

# myFlix web app

Frontend  
development  
process

```
<NavigationBar/>
```

```
<SignupView/>
```

```
<LoginView/>
```

```
<MainView/>
```

```
<MovieCard/>
```

```
<MovieView/>
```

```
<ProfileView/>
```

August	2023
--------	------

# TABLE OF CONTENTS

---

**01**

---

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

---

**02**

---

Setting up the React development environment

---

**03**

---

Implementing the first views, components and key functionalities

---

**04**

---

Loading real data from my 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 features and components

# TABLE OF CONTENTS

---

**07**

---

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

---

**08**

---

Finalizing code revision, refactoring,  
testing and design last adjustments

---

**09**

---

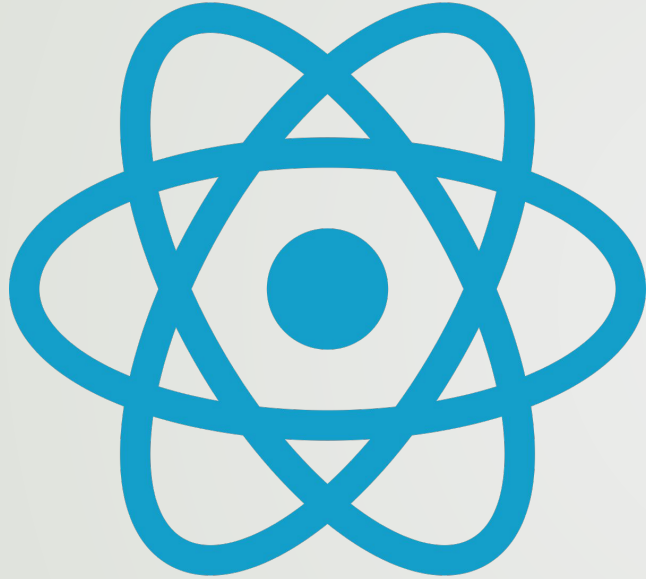
Deploying and hosting the web app

---

**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 for 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 real-life company scenarios developing a web app, whether a library or 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 the best tool for the myFlix project, highlighting some of its key features.



## **Setting up** the React development environment

---

### Skills used

---

Research  
Problem solving

# WHY WAS THIS STEP IMPORTANT

Organizing the coding environment for efficient and well-oriented work from the beginning is a key to ensure a 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 (CLI / Terminal / Shell, Node.js and Node package managers, Github repository, code editors, 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, and so it was not possible to install Parcel built-tool as well. I eventually found out, after several researches on Github, that the problem was related to my version management nvm (Node Version Manager), which I then updated to resolve my compatibility issues and launched the installation of Parcel.

```
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

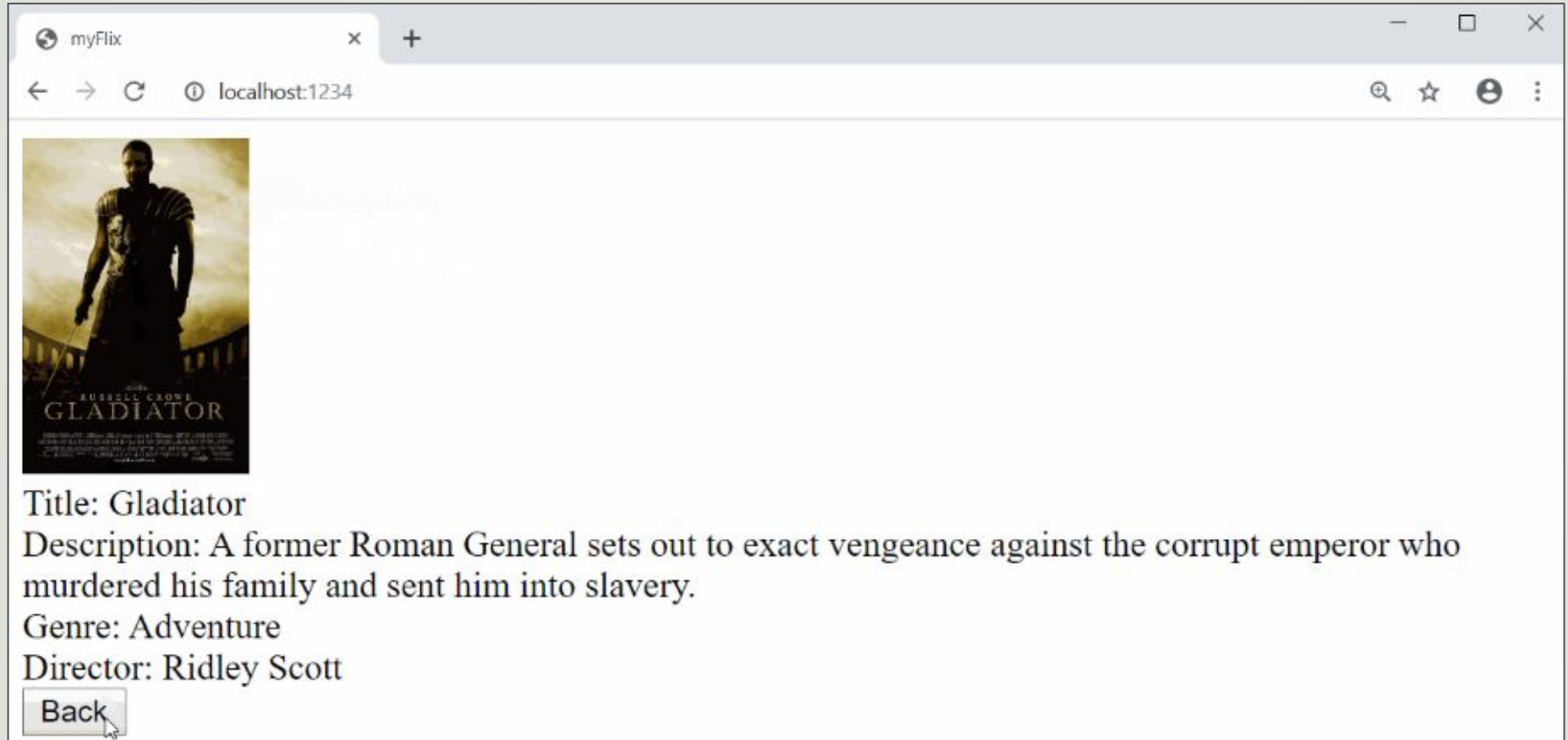
# **WHY WAS THIS STEP IMPORTANT**

Implementing and rendering the basic first views / UI structure of the web app was important at this point to start giving a form to the product, and pave the way to add future improvements and build more specific coding logics over the next steps.

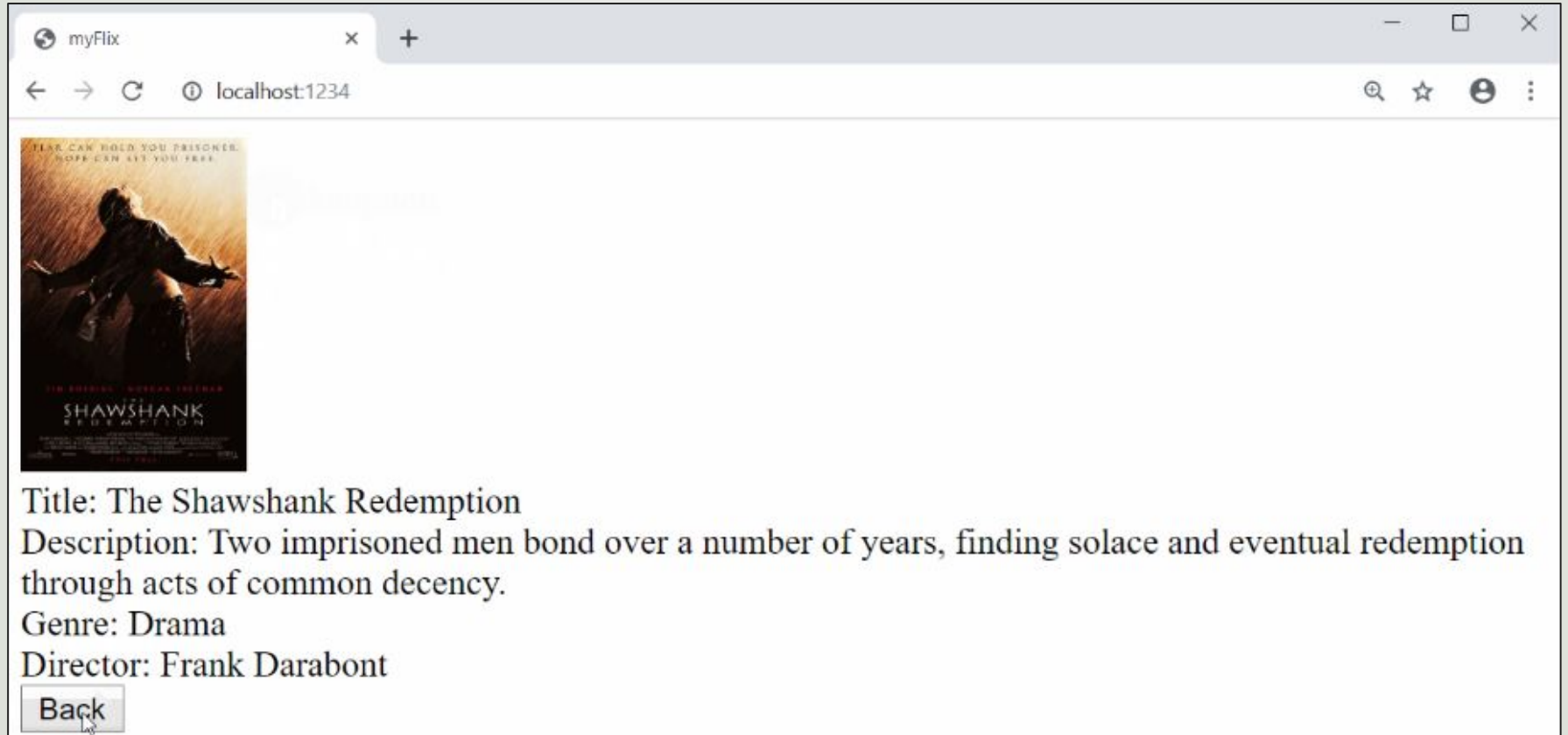
# **WHAT WAS THE GOAL**

Creating the three main React components that would form the web app using JSX and movies mock data (later to be 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 experimentations with the use of React state, props and event handling. It was therefore a good challenge to understand the mechanics behind these principles at first, but I was able to find several solutions by myself to assimilate everything and eventually deliver the all expected requirements. Some of the solutions I used to succeed in this step:

- 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 works well and what does not work well, and adapt my 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 of the project, ensuring a coherent directory set up following React industry standard.

```
myFlix
>Dist
>img
>node_modules
  V src
    V 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 my API and render it in the components

---

## Skills used

---

Research  
Problem solving  
Code writing  
Debugging



# WHY WAS THIS STEP IMPORTANT

Connecting the API I built previously to myFlix 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 fetched by the API) were rendered in the UI (using a handmade array written within one of the components for testing purposes).

# WHAT WAS THE GOAL

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

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

# MOVIE DATA FETCHED BEING STORED IN A MongoDB DATABASE

The screenshot displays the MongoDB Atlas web interface. On the left is a navigation sidebar with sections for DEPLOYMENT, Database, Data Lake, SERVICES, and SECURITY. The 'Database' section is active, showing a tree view with 'myFlixDB' expanded to reveal 'movies' and 'users' collections. The main panel shows the 'myFlixDB.movies' collection details, including storage and document statistics. Below this, there are tabs for Find, Indexes, Schema Anti-Patterns, Aggregation, and Search Indexes. The 'Find' tab is selected, showing a query filter bar with a placeholder query and buttons for Reset, Apply, and More Options. The document viewer below shows a single document for the movie 'Dracula' with fields for \_id, Title, Description, Genre, Director, ImagePath, and Featured.

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('648cc4cc9e8b6dbb7eae98b9'),
  "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

My 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 my React app at this point, I had to address this by temporarily adjusting one of my backend endpoints (/movies) and my CORS set up to allow the sending of some information on my front-end while I was working on it. I also had to make sure that my API data was being fed into the different components of my app correctly and that all the relation parent-child, useState, useEffect and props were working.

Also, at the first try, the images of my movies from my database were not displaying correctly. After investigation, I found that the link for each film image was not in the correct format. I made the necessary adjustments, updated my movie database again in MongoDB, and the images then displayed correctly.

```
<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 a web app.

# 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, SERVICES, and SECURITY. The DATABASE section is active, showing a tree view with 'myFlixDB' expanded and 'users' selected. The main panel shows the 'myFlixDB.users' collection with a storage size of 36KB and 34 documents. A filter bar at the top right includes 'VISUALIZE YOUR DATA' and 'REFRESH' buttons. Below the collection name, tabs for 'Find', 'Indexes', 'Schema Anti-Patterns', 'Aggregation', and 'Search Indexes' are visible. An 'INSERT DOCUMENT' button is located in the top right of the document list area. A filter bar with a search input 'Type a query: { field: 'value' }' and buttons for 'Reset', 'Apply', and 'More Options' is present. The document list shows a single document with the following fields: `_id`, `Username`, `Password`, `Email`, `Birthday`, and `FavoriteMovies`. The `FavoriteMovies` field is an array of three object IDs. At the bottom, navigation controls include 'PREVIOUS', '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 0 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 adjustment made on the backend (re-enabling token-based authentication on one of my API endpoints). I also needed to adjust the logic of some endpoints in my backend to ensure that it was impossible for a new user to create an account with an already existing username or email address. I later did several tests in pairs with my MongoDB database to validate the correct implementation of these backend updates.

# DECISIONS MADE

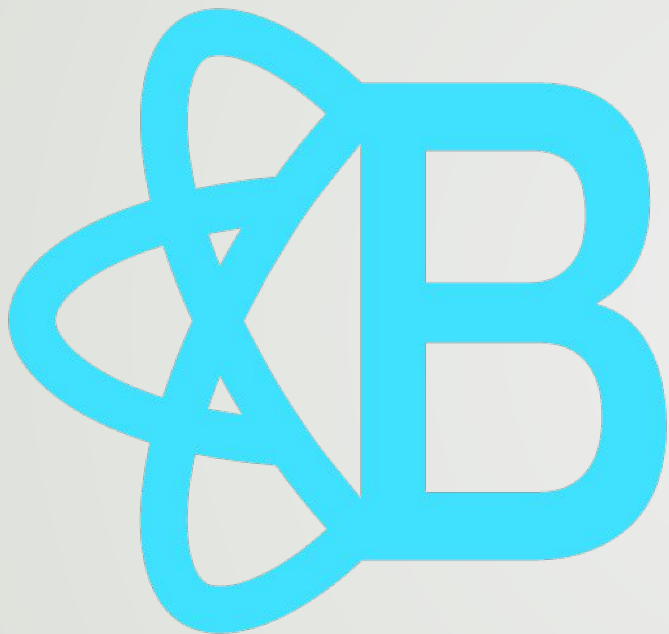
Even though it wasn't in the project requirements, I chose on my own to make additional changes to some endpoints in my API to ensure that each user has a unique username and email address, and so create a more realistic and functional web application.

Sign up status

✕

Testuser@hotmail.com already exists, please choose another email.

Close



**Styling** the web app using  
React Bootstrap features and  
components

---

## Skills used

---

Creative thinking  
User-first perspective  
Research  
Code writing

# WHY WAS THIS STEP IMPORTANT

Applying principles of good design to a React app using UI frameworks 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 it's responsive to use 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 app itself was particularly smooth.

The main challenge was more about ensuring that the design applied to the computer web version of myFlix was as visually appealing and logic for the smaller devices / mobile phones. Many thoughts and tests have been done on how to re-adjust each UI depending on the various screen scales to eventually select the ideal combinations.

# DECISIONS MADE

I choose 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 for a professional looking product.

```
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](#)



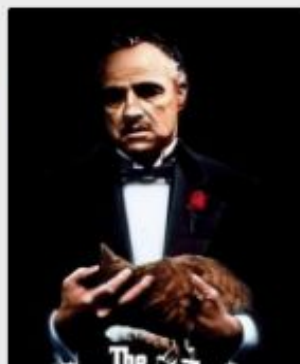
Gladiator  
Drama  
[See details](#)



Dracula  
Horror  
[See details](#)



Jurassic Park  
Science fiction  
[See details](#)



# SPECIFIC 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

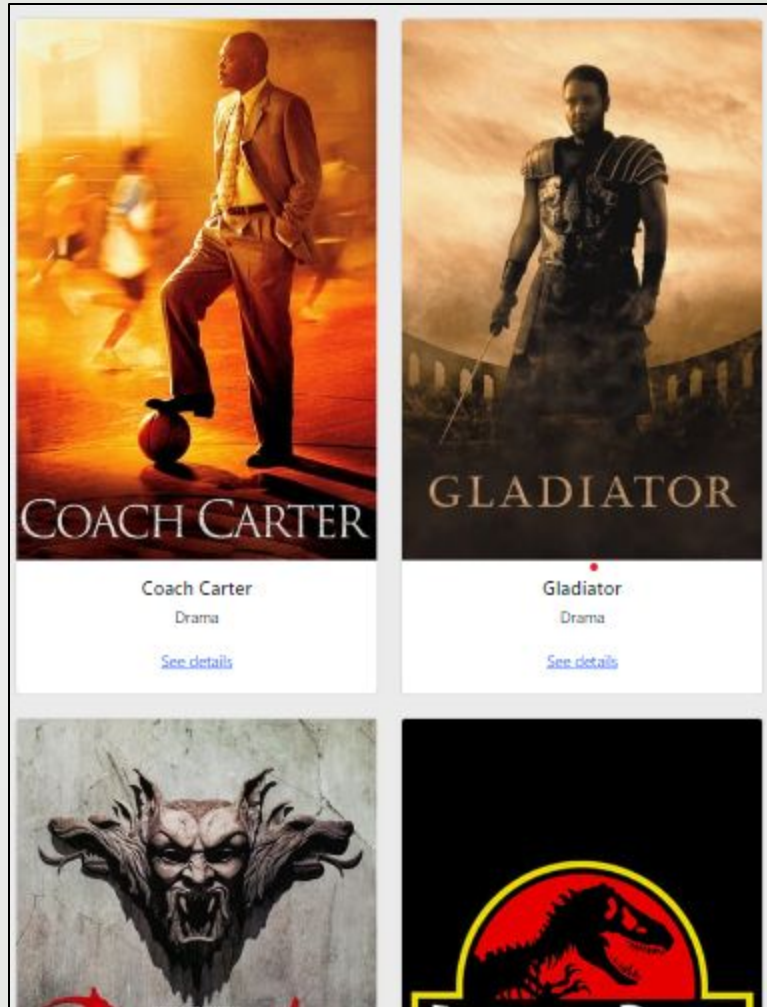
[Add to favorite](#)

[Back](#)



# MAIN VIEW

## MEDIUM SCREEN VISUAL



# MAIN VIEW

## SMALL SCREEN VISUAL

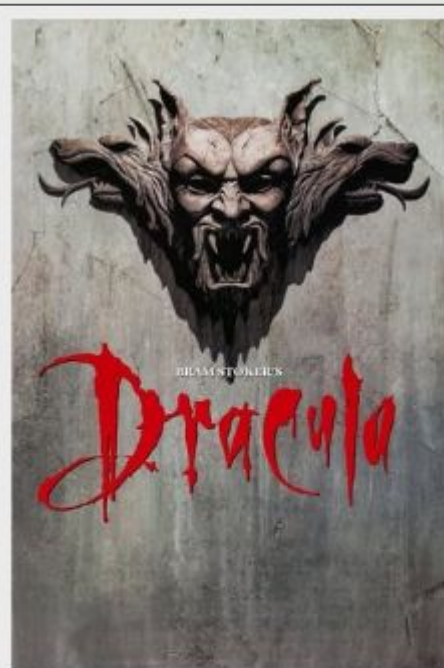




# SPECIFIC MOVIE VIEW

## 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, creating 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 favorite movie list. As for the specific features (movie filters and search bar), it made sense to implement those elements since users of this type of product generally expect this kind of functionality to facilitate their interaction.

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. The creation of a dynamic navigation bar for users to switch between different views was also part of this step.

This step was also meant to build 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. The codes to implement the filter movie button and the search bar was also meant to be developed at this point.

# NAVIGATION BAR - UNAUTHENTICATED USERS

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

## Welcome back!

Ready for your next marathon?

---

Username

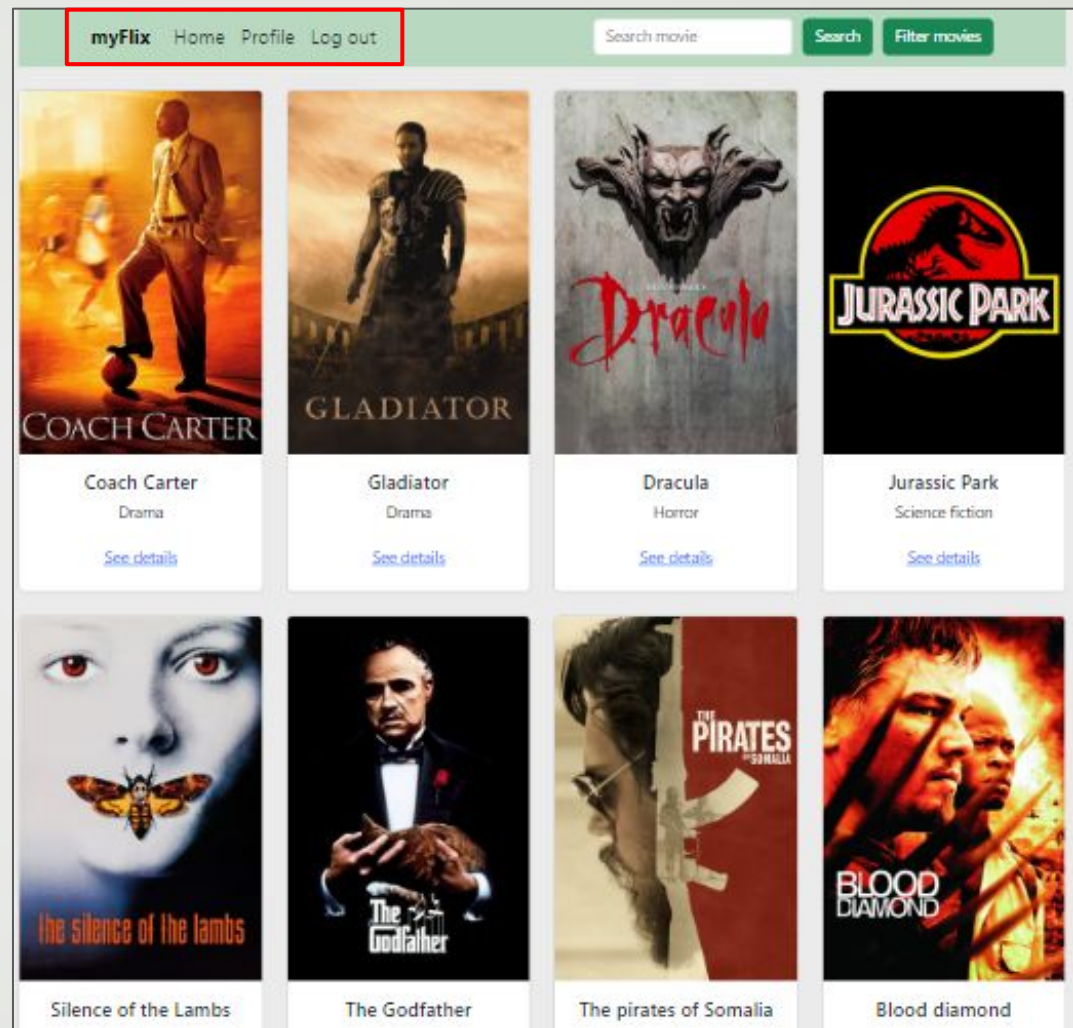
Username must be at least 5 characters long and contain only alphanumerical characters.

Password

Password can contain alphanumeric and non-alphanumeric characters.

[Log in](#)

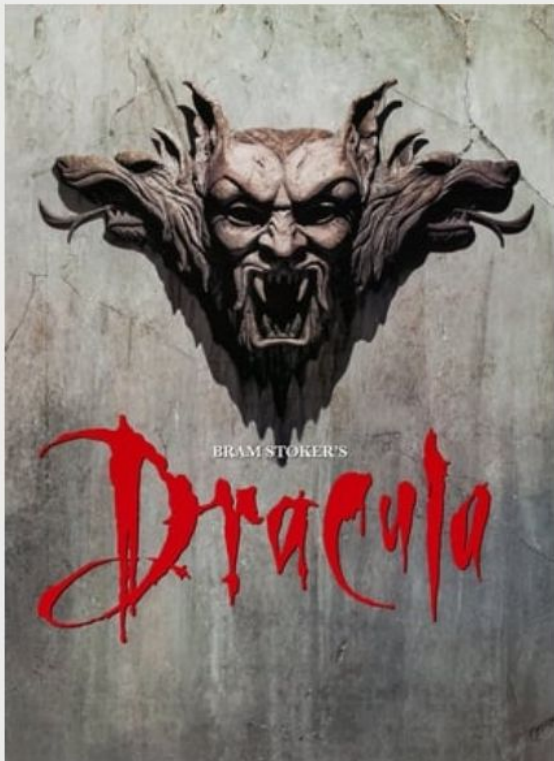
# NAVIGATION BAR AUTHENTICATED USERS



# STATE- BASED ROUTING

<https://myflix-movies-advisor.netlify.app/movies/648cc4cc9e8b6dbb7eae98b9>

myFlix Home Profile Log out



BRAM STOKER'S  
*Dracula*

**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.

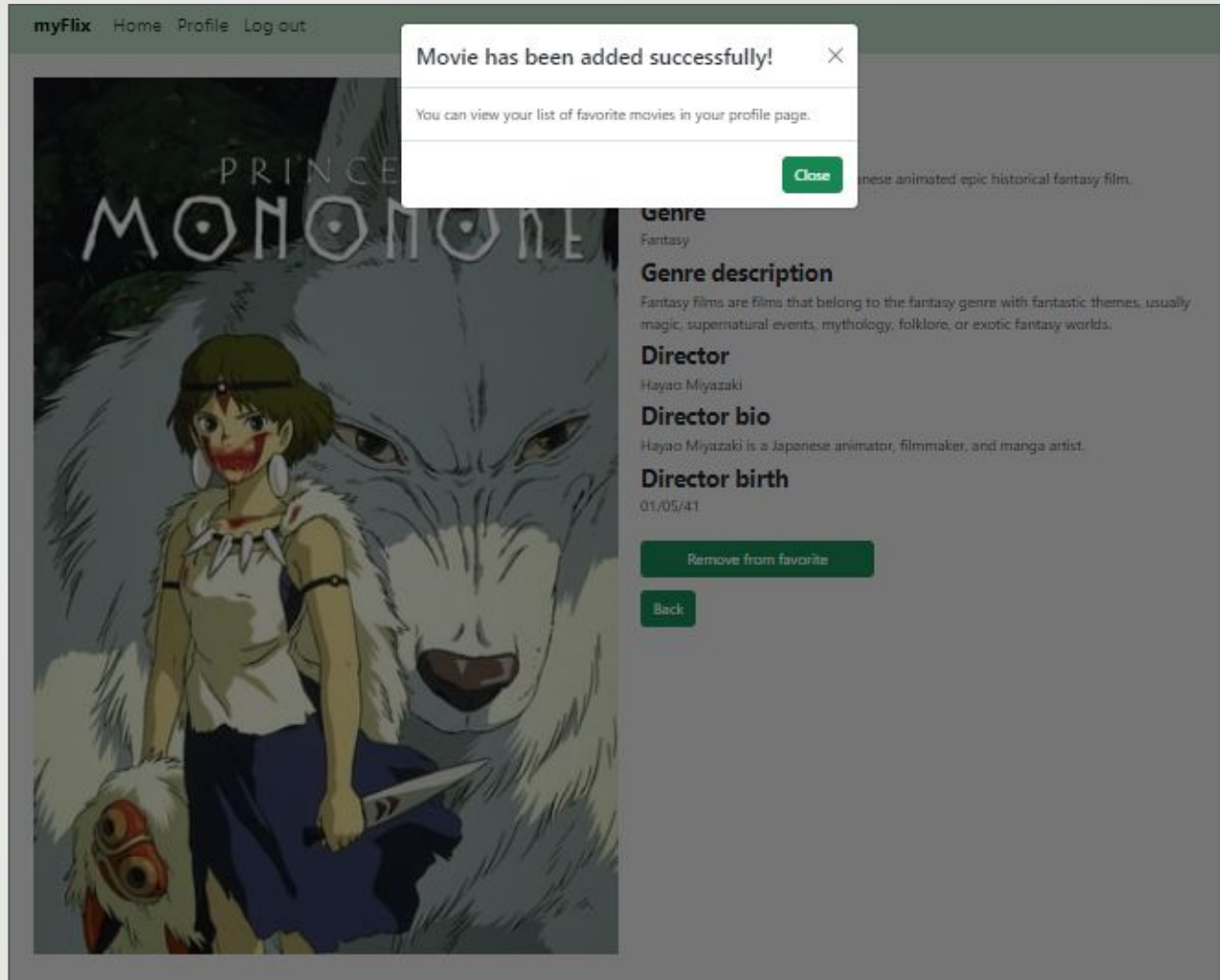
**Director birth**  
04/07/39

Add to favorite

Back



# MOVIE ADDED TO FAVORITE (SHOWN IN PROFILE VIEW)



# 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](#)



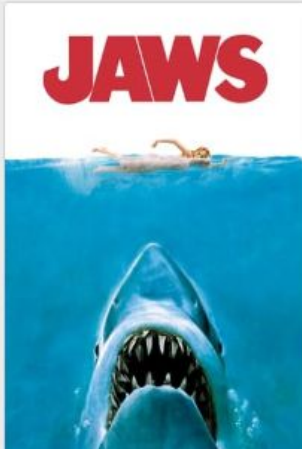
### Delete account

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

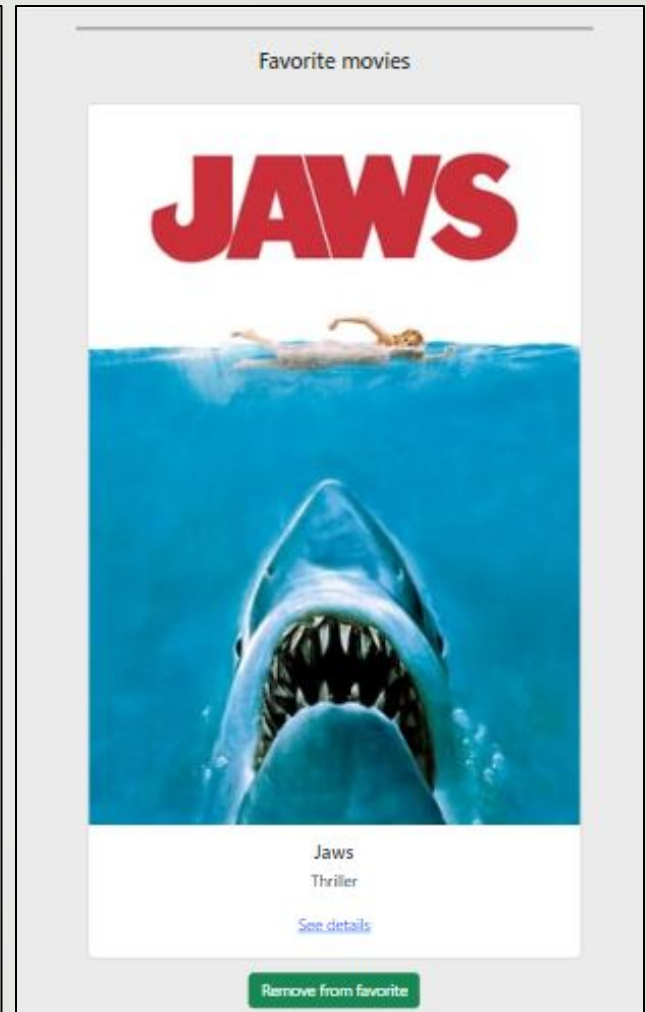
Delete account

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

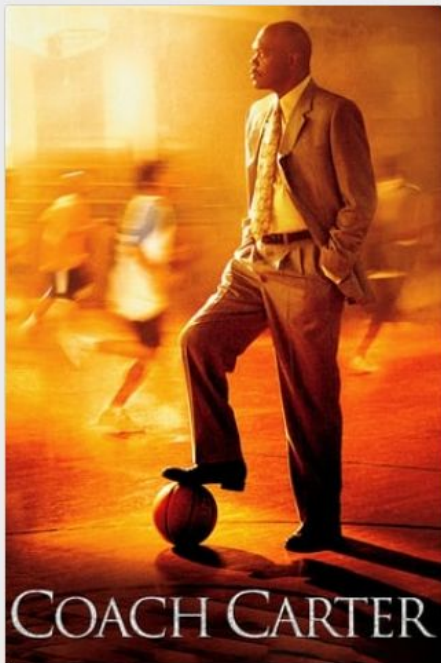


# FILTER BUTTON AND SEARCH BAR

myFlix Home Profile Log out

Search

Filter movies



Coach Carter

Drama

[See details](#)



GLADIATOR

Gladiator

Drama

[See details](#)



Dracula

Horror

[See details](#)

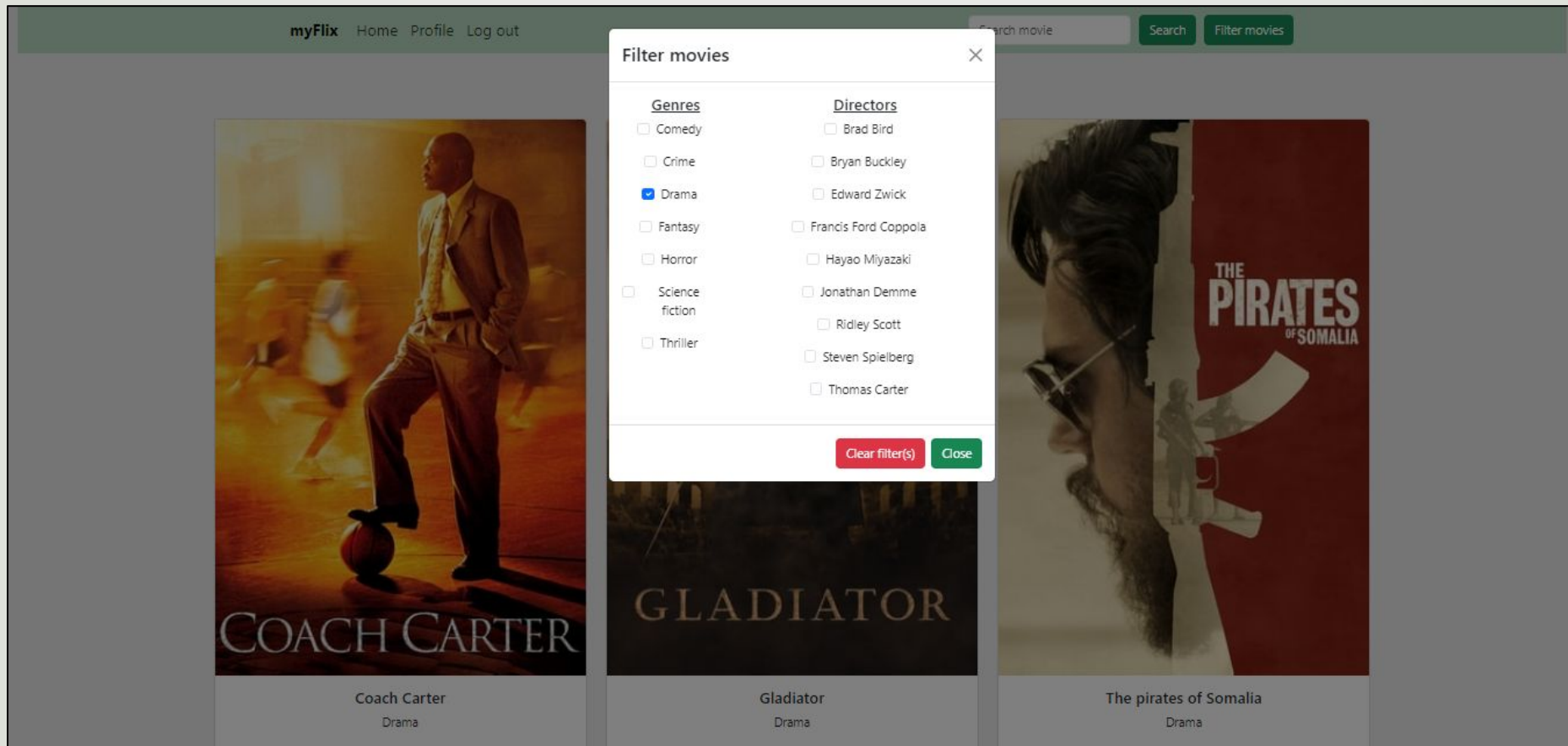


Jurassic Park

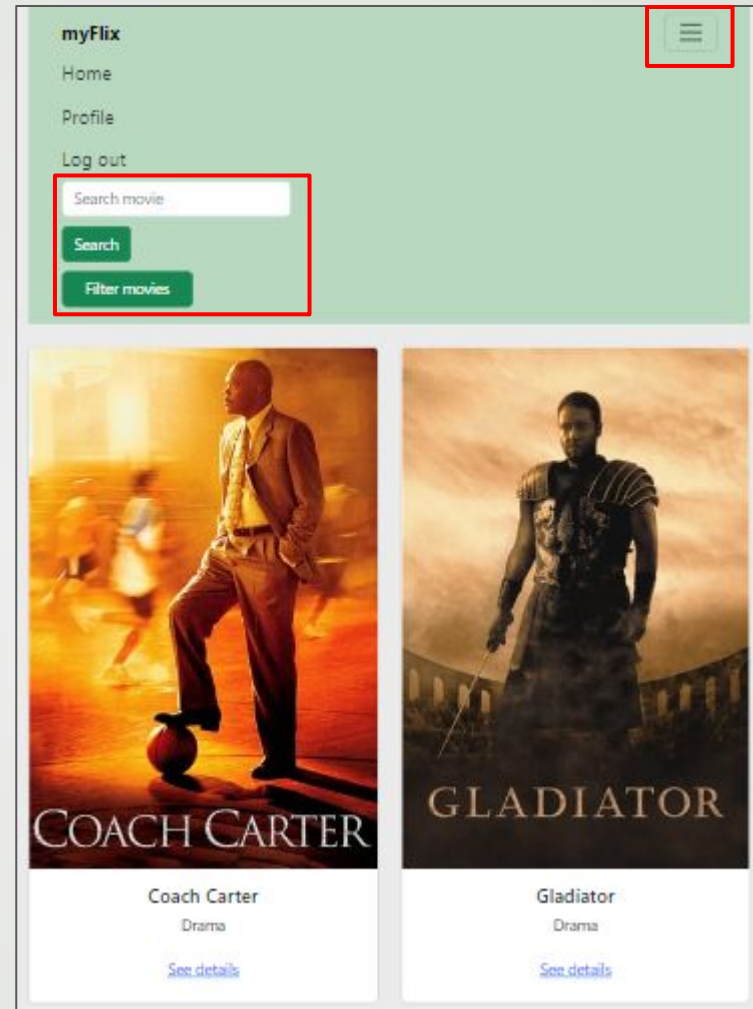
Science fiction

[See details](#)

# FILTER BUTTON MODAL



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



# CHALLENGES OR SPECIAL POINTS OF CONSIDERATION

In this step, I was also in my first experimentation with the use of react-router, react-router-dom and the implementation of state-based routing. It was therefore a good challenge to understand the mechanics behind these principles at first, but I was able to find several solutions by myself to assimilate everything and eventually deliver the all expected requirements. Some of the solutions I used to succeed in this step:

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

# DECISIONS MADE

I choose how to display the elements within the navigation bar, as well as the different parts composing the profile page. I also thought about the logic behind the movie filter function by asking myself: how could users generally expect to search for their movies using filters? I then implemented the logic accordingly within my code (filters based on genres and directors).

Search by...

Genres

Directors





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

---

Skills used

---

Critical thinking  
Detailed overview

# 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 its 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 modification, addition or adjustment to the web app.

Adding of comments and clarifications 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 functionalities of the application to ensure that it works perfectly.

Adjusting some final design details to ensure professional looking.



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

```
    /**Used to display each filtered movie on the UI (when filter parameter(s) is/are activated).
    const [filteredMovies, setFilteredMovies] = useState([]);
    /**Used to save the first list of all movie fetched, so it can be re-used when user clear filters to st
    const [originalMovies, setOriginalMovies] = useState([]);

    /**This useEffect filter the array of fetched movies based on selected genres and directors, and it up
    useEffect(() => {
        /**Filter movies based on selectedGenres and selectedDirectors.
        const filtered = movies.filter((movie) => {
            /**Check if a movie's genre is included in selectedGenres.
            const genreMatch = selectedGenres.length === 0 || selectedGenres.includes(movie.genre);
            /**Check if a movie's director is included in selectedDirectors.
            const directorMatch = selectedDirectors.length === 0 || selectedDirectors.includes(movie.director
            /**Return true if both genreMatch and directorMatch are true, meaning the movie matches the sel
            return genreMatch && directorMatch;
        });
        if (selectedGenres.length === 0 && selectedDirectors.length === 0) {
            /**If no filters are applied, show the whole list of movies.
            setFilteredMovies([]);
        } else {
            /**If filters are applied, set the filtered movies or an empty array if no movies match.
            setFilteredMovies(filtered);
        }
    }, [selectedGenres, selectedDirectors, movies]);

    /**Function the set everything back as their initial value (used for the 'clear filters' button).
    const resetFilters = () => {
        setSelectedGenres([]);
        setSelectedDirectors([]);
        setFilteredMovies([]);
```

# **CHALLENGES OR SPECIAL POINTS OF CONSIDERATION**

I positioned myself from the point of view of future colleagues who could work on my project. How can I make my 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 parts of my 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

---

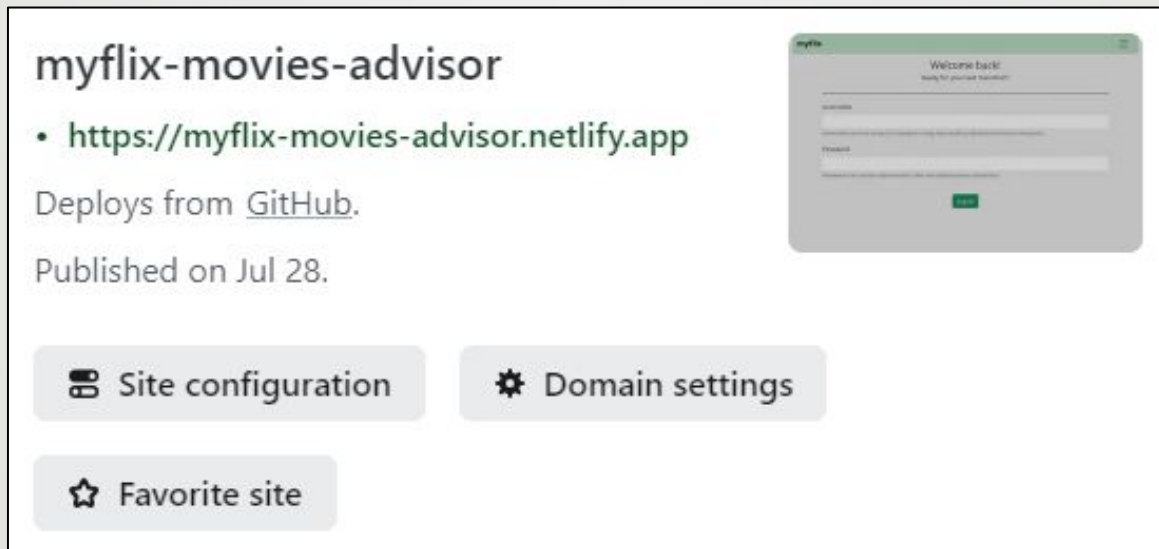
Research  
Debugging

# WHY WAS THIS STEP IMPORTANT

Ensure the web app is publicly available to everyone.

# WHAT WAS THE GOAL

Deploying and hosting myFlix online on Netlify.



# CHALLENGES OR SPECIAL POINTS OF CONSIDERATION

At first, my app would not render properly in the web browser, despite a deployment that seemed successful on Netlify. I had to identify the source of the problem by consulting the Netlify documentation. I ran several tests to troubleshoot the problem 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 my myFlix Github repository. The goal was to ensure that all relevant information regarding myFlix is accessible under these four categories:

- Project description
- User interface
- API and React
- 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 several times. I get inspired by 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

