CineDEX

A FULL STACK CASE STUDY

Jamie Tracy

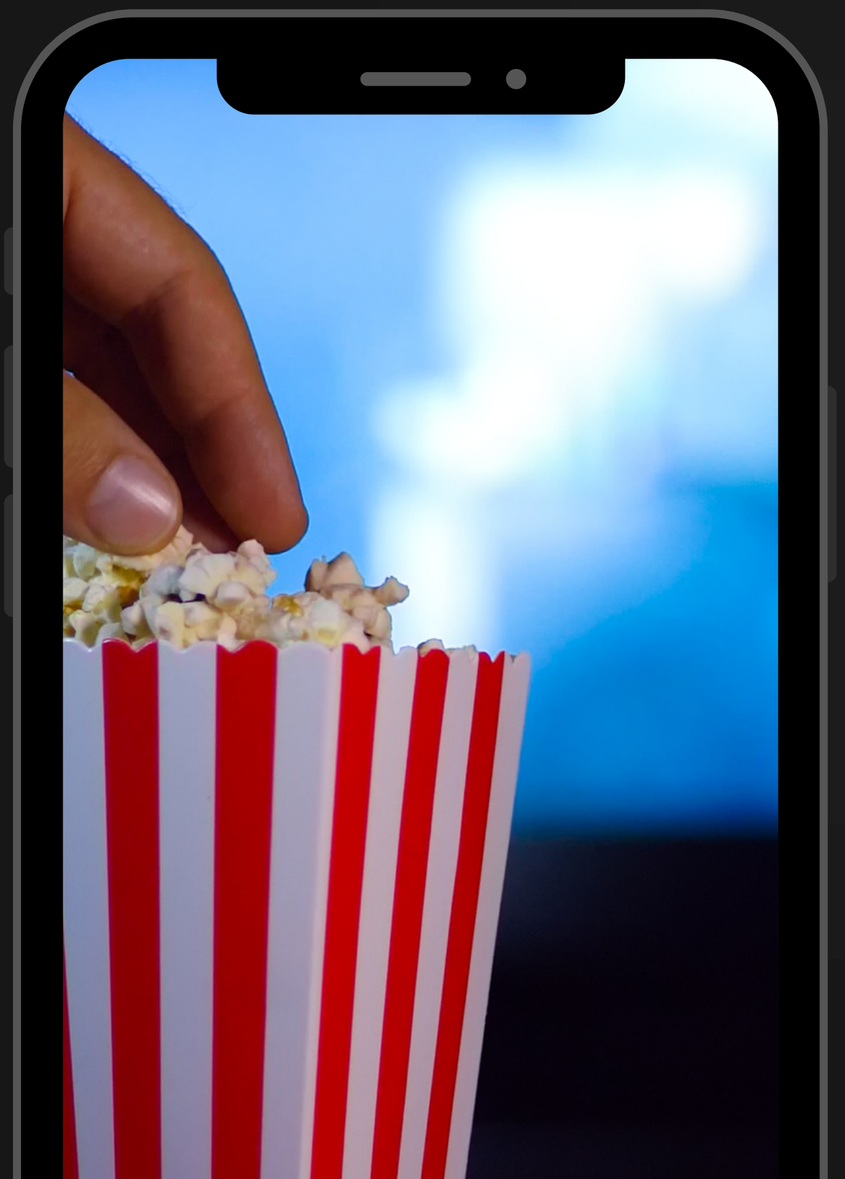
CineDEX

Overview

CineDEX is a single-paged responsive web application developed using the MERN stack and uses routing and multiple interface views to provide a number of features to film lovers. Users can register, login, update their profiles, search for movies by title, genre, or director and even create a favorite movie list where they can add or remove favorites. CineDEX also provides users with information about an assortment of movies, directors, and genres.

Check it out

CineDEX



Objective

This project was part of the the full-stack immersion course from the web development program at CareerFoundry to demonstrate mastery of full-stack JavaScript development. The goal was to create a complete web app (server-side and client-side) to showcase in my portfolio.

CineDEX

Details

Roles

Role: Lead Developer
Tutor: Ebere Iweala
Mentor: Mahesh Rodrigo

Duration

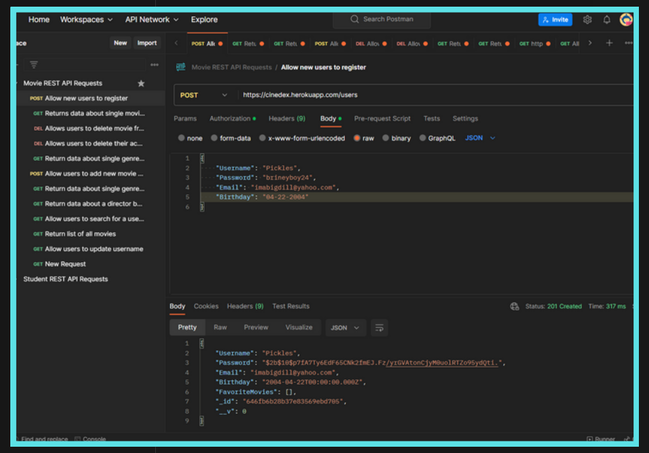
Both server-side and client-side took me approximately 7-8 weeks to complete from start to finish, but client-side took significantly longer than server-side.

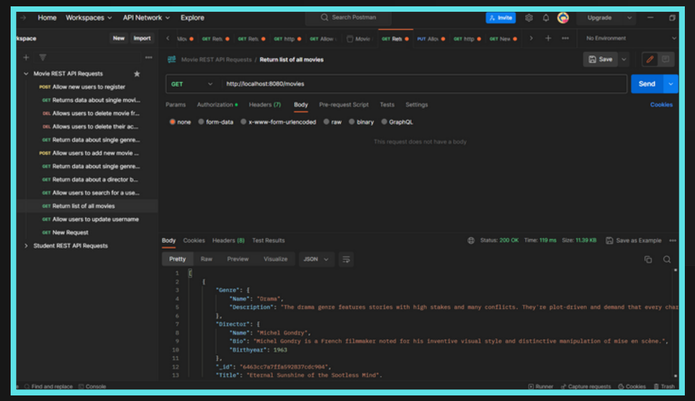
Methodologies

- MERN stack
- Postman
- Heroku
- React Bootstrap
- Redux

CineDEX

Server-Side

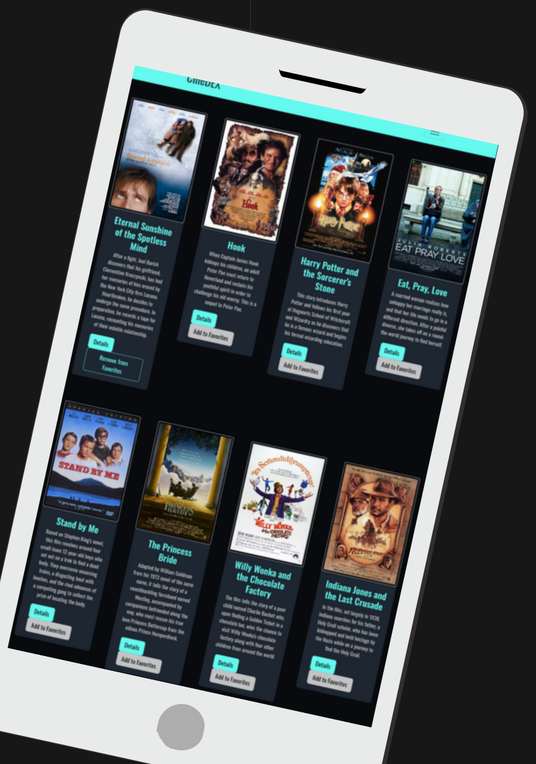




I created a REST API to interact with a non-relational database of movies I made using MongoDB. The API is accessed through HTTP methods, where requests can be sent to set endpoints to retrieve specific data from the database. I also implemented basic HTTP and JWT token-based authentication as well as password hashing and data validation. I was able to test the functionality of these endpoints using Postman during development. Last, I used MongoDB Atlas to host my database and Heroku to deploy my app.

CineDEX

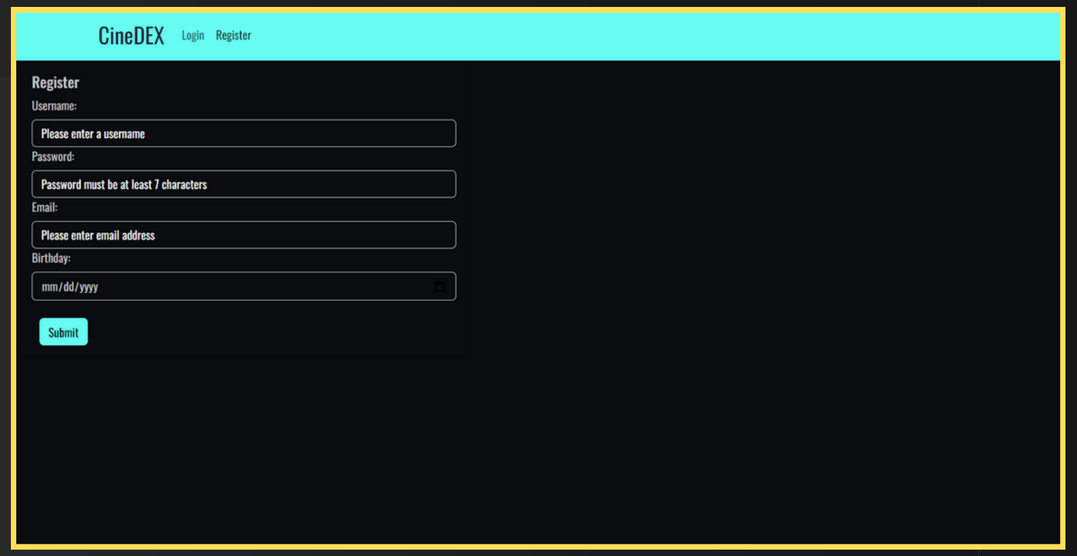
Client-Side



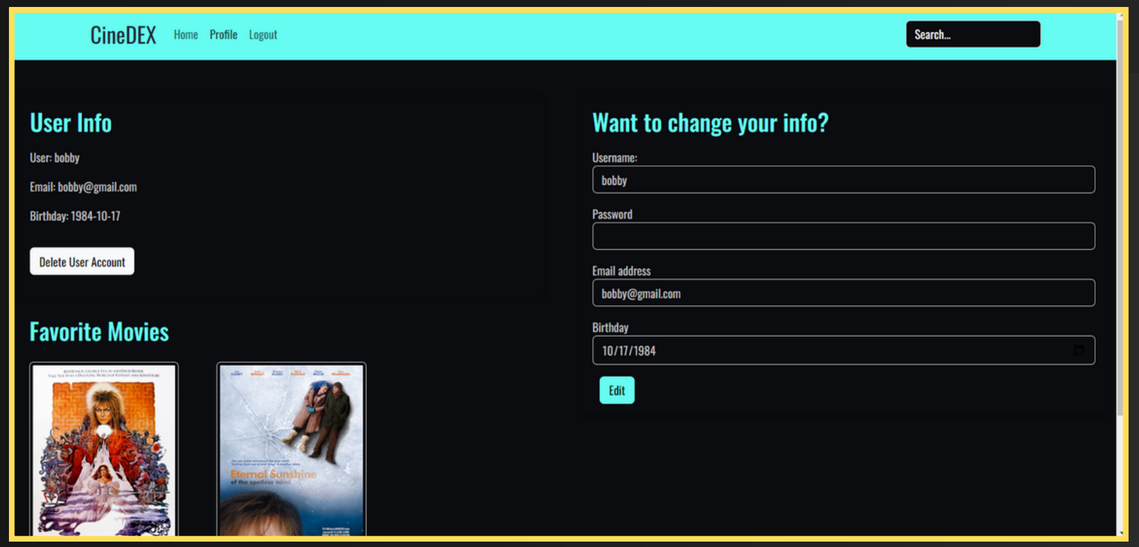
After ensuring my server-side code was fully functional, I needed to create the interface to be used by users to make requests and get responses from the server-side. Using React and Redux I created components for each of the interface views of the application.
After developing the logic to support the requests from users and handle data through the REST API endpoints, I used React Bootstrap to create aesthetically appealing and consistent styling.

CineDEX

Interface Views



Registration View

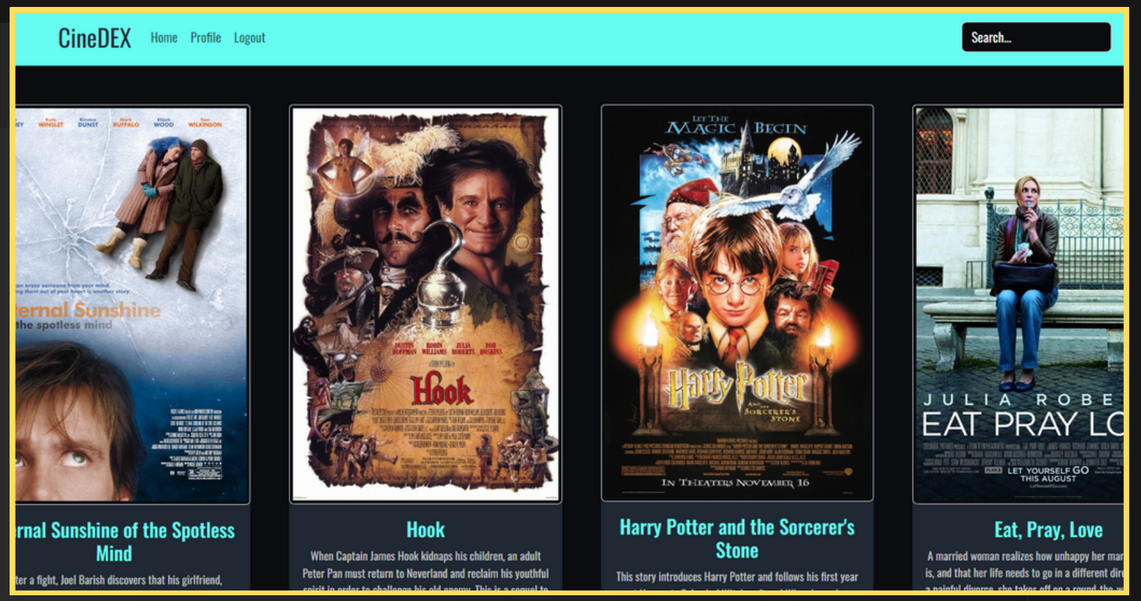


Profile View

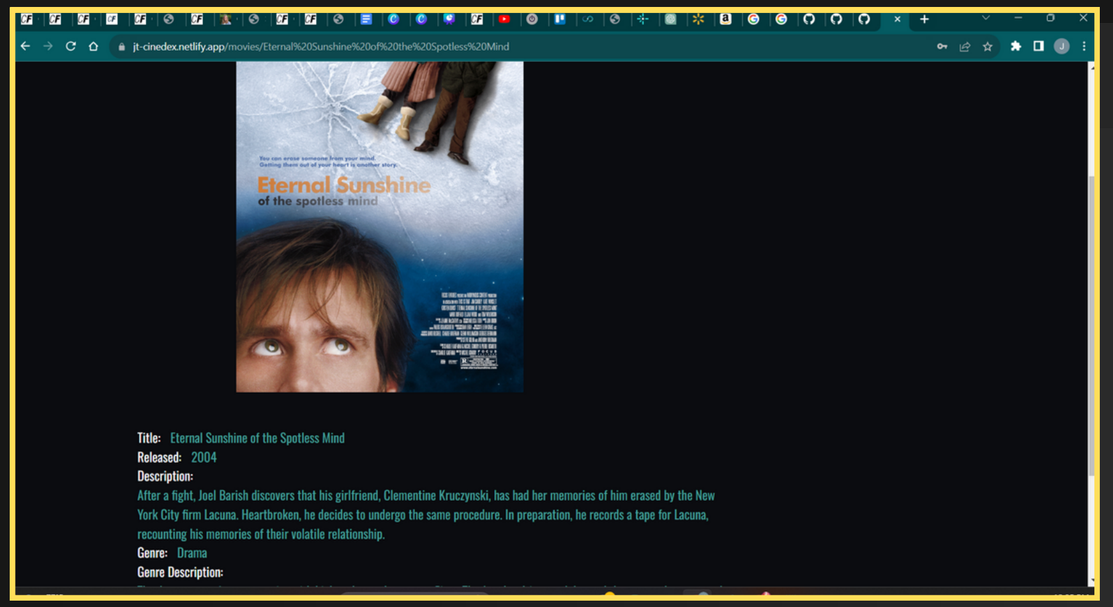
Users can register, log in, create a profile and browse movies.

CineDEX

Interface Views



Main View



Movie View

Upon clicking the details button on a particular movie, the user can access release date, a movie synopsis, and genre and director information.

CineDEX

Retrospective

What went well

As a very organized person and analytical thinker, I really enjoyed working on the server-side portion and database and felt I was able to complete it quickly and efficiently.

Challenges

While I struggled a bit at first with the logic behind the interaction between components, I tried to take more time to grasp the new concepts I learned. Finding the time to do this was also a challenge.

Final thoughts

I believe I would have benefitted from creating wireframes of each view so I could get a better feel for exactly what logic was required for each component and how it would interact with the other components as well as with styling.