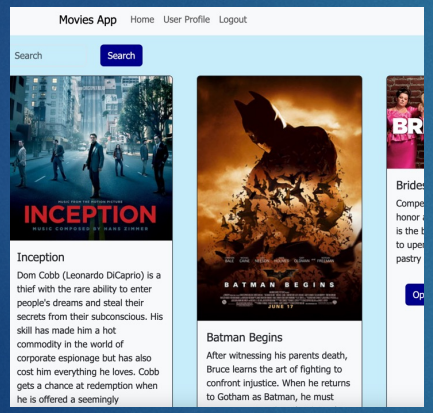
FOR THE DESIGN OF THE APP, I USED REACT BOOTSTRAP. THIS WAS DONE TO ENSURE CONSISTENT STYLING.

I USED REDUX TO MANAGE APPLICATION'S STATE. THEN I HOSTED MY APP ON NETLIFY.

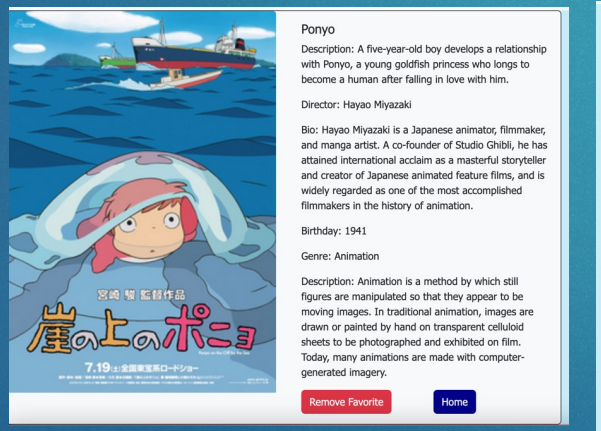
- After building the server-side API, I built the client side through which users would interact with the website. I used React and React-Redux to create a single page, responsive application. There are several views for users including, main, movie, login, registration, and sign up.

Page Views

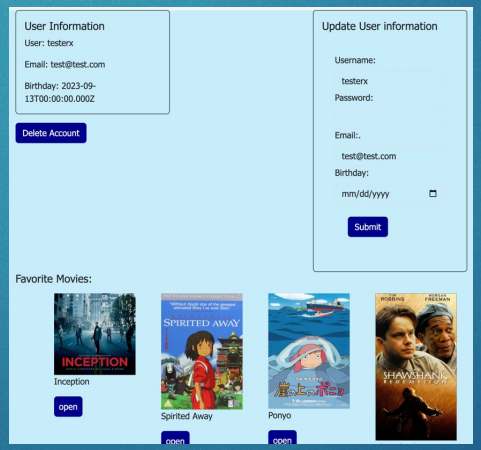
I created several page views including a secure login as well as signup page. A main view where users can see all the movies, when users click “more” they can see more information about the movie in the movie view. They can also add movies to their favorites. There is also a profile view, where users can view and update their information and favorite movies.



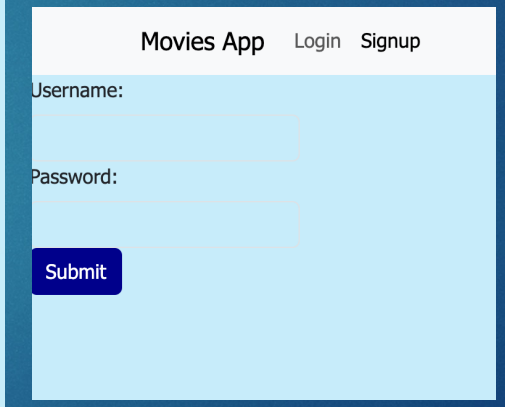
Main View



Movie View



Profile View



Sign-in view

Retrospective

- What went well: Although a bit challenging at times, I enjoyed using React to code my app. Figuring out how to make the site flow was exciting and different.
- What did not well: While exciting, there were a few hangups I encountered while building myFlix app. Specifically, syntax issues that I had to navigate and figuring out the code for Movie View.
- Future steps: I would like to add more movie options for viewers to chose from and to include more features such as displaying similar movies when user is looking at a specific movie
- Final thoughts: I am satisfied with the myFlix app I created. It helped me become more comfortable with React and the MERN stack.