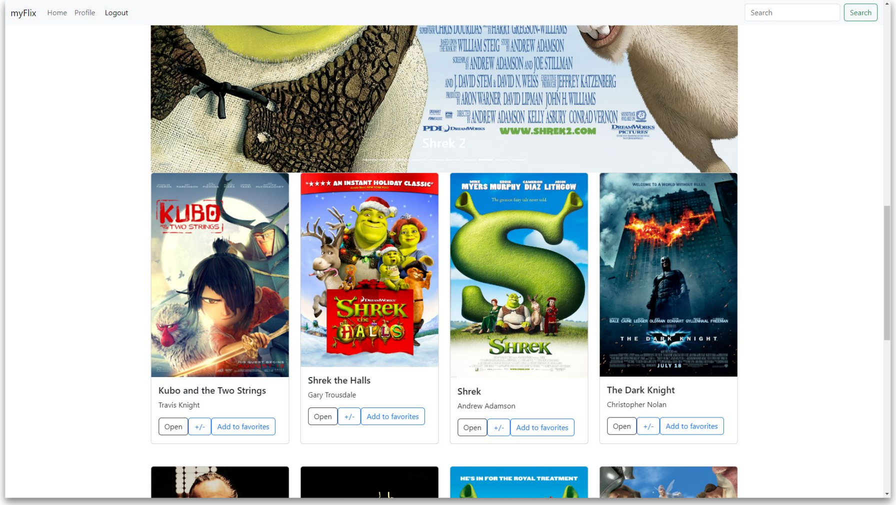
Case Study for myFlix full-stack Project

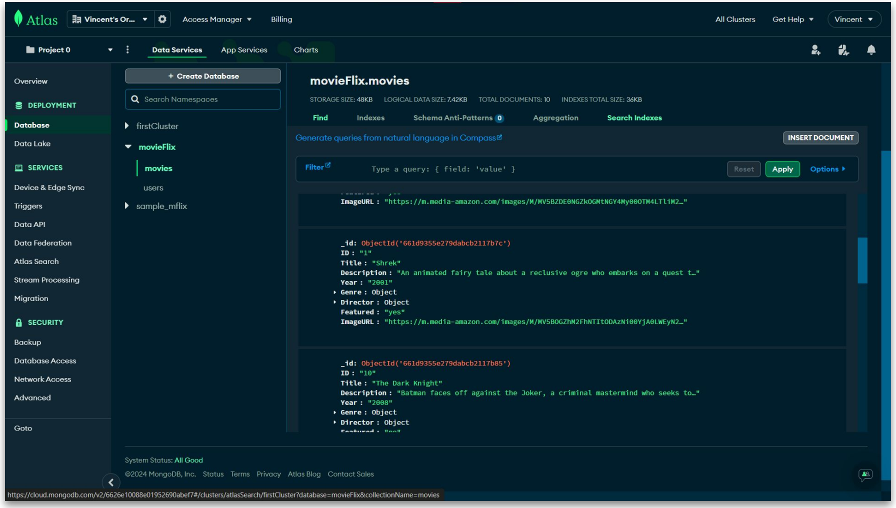


Kubo and the Two Strings
Travis Knight
Open +/- Add to favorites

Shrek the Halls
Gary Trousdale
Open +/- Add to favorites

Shrek
Andrew Adamson
Open +/- Add to favorites

The Dark Knight
Christopher Nolan
Open +/- Add to favorites



movieFlix.movies

STORAGE SIZE: 48KB LOGICAL DATA SIZE: 7.42KB TOTAL DOCUMENTS: 10 INDEXES TOTAL SIZE: 36KB

```
ImageURL : "https://m.media-amazon.com/images/M/MV5BZDE0NGZkOGMtNGY4My00OTM4LTliM2…"

_id: ObjectId('661d9355e279dabcb2117b7c')
ID : "1"
Title : "Shrek"
Description : "An animated fairy tale about a reclusive ogre who embarks on a quest t…"
Year : "2001"
▸ Genre : Object
▸ Director : Object
Featured : "yes"
ImageURL : "https://m.media-amazon.com/images/M/MV5BOGZhM2FhNTItODAzNi00YjA0LWEyN2…"

_id: ObjectId('661d9355e279dabcb2117b85')
ID : "10"
Title : "The Dark Knight"
Description : "Batman faces off against the Joker, a criminal mastermind who seeks to…"
Year : "2008"
▸ Genre : Object
▸ Director : Object
Featured : "no"
```

System Status: All Good

©2024 MongoDB, Inc. Status Terms Privacy Atlas Blog Contact Sales

https://cloud.mongodb.com/v2/6626e10088e01952690abef7#/clusters/atlasSearch/firstCluster?database=movieFlix&collectionName=movies

Overview

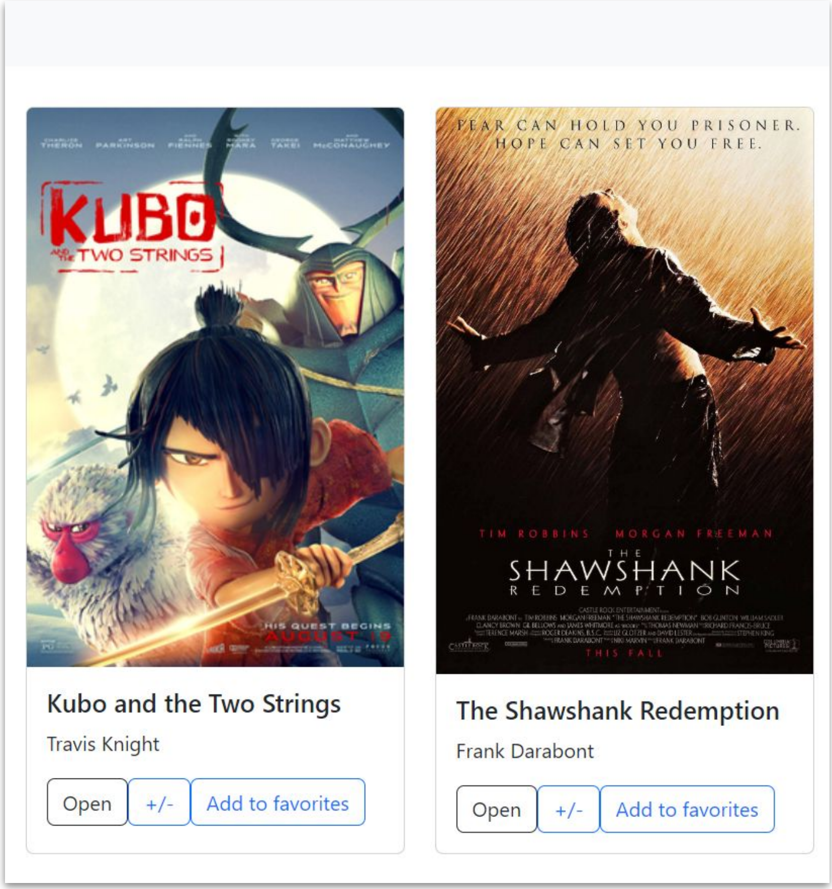
myFlix is a web application developed using the MERN stack, offering users detailed information on movies, directors, and genres. Users can create accounts, update their profiles, and curate a list of favorite films.

Purpose & Context

This project was created as part of a web development course at CareerFoundry to showcase advanced skills in full-stack JavaScript development.

Objective

The goal of myFlix was to build a comprehensive full-stack application, from server-side to client-side, as an ambitious project to feature in a professional portfolio.



Kubo and the Two Strings

Travis Knight

The Shawshank Redemption

Frank Darabont

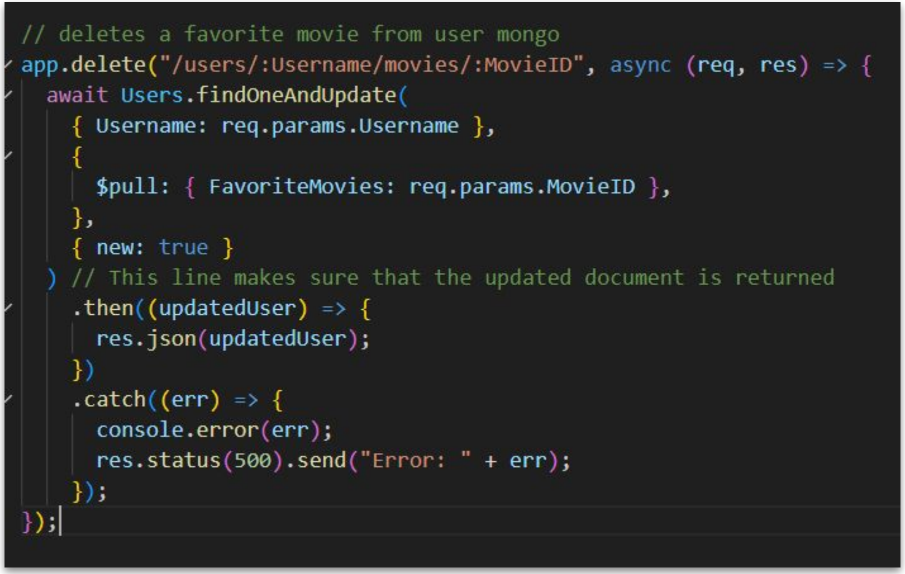
Approach

Server-side

I developed a RESTful API using Node.js and Express to interact with a non-relational MongoDB database. The API supports standard HTTP methods like GET and POST and utilizes CRUD operations for data retrieval and storage. It delivers movie data in JSON format.

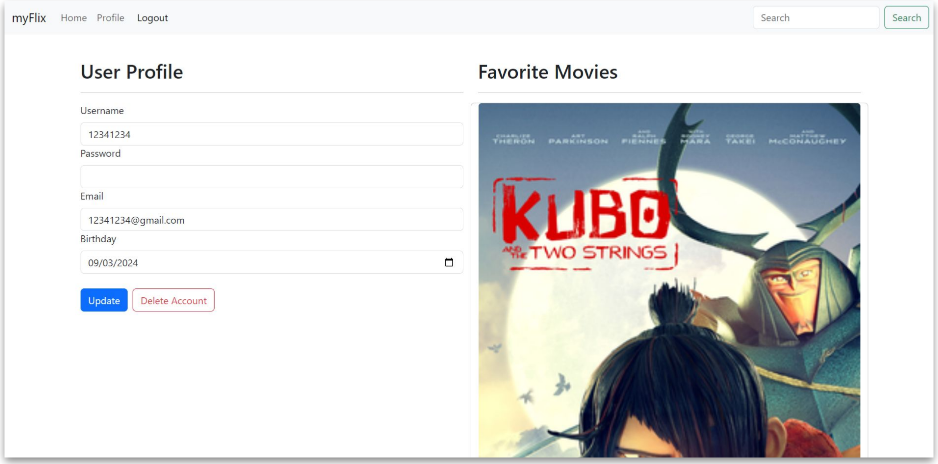
The Endpoints

To test the API, I used Postman. Additionally, I implemented both basic HTTP authentication and JWT authentication for user authentication and authorization.



```
// deletes a favorite movie from user mongo
app.delete("/users/:Username/movies/:MovieID", async (req, res) => {
  await Users.findOneAndUpdate(
    { Username: req.params.Username },
    {
      $pull: { FavoriteMovies: req.params.MovieID },
    },
    { new: true }
  ) // This line makes sure that the updated document is returned
    .then((updatedUser) => {
      res.json(updatedUser);
    })
    .catch((err) => {
      console.error(err);
      res.status(500).send("Error: " + err);
    });
});
```

Client-Side



User Profile

Username
12341234

Password

Email
12341234@gmail.com

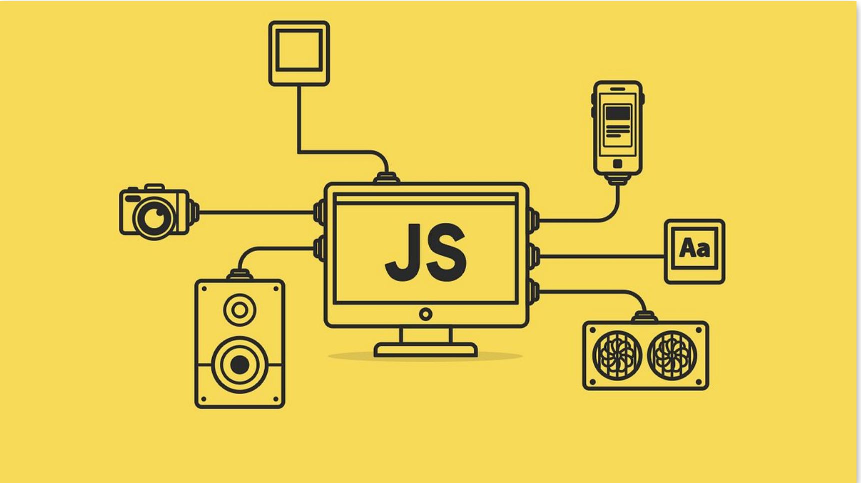
Birthday
09/03/2024

Favorite Movies

After finishing the API, I began developing the user interface for interacting with the server. This is a single-page, responsive application built using React. The app offers various views, including a main view displaying all movies, a single movie view showing detailed information and allowing users to add favorites, as well as login, registration, and profile views. The profile view enables users to update their personal information and manage their list of favorite movies.

Challenges

This project was particularly rewarding as it challenged me the most, pushing me to master complex concepts. I thoroughly enjoyed developing the API and working with database structures, and as a result, I became much more proficient in using Bash in the terminal. Building the client side allowed me to gradually refine my skills in React, gaining a deeper understanding of how to achieve desired outcomes. I’m especially grateful for the guidance of my expert tutor and mentor, whose support was instrumental in helping me complete the project to the standard I had envisioned.



Duration

This project spanned approximately 11 weeks to complete, largely due to the steep learning curve associated with mastering a new library like React. The extended timeline reflects the in-depth learning required to fully grasp and apply its concepts effectively.

Thank You For Viewing!

Would you like to connect with me?

My Gmail: VincentAThorne2005@gmail.com