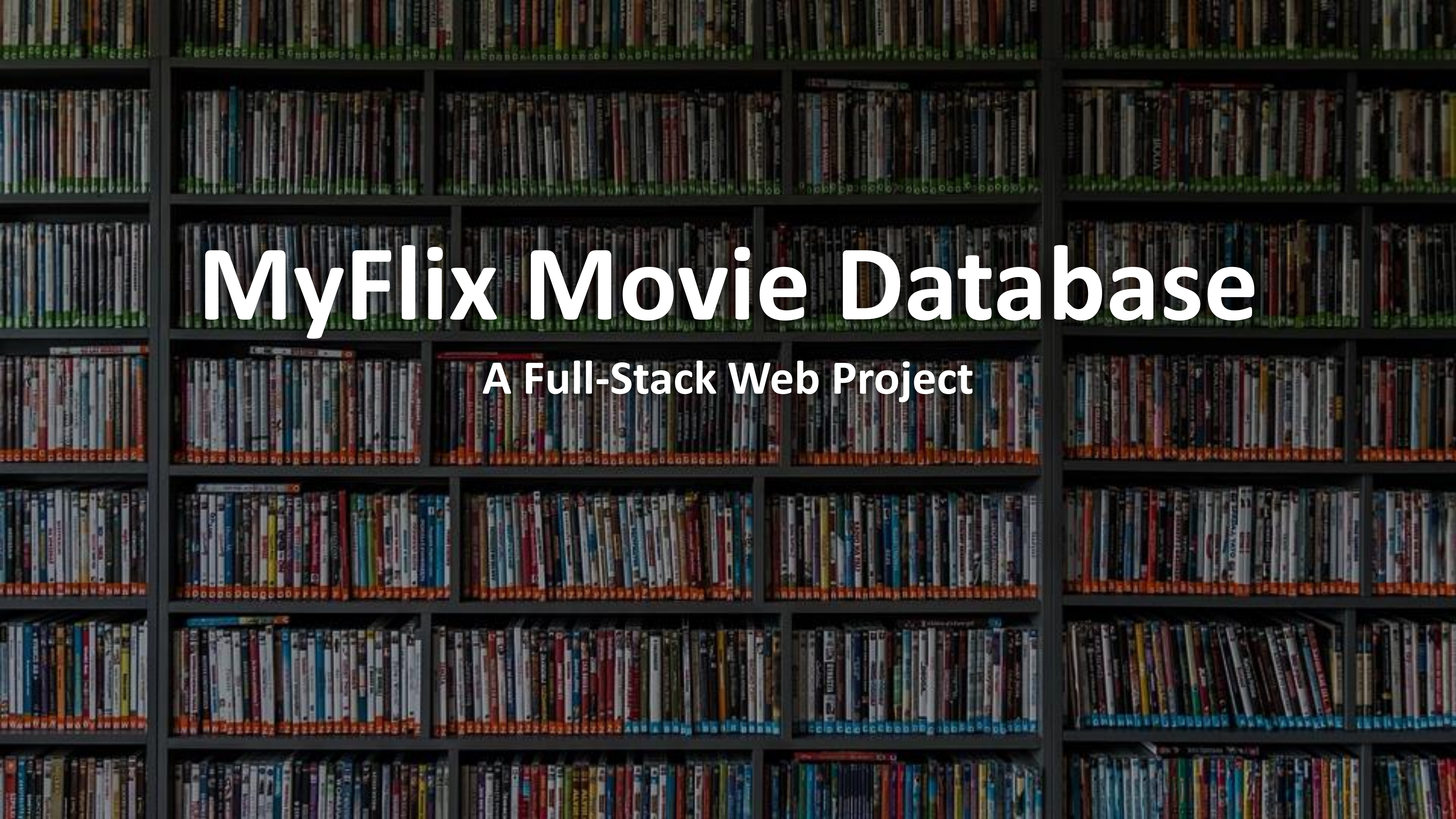


MyFlix Movie Database

A Full-Stack Web Project

Introduction

Overview

MyFlix Movie Database is a full-stack project with an API and Database as backend and a React single page application as frontend. User can register, edit their account, delete their account and add and remove movies from their list of favorites.

Purpose & Context

MyFlix was developed as part of the CareerFoundrys Full-Stack Web Development Course. This is a personal project to showcase my full-stack JavaScript skills.

Objective

The goal was to create a full-stack web application for my portfolio with server-side and client-side from scratch. The aim was to built an application where users can register, see and find movies and save them to their list of favorites.

Details

Role:

Full-Stack Web Developer

Project Scale:

2 Months

Primary Stakeholder:

CareerFoundry Full-Stack Web Development Course

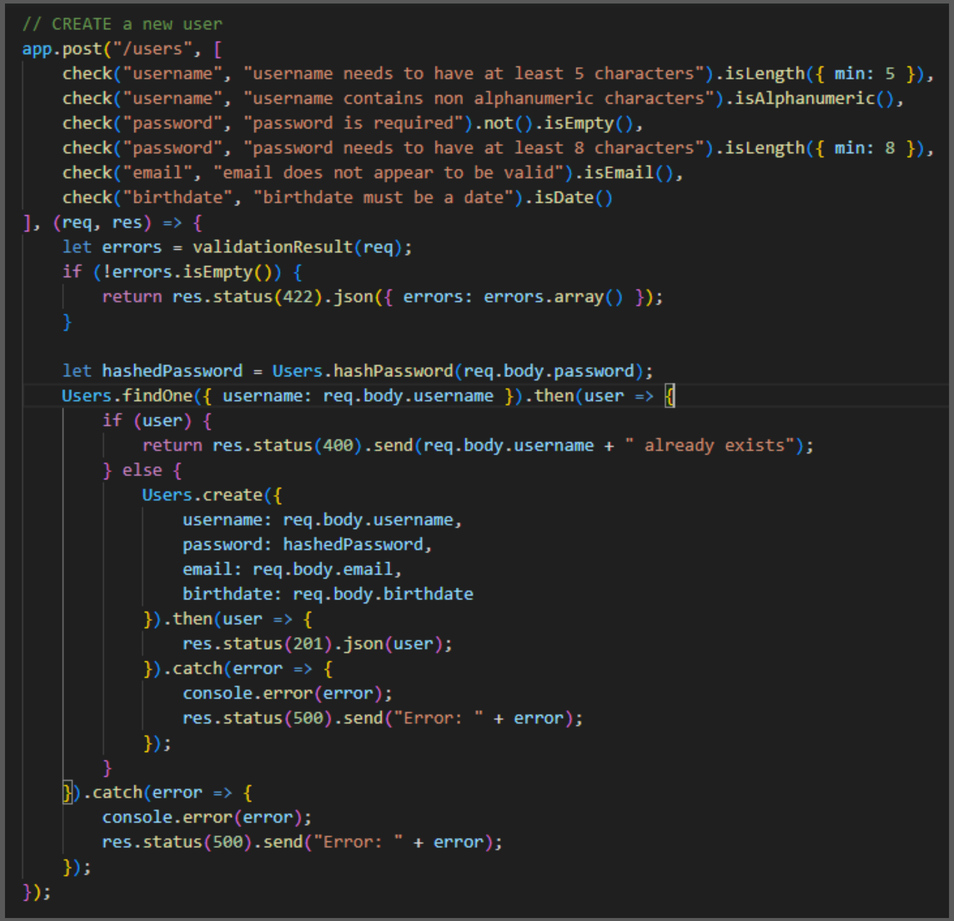
Tools used:

HTML CSS JS

node express

mongoDB

Server-side



```
// CREATE a new user
app.post("/users", [
    check("username", "username needs to have at least 5 characters").isLength({ min: 5 }),
    check("username", "username contains non alphanumeric characters").isAlphanumeric(),
    check("password", "password is required").not().isEmpty(),
    check("password", "password needs to have at least 8 characters").isLength({ min: 8 }),
    check("email", "email does not appear to be valid").isEmail(),
    check("birthdate", "birthdate must be a date").isDate()
], (req, res) => {
    let errors = validationResult(req);
    if (!errors.isEmpty()) {
        return res.status(422).json({ errors: errors.array() });
    }

    let hashedPassword = Users.hashPassword(req.body.password);
    Users.findOne({ username: req.body.username }).then(user => {
        if (user) {
            return res.status(400).send(req.body.username + " already exists");
        } else {
            Users.create({
                username: req.body.username,
                password: hashedPassword,
                email: req.body.email,
                birthdate: req.body.birthdate
            }).then(user => {
                res.status(201).json(user);
            }).catch(error => {
                console.error(error);
                res.status(500).send("Error: " + error);
            });
        }
    }).catch(error => {
        console.error(error);
        res.status(500).send("Error: " + error);
    });
});
```

I started with the backend. The backend is programmed Node.js and MongoDB. It is important in order to save Movie and User Data and the User Favorite Lists and to serve as an API for the frontend in order to provide the correct data.

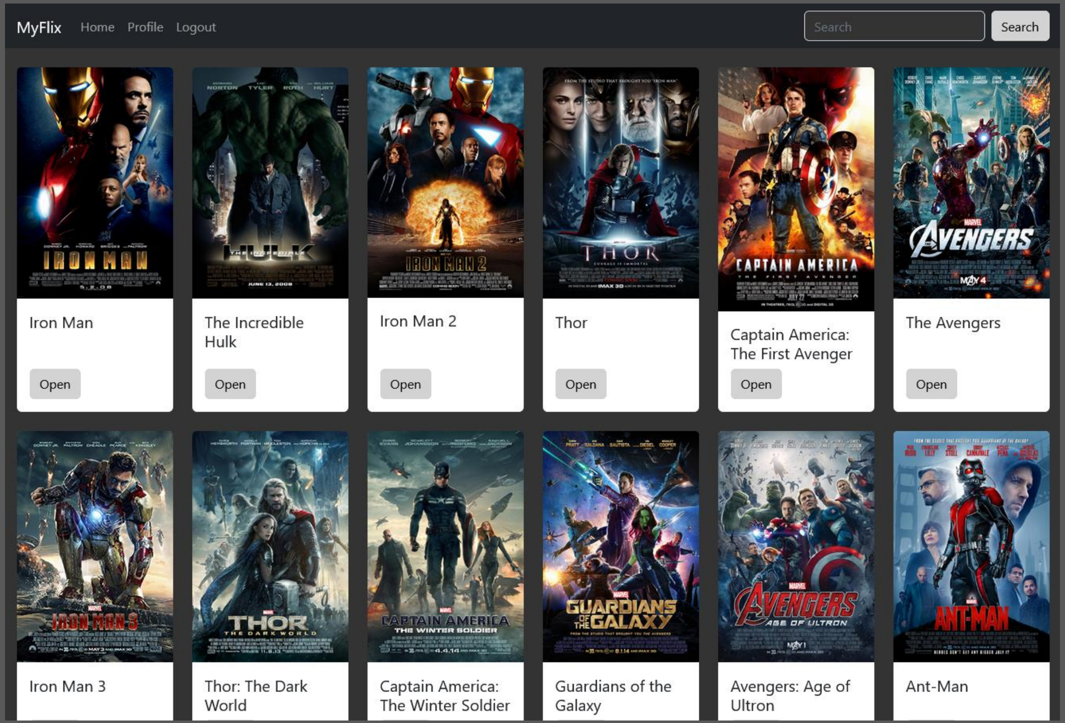
First a defined movie and user endpoints and implemented them step by step while also writing a documentation about them. This went hand in hand with connecting to the database and making the correct database queries.

Second I implemented user authentication in order to provide user accounts and favorite lists. For me, this was the most complicated part, where I really needed to dive deeper in order to understand everything. But I also liked this really much because I could learn a lot of new things.

Finally I tested all API endpoints with Postman to check whether they work as expected.

The next step was to implement the frontend and connect it with the backend to provide a seamless full stack project to the users.

Client-side



The second part of this project was the frontend which I implemented after finishing the backend. The frontend was created with React which I learned while going through this project.

I started with the MainView, where users see all movies and created React Routes in order to let the users navigate through the app.

Second I implemented the MovieView where users can see details about a movie and also can add the movie to their list of favorites.

Then I moved on to the user authentication methods and created the login and register pages for users to log in an register.

Finally I created the user details view where users can see their details and list of favorite movies and also implemented the logic for users to edit their details or delete their account.

Challenges

This was an awesome project that I really liked. I learned a lot about programming and web development from backend to frontend.

Since this was a 2 month project I had each, one month for the backend and one month for the frontend. This was very challenging because I also had to work in my full time job and only had time in the evenings and at the weekends. Due to a strict schedule and a lot of discipline I was able to finish the project on time.

I also had some challenges with the user authentication in the backend and the logic to update the user information in the frontend. But thanks to this challenges I dived deeper into those topics, learned a lot and figured out that this is really my thing that I like to do: Complicated logic in the backend.

Finally I want to say thank you to my tutor and mentor that were there to help my every time I needed help and guided me through this great project.