

myFlix web app

React frontend
development
process

```
<NavigationBar/>
```

```
<SignupView/>
```

```
<LoginView/>
```

```
<MainView/>
```

```
<MovieCard/>
```

```
<MovieView/>
```

```
<ProfileView/>
```

*All usernames, passwords or other similar information shown in this presentation is fictive and for illustrative purpose only

TABLE OF CONTENTS

01

Deciding which framework or library is the most suitable for the project needs

02

Setting up the development environment

03

Implementing the first views, components, and key functionalities

04

Loading real data from the API and render it in the components

05

Creating forms, executing frontend form validation and applying authentication logic

06

Styling the web app using React Bootstrap

TABLE OF CONTENTS

07

Implementing state-based routing,
user profile page and specific
features

08

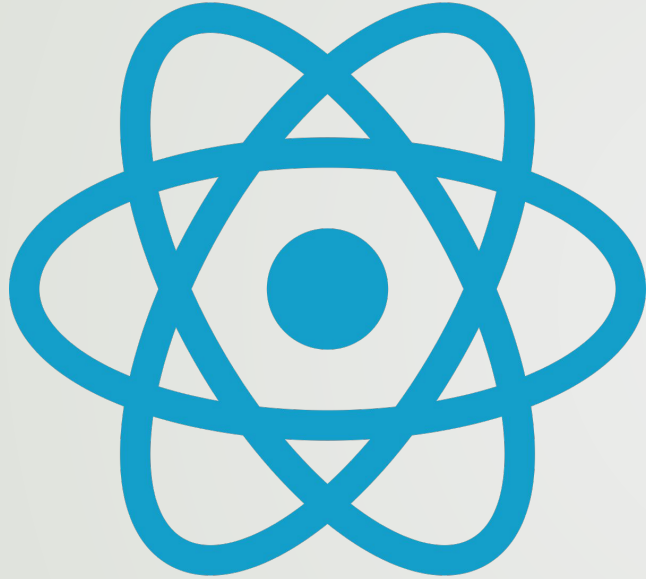
Finalizing code revision, refactoring,
testing and design last adjustments

09

Deploying and hosting the web app

10

Completing the README
documentation



Deciding which framework or library is the most suitable for the project needs

Skills used

Research
Analytical thinking

WHY WAS THIS STEP IMPORTANT

Choosing the right framework requires consideration, because it determines how an entire app will be built. It is therefore important to start on the right basis and give it a good thought at first.

Choosing which library to use can be less crucial since it won't affect the structure of the app, but it is still good practice to think about this at the beginning to have some directions / ideas that can be adapted along the way.

WHAT WAS THE GOAL

Exploring the different factors to be considered when choosing the right framework / library at the start of a project, then analyzing the different possibilities, selecting the best one for myFlix and explaining the reasons behind it.



CHALLENGES OR SPECIAL POINTS OF CONSIDERATION

To practice my analytical skills, I first analyzed, for three different company scenarios developing a web application, whether a library or a framework would be the most suitable for their needs, and explained why. For each scenario, I proposed a specific library or framework to be used between React, Angular and Vue.js, and detailed what features of the selected choice would be useful.

I then explained and justified why React was a good choice for the myFlix project, highlighting some of its key features.



Setting up the development environment

Skills used

Research
Problem solving

WHY WAS THIS STEP IMPORTANT

Organizing the development environment for efficient and well-oriented work from the beginning is not only necessary, but is also one of the keys to ensure smooth workflow and limit avoidable time loss due to inefficient project initialization.

WHAT WAS THE GOAL

Ensuring the project is properly ready to be worked on and developed, with all the necessary components and tools installed (Node.js and Node package manager, Github repository, code editor, Parcel built-tool and other dependencies).

CHALLENGES OR SPECIAL POINTS OF CONSIDERATION

At first, I had a compatibility problem with my Node.js version. I was not able to initialize my *package.json* file, so it was not possible to install Parcel as well. I eventually found out, after several researches on Github, that the issue was related to my nvm version (Node Version Manager), which I therefore updated to resolve my compatibility problem.

```
C:\Documents\myFlix>
```

```
npm init
```

```
npm install -g parcel
```

```
parcel src/index.html
```

```
<MainView/>
```

```
<MovieCard/>
```

```
<MovieView/>
```

Implementing the first
views, components, and key
functionalities

Skills used

Research
Problem solving
Code writing
Debugging

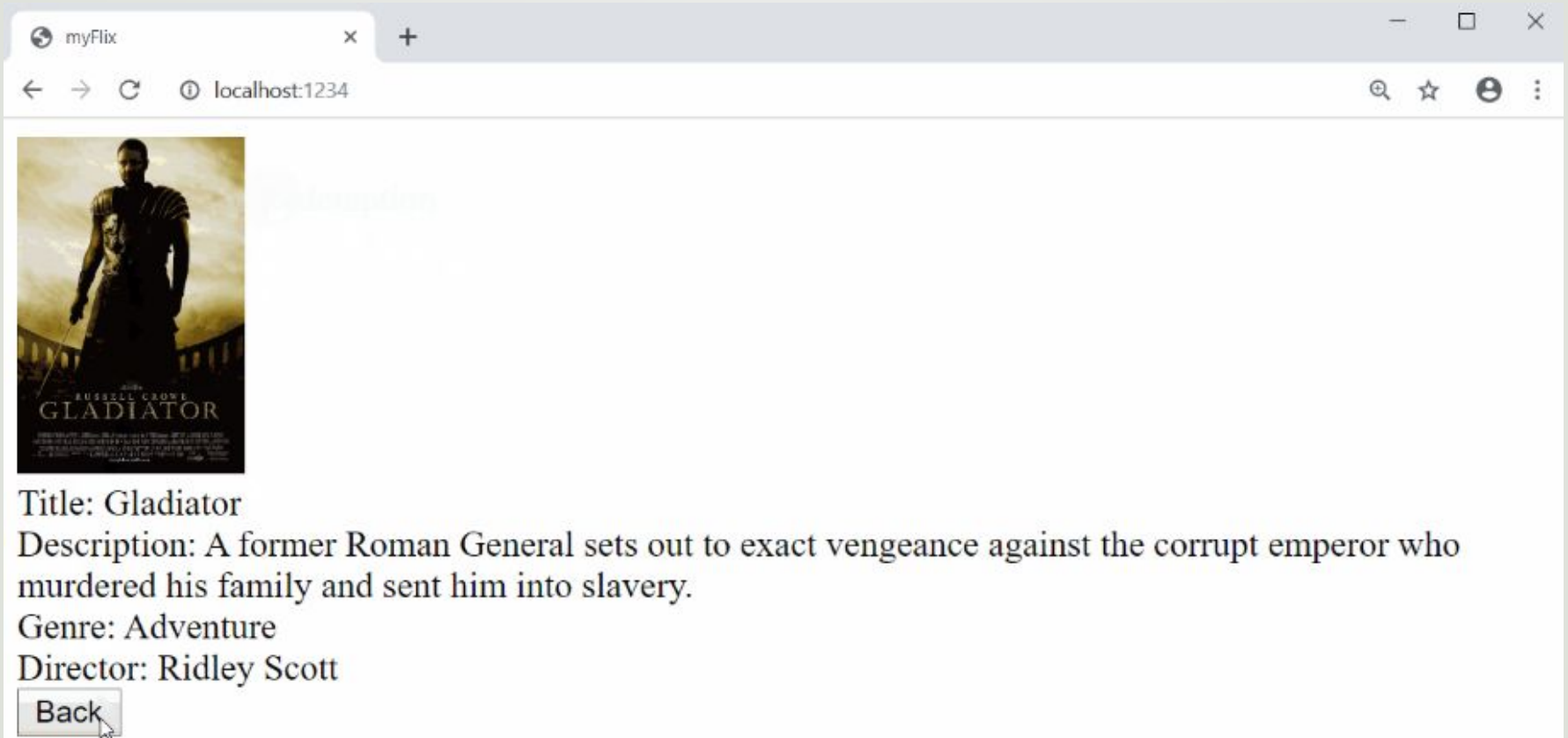
WHY WAS THIS STEP IMPORTANT

Implementing and rendering the basic first views / UIs of the web app was important at this point to start giving a form to the product, and pave the way to add future features over the next steps.

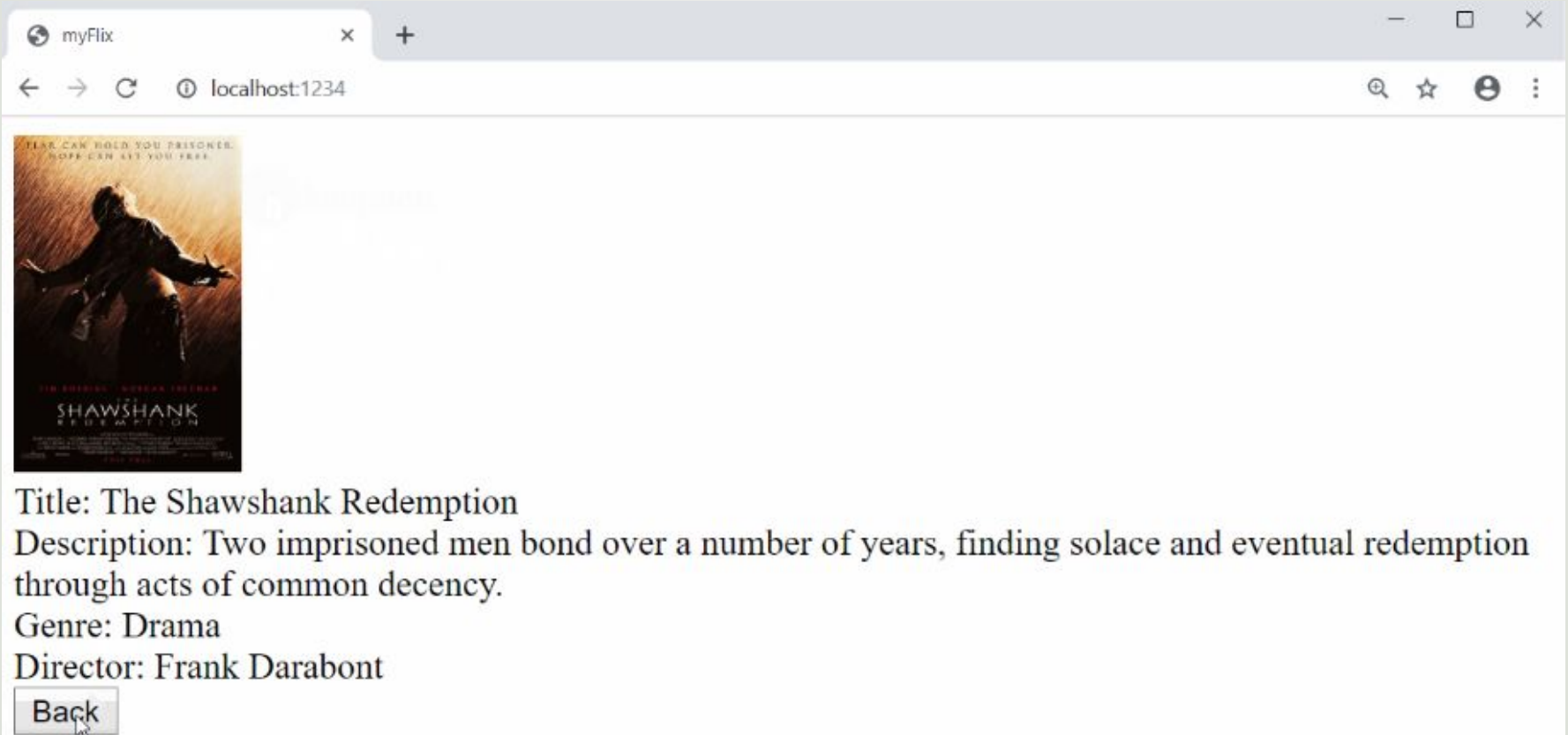
WHAT WAS THE GOAL

Creating the three main React components that would form the web app using JSX and movies mock-data (later replaced by real data fetched from an API).

FIRST RENDERED VIEW



FIRST RENDERED VIEW



CHALLENGES OR SPECIAL POINTS OF CONSIDERATION

I was in my first experimentation with the use of React state, props and event handling. It was therefore challenging at times to understand the mechanics behind these principles, but I was able to find several solutions by myself to assimilate everything and eventually deliver all expected requirements. Some solutions used to face the challenges included:

- Doing several researches on various platforms (Stack Overflow, Github...)
- Reading the official React documentation
- Analyzing other codes / projects using React to understand their logics
- Carrying out multiple tests locally to understand what work well and what doesn't, and adapt the codes consequently

DECISIONS MADE

As more files and components were created in this step, I had to organize the location and structure of each newly created folders and files, ensuring a coherent directory set up following React industry standards.

myFlix

```
>Dist
>img
>node_modules
  √ src
    √ components
      >main-view
        main-view.jsx
        main-view.scss
      >movie-card
        movie-card.jsx
        movie-card.scss
      >movie-view
        movie-view.jsx
        movie-view.scss
Index.html
Index.jsx
Index.scss
.gitignore
package-lock.json
package.json
README.md
```



Fetching real data from the API and render it in the components

Skills used

Research
Problem solving
Code writing
Debugging

WHY WAS THIS STEP IMPORTANT

Connecting the backend API I built previously to the frontend React was a key step to deliver real movie information to users. Up until this step, only non-significant movie mock-data (with the same structure as the data later fetched from the API) were rendered in the UI (using a handmade array written for testing purposes).

WHAT WAS THE GOAL

Replacing the mock-data with the real data fetched from my previously built API and database.

```
useEffect(() => {  
  fetch("https:...")  
    .then  
    ...  
}, [...]);
```

MOVIE DATA FETCHED (STORED IN A MongoDB DATABASE)

The screenshot displays the MongoDB Atlas web interface. On the left is a navigation sidebar with sections for 'DEPLOYMENT' (containing Database, Data Lake, and SERVICES like Device Sync, Triggers, Data API, Data Federation, Search, and Stream Processing) and 'SECURITY' (containing Backup, Database Access, Network Access, and Advanced). The main panel shows the 'myFlixDB.movies' collection. At the top right of the main panel are buttons for 'VISUALIZE YOUR DATA' and 'REFRESH'. Below the collection name, statistics are listed: STORAGE SIZE: 24KB, LOGICAL DATA SIZE: 7.86KB, TOTAL DOCUMENTS: 11, and INDEXES TOTAL SIZE: 20KB. A row of tabs includes 'Find' (selected), 'Indexes', 'Schema Anti-Patterns' (with a count of 0), 'Aggregation', and 'Search Indexes'. An 'INSERT DOCUMENT' button is in the top right of the main panel. A query filter bar contains a 'Filter' icon, a text input with the placeholder 'Type a query: { field: 'value' }', and buttons for 'Reset', 'Apply', and 'More Options'. The document content is displayed in a code editor with syntax highlighting. A small chat icon is visible in the bottom right corner.

DATABASES: 1 COLLECTIONS: 2

+ Create Database

Search Namespaces

myFlixDB

movies

users

myFlixDB.movies

STORAGE SIZE: 24KB LOGICAL DATA SIZE: 7.86KB TOTAL DOCUMENTS: 11 INDEXES TOTAL SIZE: 20KB

Find Indexes Schema Anti-Patterns 0 Aggregation Search Indexes

INSERT DOCUMENT

Filter Type a query: { field: 'value' } Reset Apply More Options

```
{
  "_id": ObjectId('648cc4cc9e8b6ddb7eae98b9'),
  "Title": "Dracula",
  "Description": "Bram Stoker's Dracula is a 1992 American Gothic horror film directed a...",
  "Genre": {
    "Name": "Horror",
    "Description": "Horror is a film genre that seeks to elicit fear or disgust in its aud..."
  },
  "Director": {
    "Name": "Francis Ford Coppola",
    "Bio": "Francis Ford Coppola is an American film director, producer, and scree...",
    "Birth": "1939-04-07",
    "Death": "NA"
  },
  "ImagePath": "https://www.themoviedb.org/t/p/w300_and_h450_bestv2/scFDS0U5uYAjcVTyjN...",
  "Featured": false
}
```

CHALLENGES OR SPECIAL POINTS OF CONSIDERATION

The API for myFlix had been previously designed to have an authentication system with token generation applied to it, and specific Cross-Origin Resource Sharing (CORS) parameters.

In order to render fetched data in the React frontend at this point, I had to address this by temporarily adjusting one of the backend endpoints (/movies) and the CORS set up to allow sending some information to the frontend while I was working on it (without token). I also had to make sure that the API data was being sent into the different components of my frontend correctly and that all the relations parent-child, useState, useEffect and props were working.

At the first try, the movies' data from the API was passing to the frontend, but the images of the movies from the database were not displaying correctly. After investigation, I found that the link for each movie image was not in the correct format. I therefore made the necessary adjustments, updated my movie database again in MongoDB, and the images then displayed correctly.

05

```
<SignupView/>
```

```
<LoginView/>
```

Creating forms, **executing**
frontend form validation and
applying authentication

Skills used

Research
Problem solving
Code writing
Debugging

WHY WAS THIS STEP IMPORTANT

Implementing well-designed forms is an important way of facilitating user input and interaction through React components and web apps.

WHAT WAS THE GOAL

Creating a user-friendly sign up form and log in form, as well as applying authentication measures and logic to ensure persistent access into the web app for logged-in users while they navigate between views.

LOG IN VIEW

Welcome back!

Ready for your next marathon?

Username

Username must be at least 5 characters long and contain only alphanumeric characters.

Password

Password can contain alphanumeric and non-alphanumeric characters.

Log in

SIGN UP VIEW

First time here?

Sign up now.

Username

Username must be at least 5 characters long and contain only alphanumerical characters.

Password

Password can contain alphanumeric and non-alphanumeric characters.

Email

Email must be in the following format : abc@domain.abc.

Birthday

Birthday is optional.

Sign up

USER ACCOUNT CREATED IN MongoDB DATABASE UPON SUCCESSFUL SIGN UP

The screenshot displays the MongoDB Atlas web interface. On the left, a sidebar contains navigation links for DEPLOYMENT, Database, Data Lake, SERVICES, Device Sync, Triggers, Data API, Data Federation, Search, Stream Processing, SECURITY, Backup, Database Access, Network Access, and Advanced. The 'Database' section is highlighted. The main panel shows the 'myFlixDB' database with two collections: 'movies' and 'users'. The 'users' collection is selected, displaying its details: STORAGE SIZE: 36KB, LOGICAL DATA SIZE: 6.39KB, TOTAL DOCUMENTS: 34, and INDEXES TOTAL SIZE: 36KB. Below these details are tabs for Find, Indexes, Schema Anti-Patterns, Aggregation, and Search Indexes. The 'Find' tab is active, showing a filter bar with a search query input field containing '{ field: 'value' }'. Below the filter bar, a document is displayed in a code editor. The document contains the following fields: _id (ObjectId), Username ('Testuser'), Password ('\$2b\$10\$RlSazJP\$RLhocNveTr.AI.QxXwxbt0D06PcntBeK.KeDgPmjEQIt.'), Email ('Testuser@hotmail.com'), Birthday (null), and FavoriteMovies (an array of three ObjectId values). At the bottom of the interface, there are navigation buttons for 'PREVIOUS' and 'NEXT', and a status bar indicating '21-34 of 34 results'.

DATABASES: 1 COLLECTIONS: 2

+ Create Database

Search Namespaces

myFlixDB

movies

users

myFlixDB.users

STORAGE SIZE: 36KB LOGICAL DATA SIZE: 6.39KB TOTAL DOCUMENTS: 34 INDEXES TOTAL SIZE: 36KB

Find Indexes Schema Anti-Patterns Aggregation Search Indexes

INSERT DOCUMENT

Filter Type a query: { field: 'value' } Reset Apply More Options

```
_id: ObjectId('64e03b3075564527c9c96af3')
Username: "Testuser"
Password: "$2b$10$RlSazJP$RLhocNveTr.AI.QxXwxbt0D06PcntBeK.KeDgPmjEQIt."
Email: "Testuser@hotmail.com"
Birthday: null
FavoriteMovies: Array
  0: ObjectId('648cc25b9e8b6dbb7eae98b2')
  1: ObjectId('648cc4289e8b6dbb7eae98b7')
  2: ObjectId('648cc47b9e8b6dbb7eae98b8')
```

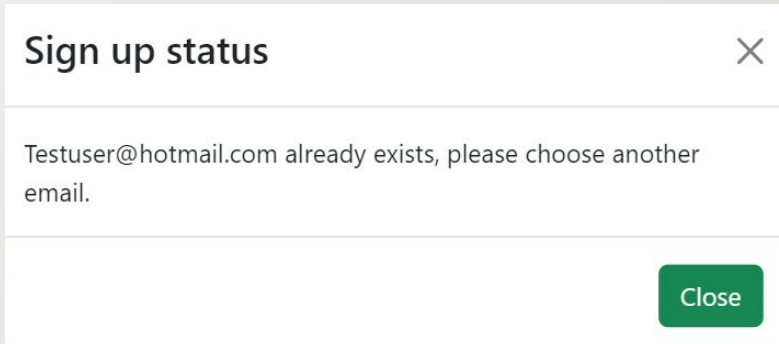
PREVIOUS 21-34 of 34 results NEXT

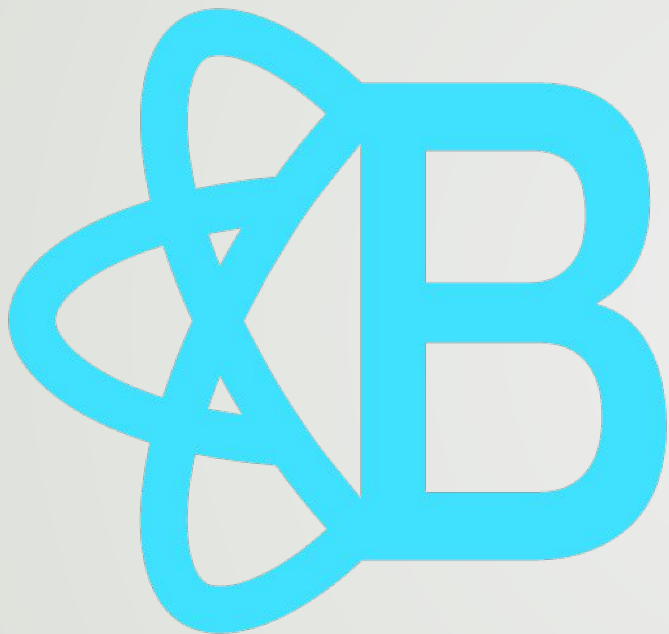
CHALLENGES OR SPECIAL POINTS OF CONSIDERATION

To test the authentication implemented, I had to revert the previous adjustments made on the backend (re-enabling token-based authentication). I also needed to adjust the logic of some endpoints in the backend to ensure that it was impossible for a user to create or update an account with an already existing username or email address in the database. I later did several tests to validate the correct implementation of these backend updates.

DECISIONS MADE

Even though it wasn't in the project requirements, I chose to make additional updates to some endpoints in the API to ensure that each user has a unique username and email address, and so create a more realistic and functional web app.





Styling the web app using React Bootstrap

Skills used

Creative thinking
Research
Code writing

WHY WAS THIS STEP IMPORTANT

Applying principles of good design to a React web app is as important as the app structure and functioning, since it plays a big part in how users see the product and want to interact with it.

WHAT WAS THE GOAL

Shifting gears from technical implementation of components to focus on app's design, polishing his appearance and ensuring responsiveness to be fit on any screen size devices.

```
const [showAddFavoriteModal, setShowAddFavoriteModal] = useState(false);  
const [showDeleteFavoriteModal, setShowDeleteFavoriteModal] = useState(false);  
const [showConfirmationModal, setShowConfirmationModal] = useState(false);  
const [showErrorModal, setShowErrorModal] = useState(false);
```

CHALLENGES OR SPECIAL POINTS OF CONSIDERATION

Thanks to the React Bootstrap extensive online documentation, the design of the web app itself was particularly smooth.

The main challenge was more about ensuring that the design applied to the computer web version of myFlix was equally appealing and user-friendly for the smaller devices / mobile phones. Tests have been done to adjust each UI depending on various screen sizes.

DECISIONS MADE

I chose all the visuals and the content organization within each view of myFlix. I made all my decisions based on how I thought users would like to interact with myFlix, and I've implemented, inspired by other well-designed web apps, a consistent styling across each view.

```
import { Row } from "react-bootstrap";  
import { Col } from "react-bootstrap";  
import { Button } from "react-bootstrap";  
import { Form } from "react-bootstrap";  
import { Modal } from "react-bootstrap";  
import { Card } from "react-bootstrap";
```

MAIN VIEW

LARGE

SCREEN

VISUAL



Coach Carter
Drama
[See details](#)



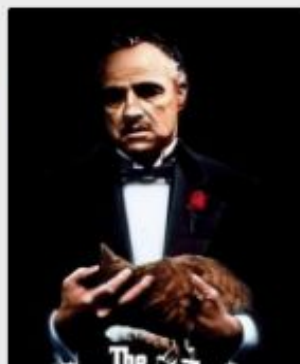
Gladiator
Drama
[See details](#)



Dracula
Horror
[See details](#)



Jurassic Park
Science fiction
[See details](#)



DETAILED MOVIE VIEW LARGE SCREEN VISUAL



Title

Jurassic Park

Description

The film is set on the fictional island of Isla Nublar, off Central America's Pacific Coast near Costa Rica, where a wealthy businessman, John Hammond, and a team of genetic scientists have created a wildlife park of de-extinct dinosaurs. When industrial sabotage leads to a catastrophic shutdown of the park's power facilities and security precautions, a small group of visitors, including Hammond's grandchildren, struggle to survive and escape the now perilous island.

Genre

Science fiction

Genre description

Science fiction (or sci-fi) is a film genre that uses speculative, fictional science-based depictions of phenomena that are not fully accepted by mainstream science, such as extraterrestrial lifeforms, spacecraft, robots, cyborgs, dinosaurs, mutants, interstellar travel, time travel, or other technologies.

Director

Steven Spielberg

Director bio

Steven Spielberg is an American filmmaker and a major figure of the New Hollywood era.

Director birth

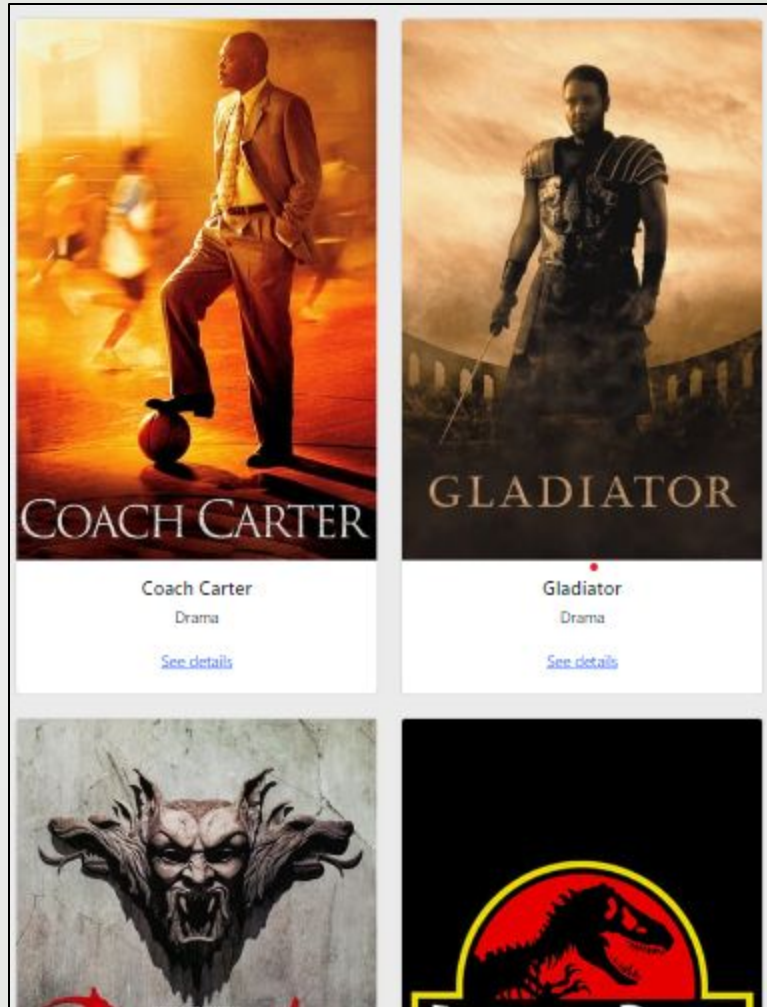
12/18/46

[Add to favorite](#)

[Back](#)

MAIN VIEW

MEDIUM SCREEN VISUAL



MAIN VIEW

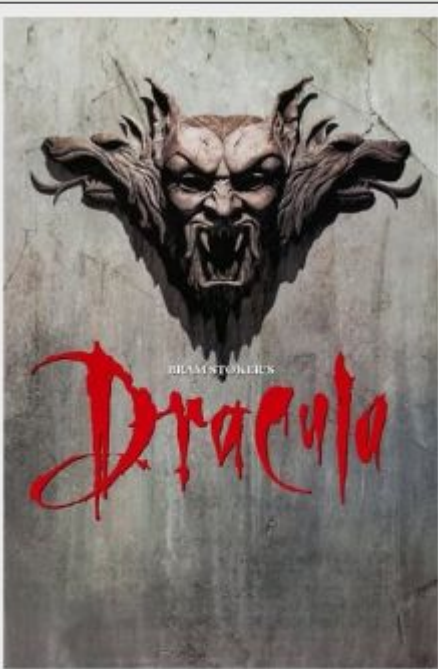
SMALL SCREEN VISUAL



DETAILED MOVIE VIEW

MEDIUM AND SMALL

SCREEN VISUAL



Title

Dracula

Description

Bram Stoker's Dracula is a 1992 American Gothic horror film directed and produced by Francis Ford Coppola, based on the 1897 novel Dracula by Bram Stoker.

Genre

Horror



Title

Dracula

Description

Bram Stoker's Dracula is a 1992 American Gothic horror film directed and produced by Francis Ford Coppola, based on the 1897 novel Dracula by Bram Stoker.

Genre

Horror

Genre description

Horror is a film genre that seeks to elicit fear or disgust in its audience for entertainment purposes.

Director

Francis Ford Coppola

Director bio

Francis Ford Coppola is an American film director, producer, and screenwriter. He is considered one of the major figures of the New Hollywood filmmaking movement of the 1960s and 1970s.

Director birth

04/07/39

Add to favorite

Back

```
<NavigationBar/>
```

```
<ProfileView/>
```

Implementing state-based routing, user profile page and specific features

Skills used

Research
Problem solving
Code writing
Debugging

WHY WAS THIS STEP IMPORTANT

Creating a user profile page was necessary to ensure users have access to their information and list of favorite movies. As for the specific features (movie filters and search bar), it made sense to implement those elements since users generally expect this kind of functionality to facilitate their navigation.

GET

DELETE

POST

PUT

WHAT WAS THE GOAL

Implementing state-based routing (with React router and React router DOM), so that users can navigate between the different views and obtain unique URLs.

Implementing a dynamic navigation bar.

Building the profile page so that users can view, update or delete their account, as well as consult their list of favorite movies and update it.

Implementing the filter movie button and the search bar.

NAVIGATION BAR - UNAUTHENTICATED USERS

myFlix [Log in](#) [Sign up](#)

Welcome back!

Ready for your next marathon?

Username

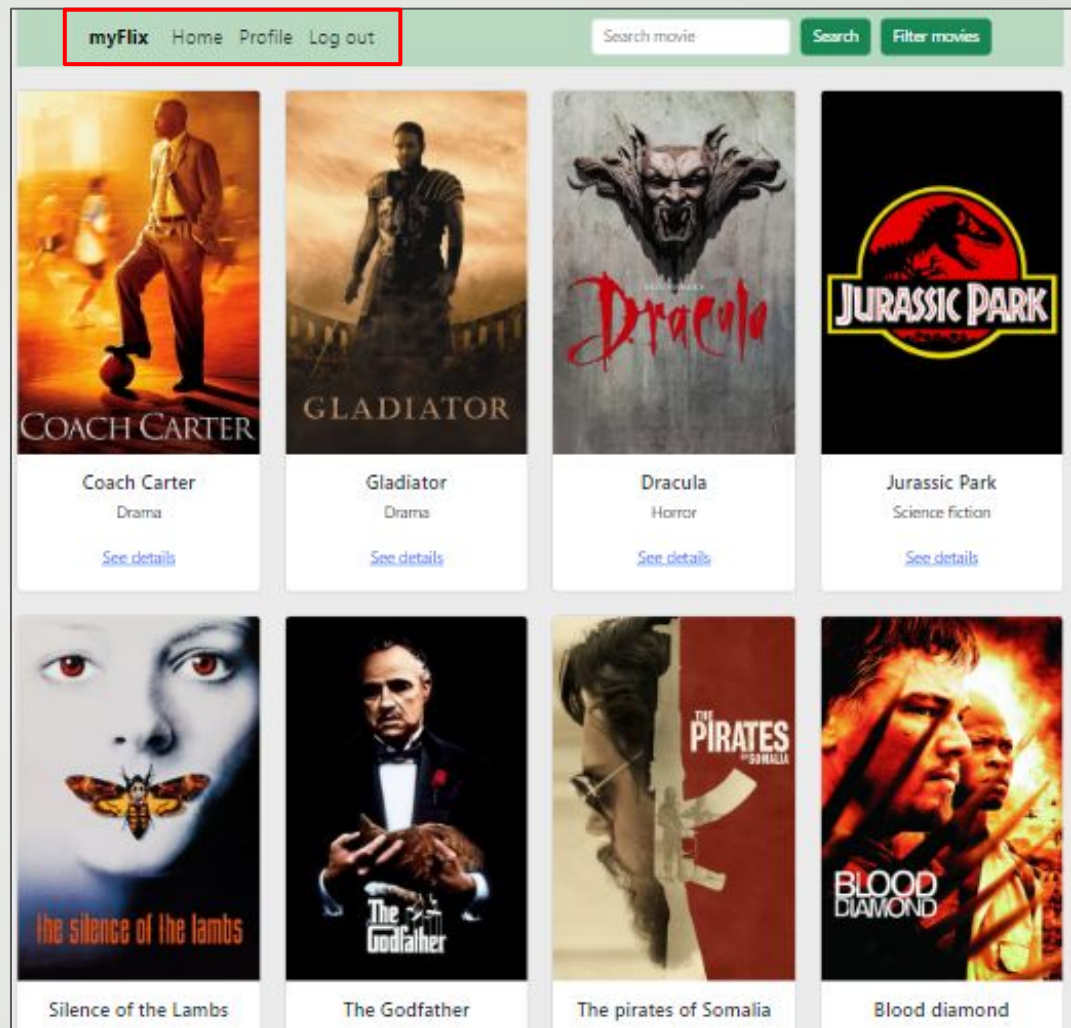
Username must be at least 5 characters long and contain only alphanumerical characters.

Password

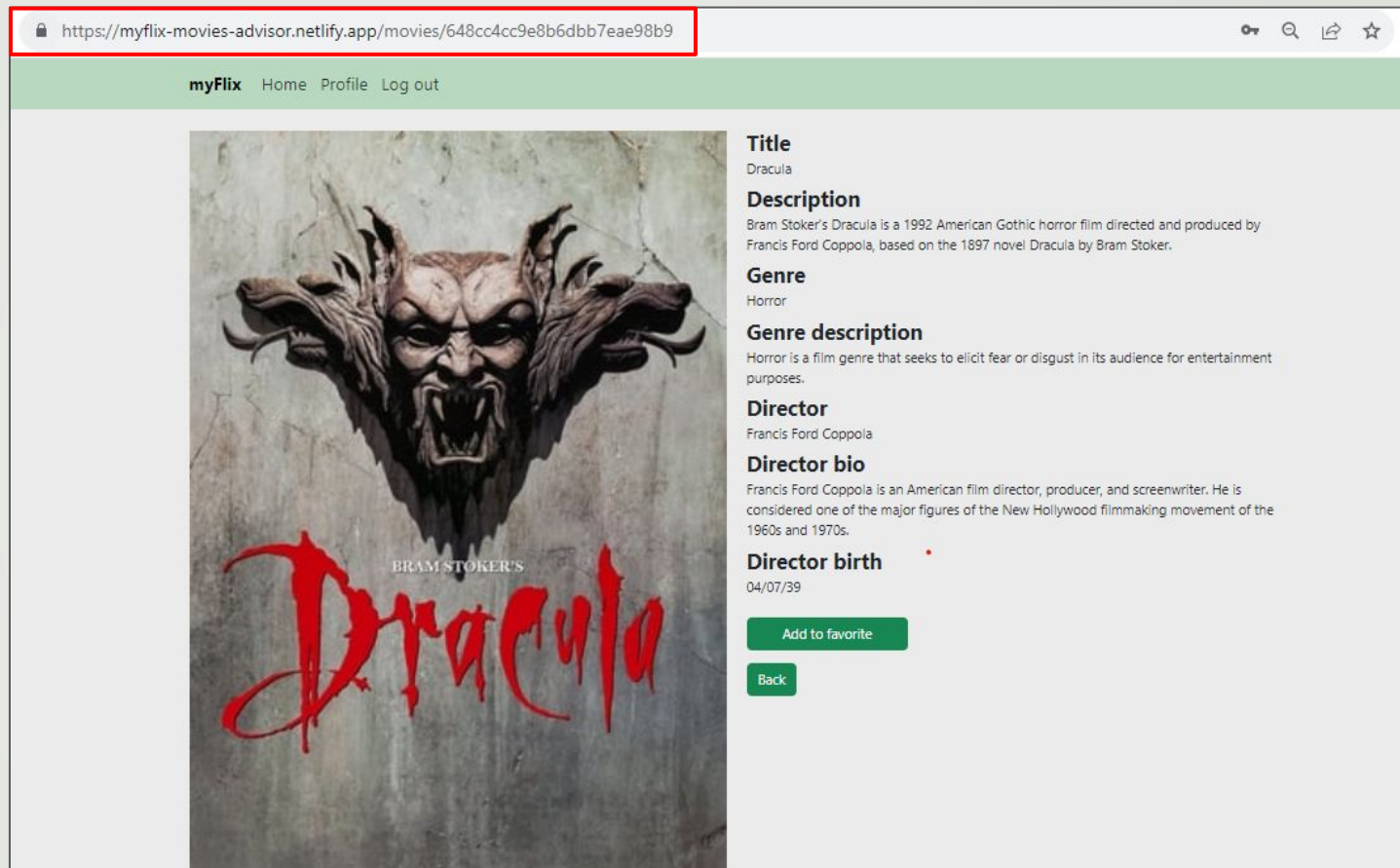
Password can contain alphanumeric and non-alphanumeric characters.

[Log in](#)

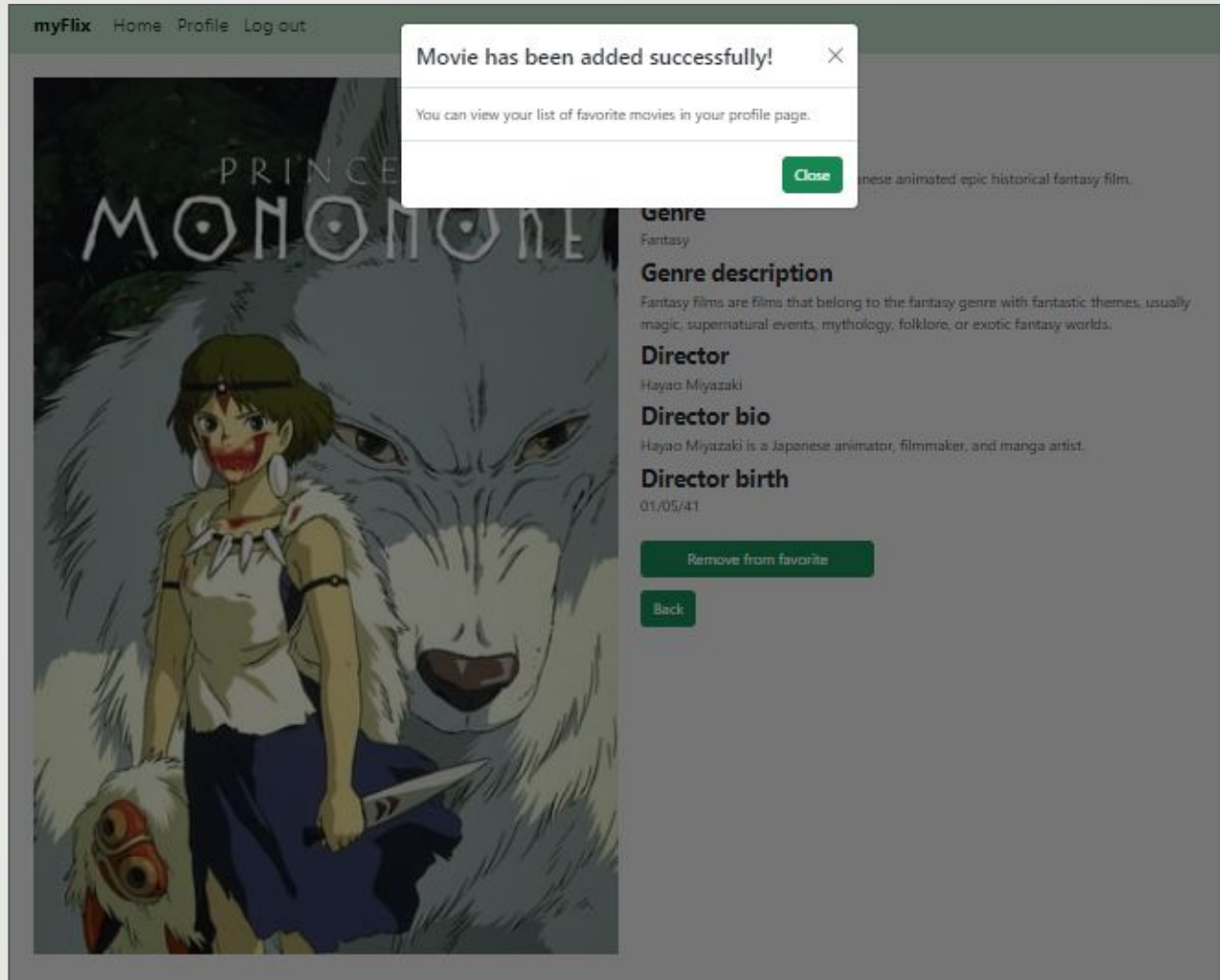
NAVIGATION BAR AUTHENTICATED USERS



STATE- BASED ROUTING



MOVIE ADDED TO FAVORITE (SHOWN IN PROFILE PAGE)



PROFILE VIEW (LARGE SCREEN VISUAL)

myFlix [Home](#) [Profile](#) [Log out](#)

User info

User: **Testuser**

Email: **Testuser@hotmail.com**

If you wish to update your information, please fill in the update form. All fields must be completed.

If you only want to change some information, enter your current information you want to keep in the corresponding field (e.g. username) along with the information you want to change in the other field(s).
Following successful update, you will be redirected to the log in page.

Update info

Username

Password

Email

Birthday
 ☐

Update



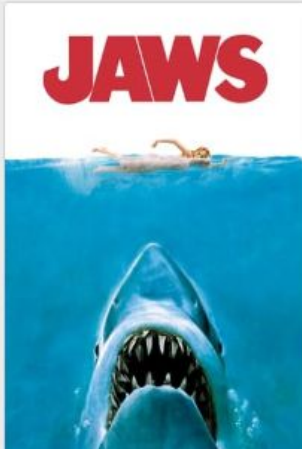
Delete account

Want to leave us?

By deleting your account, your data will be permanently deleted, and you will have to create a new account if you wish to return.

Delete

Favorite movies



PROFILE VIEW (MEDIUM AND SMALL SCREEN VISUAL)

myFlix

User info

User: **Testuser**

Email: **Testuser@hotmail.com**

If you wish to update your information, please fill in the update form. All fields must be completed.

If you only want to change some information, enter your current information you want to keep in the corresponding field (e.g. username) along with the information you want to change in the other field(s).
Following successful update, you will be redirected to the log in page.

Update info

Username

Password

Email

Birthday

mm/dd/yyyy

Update

Delete account

Want to leave us?

By deleting your account, your data will be permanently deleted, and you will have to create a new account if you wish to return.

Delete

Favorite movies

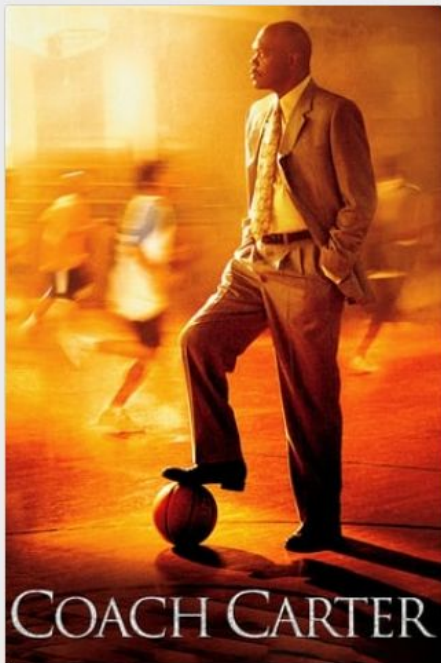


FILTER BUTTON AND SEARCH BAR

myFlix Home Profile Log out

Search

Filter movies



Coach Carter

Drama

[See details](#)



GLADIATOR

Gladiator

Drama

[See details](#)



BRAM STOKER'S
Dracula

Dracula

Horror

[See details](#)

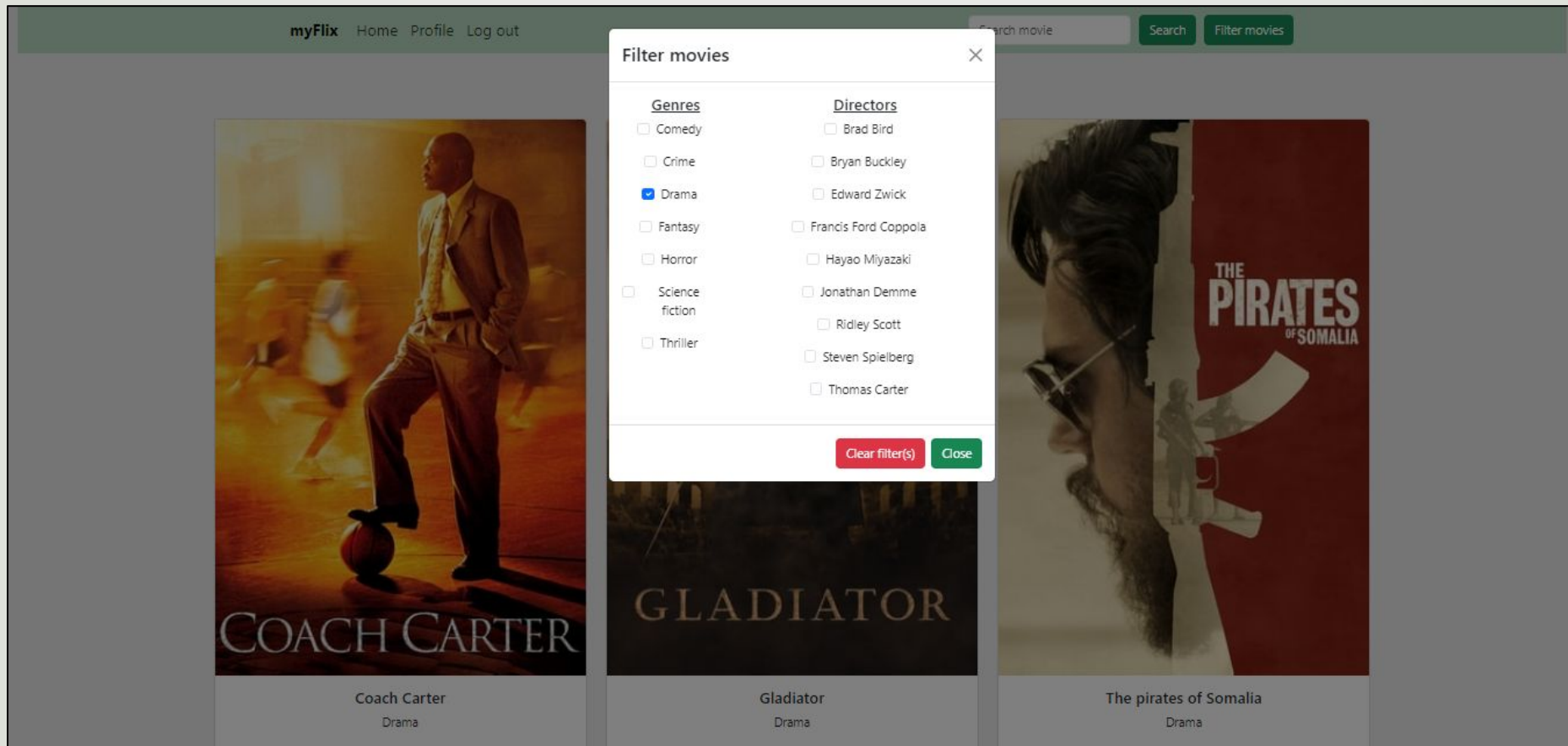


Jurassic Park

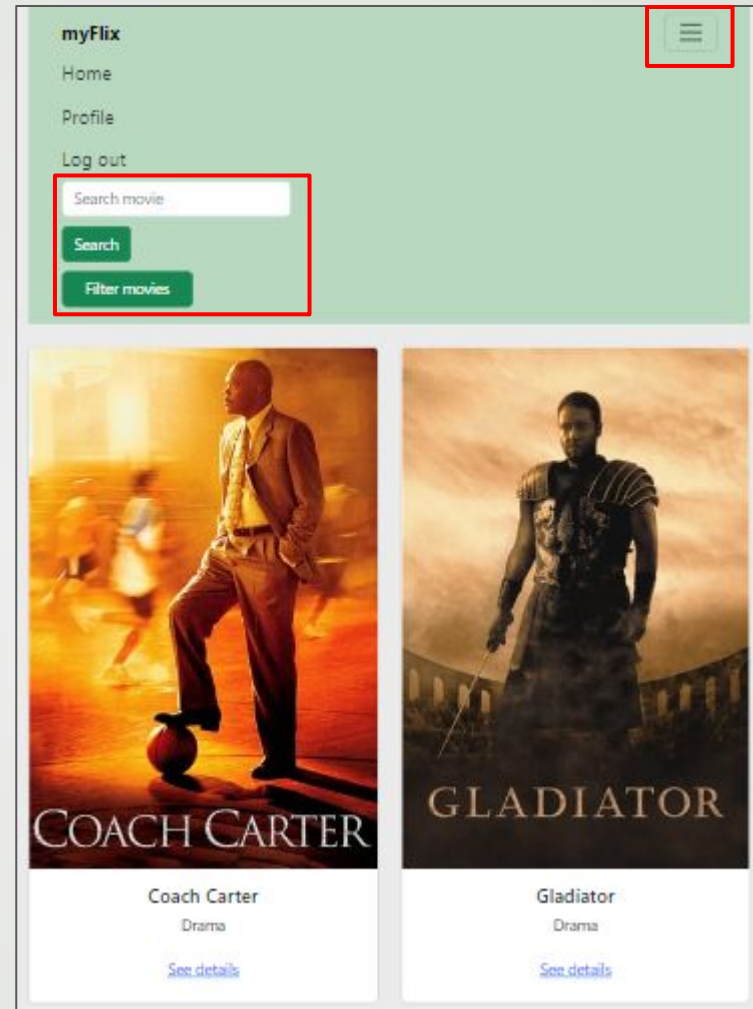
Science fiction

[See details](#)

FILTER BUTTON MODAL



FILTER BUTTON AND SEARCH BAR (COLLAPSED FOR MEDIUM AND SMALL SCREEN VISUAL)



DECISIONS MADE

I chose how to display the elements within the navigation bar, as well as the different parts composing the profile page. I also decided the logic behind the movie filter button, based on what I thought users would like to have as filtering options (genres and directors).

Search by...

Genres

Directors



**Finalizing code revision,
refactoring, testing and last
adjustments**

Skills used

Critical thinking

WHY WAS THIS STEP IMPORTANT

Ensuring that the codes are optimized, that the web app passes all the tests and that the final product is in it's best visual version was a logic thing to do for a sound public deployment.

WHAT WAS THE GOAL

Reviewing each code to make sure everything was optimized as much as possible in order to facilitate future modifications, additions or adjustments to the web app.

Adding comments and clarification points in the code where important for the benefit and better understanding of anyone else who may work on this project later.

Carrying out several tests on all the features to ensure everything work as it should.

Adjusting some final design details to ensure professional looking.

**CODE
COMMENTS
TO
FACILITATE
FUTURE
UPDATES AND
WORK**

```
→ /**Main logics for UPDATE user, DELETE user and remove MOVIE FROM FAVORITE LIST.
```



```
→ /**Logic to allow a user to update his information.
```

```
const handleUpdate = (event) => {  
  event.preventDefault();  
  const userData = {  
    Username: username,  
    Password: password,  
    Email: email,  
    Birthday: birthday  
  };  
  fetch(`https://my-weekend-movie-app-53a46e3377d7.herokuapp.com/users/${user.Username}`, {  
    method: "PUT",  
    body: JSON.stringify(userData),  
    headers: {  
      Authorization: `Bearer ${token}`,  
      "Content-Type": "application/json"  
    }  
  })  
    .then((response) => {  
      if (response.ok) {  
        setResponseMessage("Update successful");  
        setShowConfirmationModal(true);  
      } else if (response.status === 409) {  
        response.text().then((text) => {  
          setResponseMessage(text);  
          setShowErrorModal(true);  
        }).catch((error) => {  
          console.error("Error reading response data:", error);  
          setResponseMessage("Make sure you enter a username and email address that are
```


CHALLENGES OR SPECIAL POINTS OF CONSIDERATION

I positioned myself from the point of view of future colleagues who could work on this project. How can I make this project and my codes as clear as possible to promote its easy appropriation? I reviewed each file to bring improvements in certain places and add comments where I thought it could be useful.

DECISIONS MADE

This step was done on my own initiative and was not required in the project requirements. I made decisions regarding the improvement of certain codes, and the addition of comments where necessary, in order to set my mind to work in a collaborative environment already.



Deploying and hosting the web app

Skills used

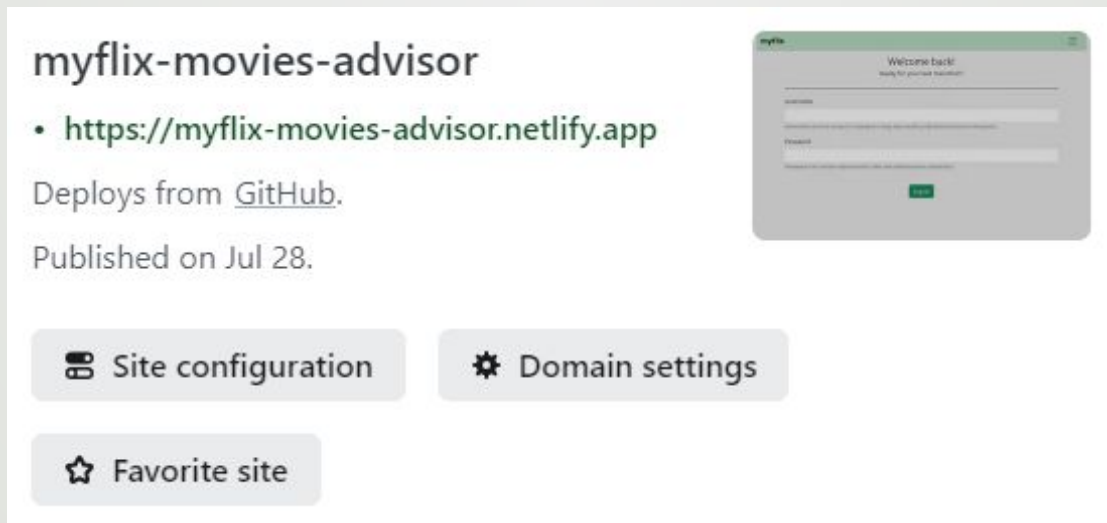
Research
Debugging

WHY WAS THIS STEP IMPORTANT

Ensure the web app is publicly available.

WHAT WAS THE GOAL

Deploying and hosting myFlix online on Netlify.



CHALLENGES OR SPECIAL POINTS OF CONSIDERATION

At first, the web app would not render properly in the web browser, despite a deployment that seemed successful on Netlify. I later identified the source of the problem by consulting the Netlify documentation. I ran several tests to troubleshoot the issue until I finally managed to adjust some parameters within Netlify, and redeploy my web app successfully.

DECISIONS MADE

Choosing the name of myFlix URL before deploying it online, having in mind an easy-to-remember domain for users.



README .

md

Completing the README
document for myFlix web
app

Skills used

Communication
Content writing

WHY WAS THIS STEP IMPORTANT

Ensure the web app is well documented and easily accessible by anyone interested.

WHAT WAS THE GOAL

Updating and completing the README file located in the myFlix React Github repository. The goal was to ensure that all relevant information regarding myFlix React is accessible under these four categories:

- Project description
- User interface
- Technical aspects
- App dependencies

CHALLENGES OR SPECIAL POINTS OF CONSIDERATION

Finding the right balance between giving the right level of information, while remaining as synthetic as possible. To help me, I made a first draft, which I then modified at times. I also drew inspiration from other READMEs I've consulted for similar projects and for which I found that the information presented was relevant.

DECISIONS MADE

I wrote the README documentation from A to Z, in terms of content, presentation and structure.

README SAMPLE - FULL VERSION ON GITHUB

📖 README



myFlix web app documentation (frontend - React)

Content

- Project description
- User interface
- Technical aspects
- App dependencies

Project description

myFlix web app has been created to serve as a reference in the domain of visual entertainment. Users can create an account and then log into *myFlix* to have access to information about different movies. They can search for movies, filter results based on different criteria and create lists of favorites. *myFlix* has been built in two parts: the frontend (here) and the backend ([see this repository for the backend part of *myFlix*](#)).

The objective of this part of the project (frontend) was to develop an easy-to-use and responsive web app using React for the best user-experience whenever they are accessing *myFlix* to read details about different movies or update their information.