

# myFlix web app

Angular frontend  
development  
process



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

# TABLE OF CONTENTS

---

**01**

---

Analyzing the project user stories, setting up a Kanban board and creating the web app user flow

---

**02**

---

Creating the project using Angular CLI, configuring the files and creating an Angular service file to connect with the API

---

**03**

---

Creating user registration and user log in components / UIs

---

**04**

---

Implementing the web app main view and profile page (for logged-in users), and routing between pages

---

**05**

---

Final code revision, testing, and design last adjustments

---

**06**

---

Deploying the app

# TABLE OF CONTENTS

---

**07**

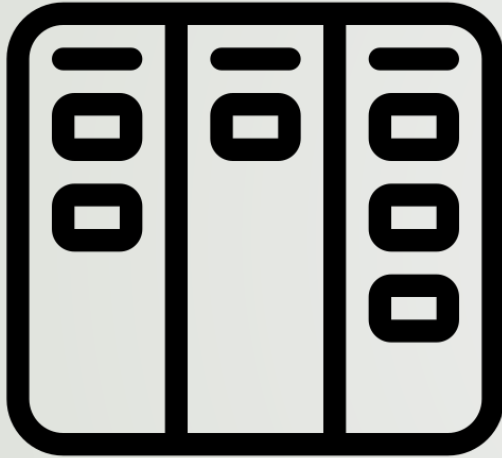
---

Creating all the necessary  
documentation

---

# 01

---



**Analyzing** the project user stories, **setting up** a Kanban board and **creating** the web app user flow

---

## Skills used

---

Project management  
Analytical thinking  
Visual communication (schemas)

# WHY WAS THIS STEP IMPORTANT

Analyzing the user stories is essential to pinpoint the exact features that need to be implemented in a product, and thus fulfil stakeholders and users expectations.

Knowing how to work with Kanban is also important, since it's a project management method widely used in various work environments. Kanban is interesting since it's based on simple visuals and allows tasks to be presented in a clear and effective way, which makes it a useful project management and monitoring tool, especially in coding.



Jira Software

# WHAT WAS THE GOAL

Analyzing each user story (2) and note down the technical requirements and functionalities needed to meet the desired results.


Migrating the technical requirements extracted from the project's user stories into Kanban cards / tasks (in Jira) to structure and align the project over the development (see images on next slides).

- Adding story points to each card based on the expected effort required.

Creating the users flow on myFlix Angular (see images on next slides).

- Showing the steps users have to complete to achieve their end goal while using the web app.

# KANBAN BOARD AT THE BEGINNING OF THE PROJECT

 Jira Software

Your work ▾

Projects ▾


Filters ▾





Dashboards ▾


Teams ▾

Apps ▾

Create

 Search


   


 Achievement 6 - Angular...  
Software project

You're on the Free plan


UPGRADE


PLANNING


 Timeline


 Board

DEVELOPMENT

 Code

 Project pages

 Add shortcut


 Project settings


You're in a team-managed project


Learn more

Projects / Achievement 6 - Angular project

myFlix - Angular - Board



 AC



Epic ▾

GROUP BY 


None ▾

Insights

TO DO 23

FRONTEND - Creating a landing page where all available movies are displayed once users are logged-in.

USER STORY 1 - AS A USER, I WANT TO B...


☒ A6AP-3 

FRONTEND - Adding basic information for each movie on the landing page such as picture, title and genre (so users can know what they are looking at and choose accordingly).

USER STORY 1 - AS A USER, I WANT TO B...

IN PROGRESS

DONE ✓







# KANBAN BOARD IN PROGRESS

Jira Software

Your work ▾

Projects ▾

Filters ▾

Dashboards ▾

Teams ▾

Apps ▾

Create

Search



Achievement 6 - Angular...  
Software project

You're on the Free plan

UPGRADE

PLANNING

Timeline

Board

DEVELOPMENT

Code

Project pages

Add shortcut

Project settings

You're in a team-managed project

Learn more

Projects / Achievement 6 - Angular project

## myFlix - Angular - Board

Search



Epic ▾

GROUP BY

None ▾

Insights

TO DO 19

FRONTEND - Creating a landing page where all available movies are displayed once users are logged-in.

USER STORY 1 - AS A USER, I WANT TO B...

✓ A6AP-3



FRONTEND - Adding basic information for each movie on the landing page such as picture, title and genre (so users can know what they are looking at and choose accordingly).

USER STORY 1 - AS A USER, I WANT TO B...

IN PROGRESS 4

BACKEND - Creating an API endpoint call to GET all the movies for the landing page.

USER STORY 1 - AS A USER, I WANT TO B...

✓ A6AP-20



BACKEND - Creating an API endpoint call to GET specific detailed on each specific movie, based on movie title (when user clicked on a movie on the landing page).

USER STORY 1 - AS A USER, I WANT TO B...

✓ A6AP-21




DONE ✓



# KANBAN TIMELINE IN PROGRESS

## Timeline

	SEP
	18 19 20 21 22
▼  A6AP-1 User story 1 - As a user, I want to be able to receive information on movies, directors, and genres so that I can learn more a...	
<input checked="" type="checkbox"/> A6AP-3 FRONTEND - Creating a landing page where all available movies are displayed once users are logged-in.	TO DO AC
<input checked="" type="checkbox"/> A6AP-4 FRONTEND - Adding basic information for each movie on the landing page such as picture, title and genre...	TO DO AC
<input checked="" type="checkbox"/> A6AP-10 FRONTEND - Making all movies from the landing page clickable in order to, once clicked, take users to a si...	TO DO AC
<input checked="" type="checkbox"/> A6AP-11 FRONTEND - For each single movie view, show a list of more detailed information on the selected movi...	TO DO AC
<input checked="" type="checkbox"/> A6AP-15 FRONTEND - Ensuring a clear and simple app visual for these UIs so users can easily find the information t...	TO DO AC
<input checked="" type="checkbox"/> A6AP-14 FRONTEND - Implementing responsiveness in the app landing page and single movie view, as users might...	TO DO AC
<input checked="" type="checkbox"/> A6AP-20 BACKEND - Creating an API endpoint call to GET all the movies for the landing page.	IN PROGRESS AC
<input checked="" type="checkbox"/> A6AP-21 BACKEND - Creating an API endpoint call to GET specific detailed on each specific movie, based on ..	IN PROGRESS AC
<input checked="" type="checkbox"/> A6AP-22 BACKEND - Creating an API endpoint call to GET specific details on a director, based on director na...	IN PROGRESS AC
<input checked="" type="checkbox"/> A6AP-23 BACKEND - Creating an API endpoint call to GET specific details on a genre, based on genre title (wh...	IN PROGRESS AC

# KANBAN BOARD IN PROGRESS

Jira Software

Your work ▾

Projects ▾

Filters ▾

Dashboards ▾

Teams ▾

Apps ▾

Create

Search



Achievement 6 - Angular...

Software project

You're on the Free plan

UPGRADE

PLANNING



Timeline



Board

DEVELOPMENT



Code



Project pages



Add shortcut



Project settings

You're in a team-managed project

Learn more

Projects / Achievement 6 - Angular project

## myFlix - Angular - Board



Epic ▾

GROUP BY

None ▾

Insights

TO DO 15

FRONTEND - Ensuring a clear and simple app visual for these UIs so users can easily find the information they are looking for.

USER STORY 1 - AS A USER, I WANT TO B...

✓ A6AP-15



FRONTEND - Implementing responsiveness in the app landing page and single movie view, as users might access it through computer or mobile.

USER STORY 1 - AS A USER, I WANT TO B...

✓ A6AP-14



IN PROGRESS 4

FRONTEND - Creating a landing page where all available movies are displayed once users are logged-in.

USER STORY 1 - AS A USER, I WANT TO B...

✓ A6AP-3



FRONTEND - Making all movies from the landing page clickable in order to, once clicked, take users to a single movie view where additional movie details are displayed.

USER STORY 1 - AS A USER, I WANT TO B...

DONE 4 ✓

BACKEND - Creating an API endpoint call to GET all the movies for the landing page.

USER STORY 1 - AS A USER, I WANT TO B...

✓ A6AP-20



BACKEND - Creating an API endpoint call to GET specific detailed on each specific movie, based on movie title (when user clicked on a movie on the landing page).

USER STORY 1 - AS A USER, I WANT TO B...

✓ A6AP-21



# KANBAN TIMELINE IN PROGRESS

Jira Software

Your work ▾

Projects ▾

Filters ▾

Dashboards ▾

Teams ▾

Apps ▾

Create

Search



Achievement 6 - Angular...

Software project

You're on the Free plan

UPGRADE

PLANNING



Timeline



Board

DEVELOPMENT



Code



Project pages



Add shortcut



Project settings

You're in a team-managed project

Learn more

Projects / Achievement 6 - Angular project

## Timeline

Give feedback

Share

Export ...

Search



Status category ▾

Epic ▾



- ▼ ⚡ A6AP-1 User story 1 - As a user, I want to be able to receive information on movies, directors, and genres so that I can learn more a...
- ✓ A6AP-3 FRONTEND - Creating a landing page where all available movies are displayed once users are logged... IN PROGRESS AC
- ✓ A6AP-10 FRONTEND - Making all movies from the landing page clickable in order to, once clicked, take users... IN PROGRESS AC
- ✓ A6AP-4 FRONTEND - Adding basic information for each movie on the landing page such as picture, title and... IN PROGRESS AC
- ✓ A6AP-11 FRONTEND - For each single movie view, show a list of more detailed information on the selected... IN PROGRESS AC
- ✓ A6AP-15 FRONTEND - Ensuring a clear and simple app visual for these UIs so users can easily find the information t... TO DO AC
- ✓ A6AP-14 FRONTEND - Implementing responsiveness in the app landing page and single movie view, as users might... TO DO AC
- ✓ A6AP-20 BACKEND - Creating an API endpoint call to GET all the movies for the landing page. DONE AC
- ✓ A6AP-21 BACKEND - Creating an API endpoint call to GET specific detailed on each specific movie, based on movie t... DONE AC
- ✓ A6AP-22 BACKEND - Creating an API endpoint call to GET specific details on a director, based on director name (wh... DONE AC
- ✓ A6AP-23 BACKEND - Creating an API endpoint call to GET specific details on a genre, based on genre title (when use... DONE AC

# User flow 1

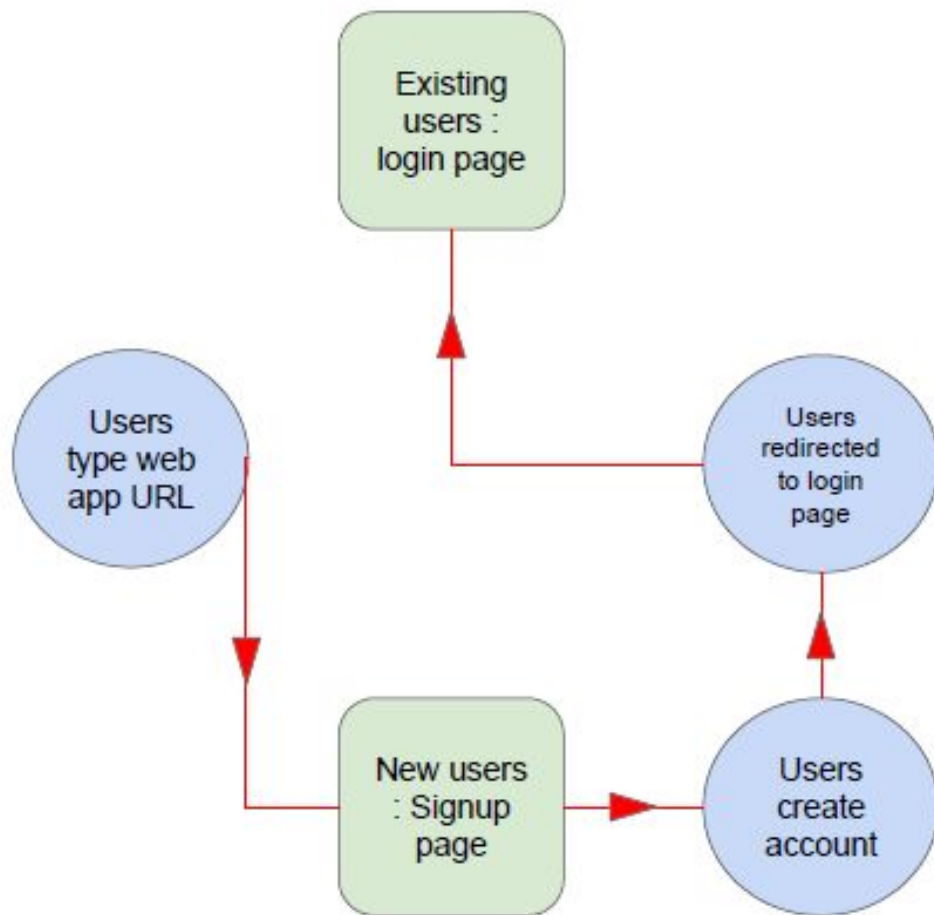
*As a user, I want to be able to create a profile so I can save data about my favorite movies.*

## Legend

Users  
action

App UI

Current user  
flow





## User flow 2

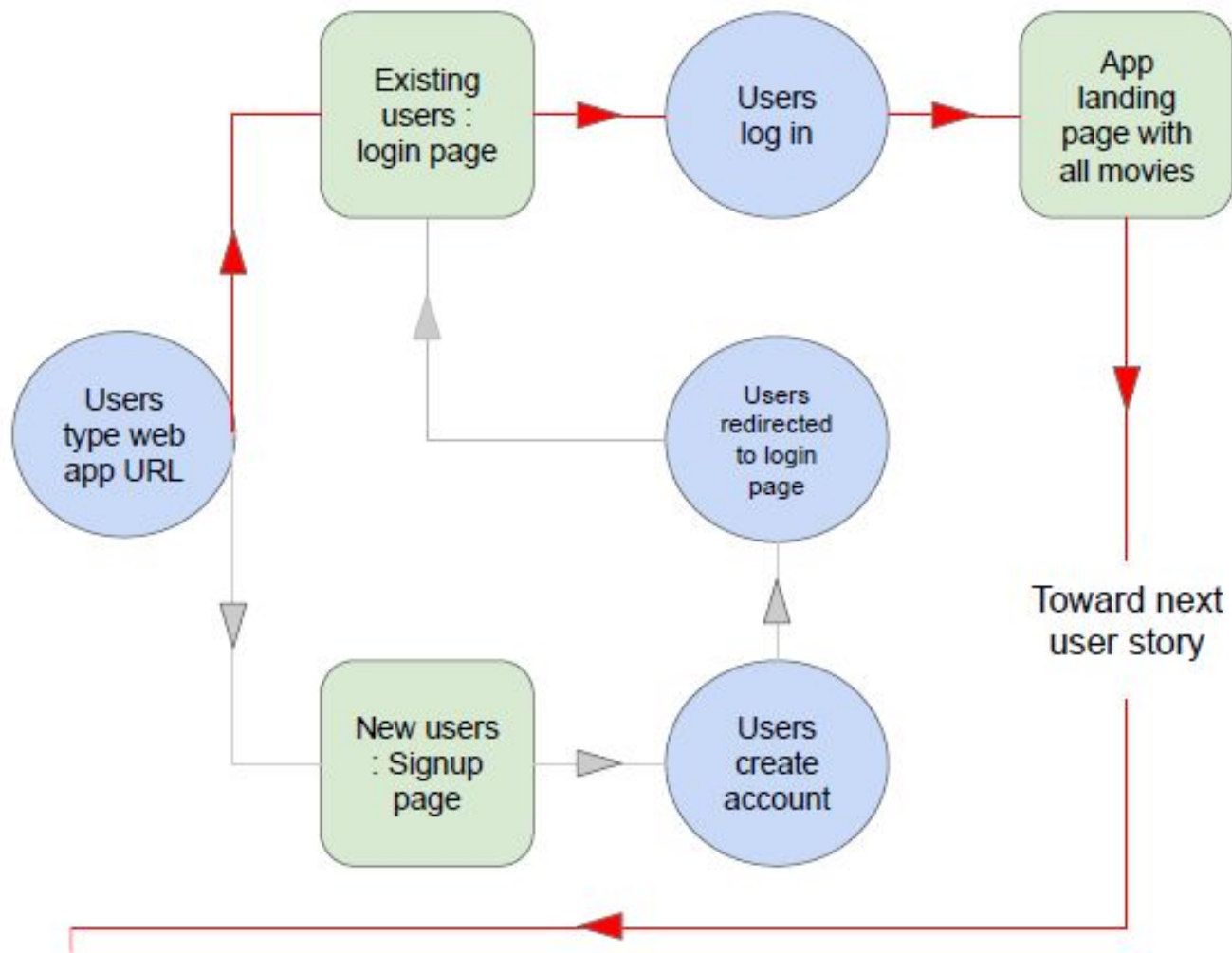
*As a user, I want to be able to create a profile so I can save data about my favorite movies.*

### Legend

Users action

App UI

Current user flow



## User flow 3

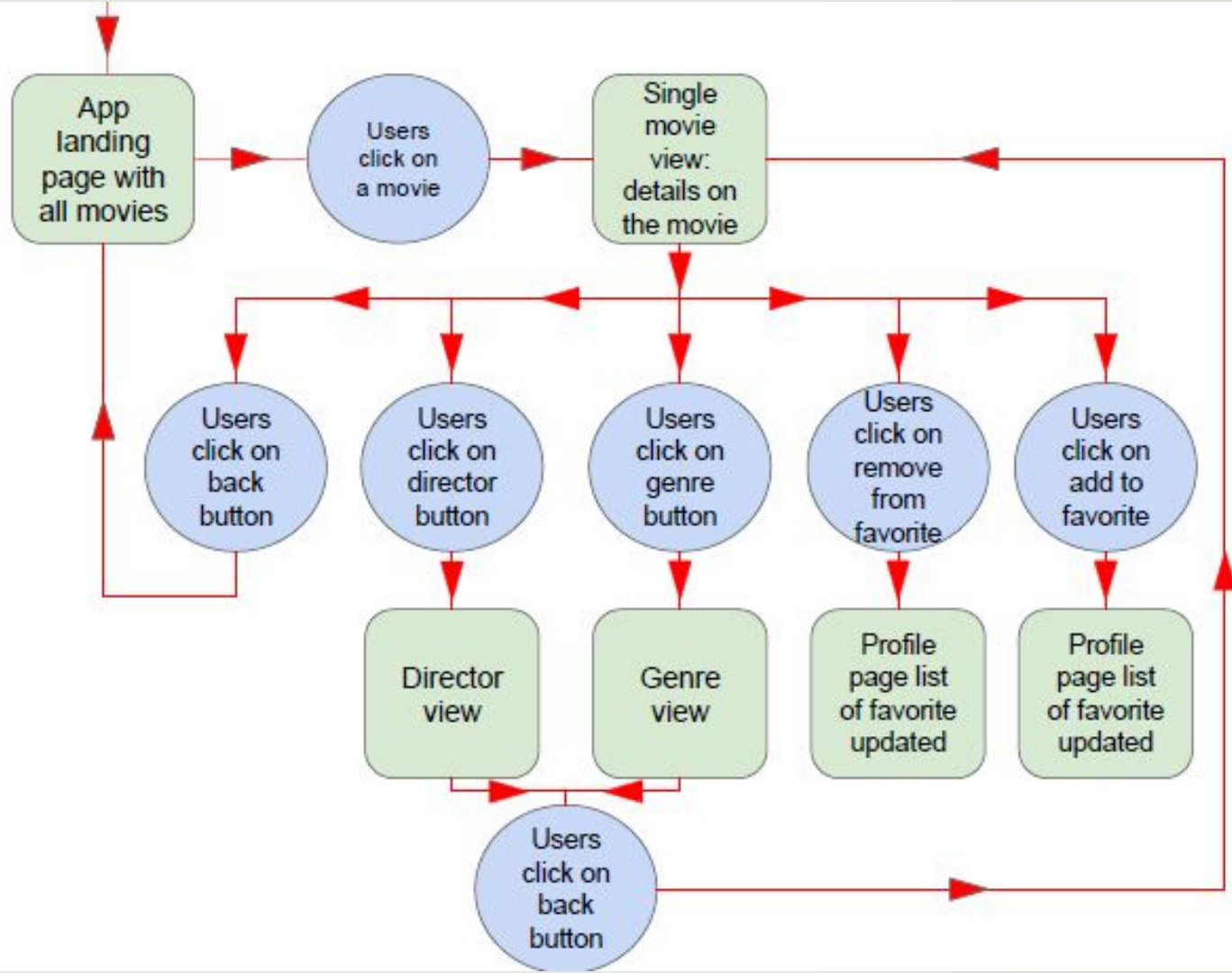
*As a user, I want to be able to receive information on movies, directors, and genres so that I can learn more about movies I've watched or am interested in.*

### Legend

Users action

App UI

Current user flow



# CHALLENGES OR SPECIAL POINTS OF CONSIDERATION

This first step was familiar to me, since I already had good experience with project management softwares (e.g.: Asana) as well as analyzing user stories.

# DECISION MADE

I chose the project management tool to be used for the project (in this case - Jira). I also decided how to formulate and present the tasks for the projects in the Jira's cards, and added their story points.



---

# 02

---



**Creating** the project using Angular CLI, **configuring** the files and **creating** an Angular service file to connect with the API

---

## Skills used

---

Research  
Problem solving  
Debugging

# **WHY** WAS THIS STEP IMPORTANT

Organizing the development environment for efficient and well-oriented work from the beginning is one of the keys to ensure smooth workflow and limit avoidable time loss due to inefficient project initialization.

As for the Angular service, it was essential to create it to ensure communication between the frontend and the backend of the web app.

# WHAT WAS THE GOAL

Creating the project by running `$ ng new myFlix-Angular-client` in the Angular CLI (similar to running `create-react-app` in React).

Setting up and configuring Angular HttpClient (with Angular *HttpClientModule*) to ensure the web app is able to communicate and fetch data from the movie API.

Writing codes to load the data from the API:

- In the newly *fetch-api-data.service.ts* created, I've added the necessary imports (*HttpClient*, *HttpHeaders*, *HttpErrorResponse*, etc.).
- I then implemented in this same file the codes used to make the calls from myFlix frontend to each endpoint in the backend API (see examples on the next slides).

## CODE CALLING THE ENDPOINT ALLOWING TO GET ALL MOVIES

```
getAllMovies(): Observable<any> {  
  const token = localStorage.getItem('token');  
  return this.http.get(apiUrl + 'movies', {  
    headers: new HttpHeaders(  
      {  
        Authorization: 'Bearer ' + token,  
      }  
    ))  
  }).pipe(  
    map(this.extractResponseData),  
    catchError(this.handleError)  
  );  
}
```

## CODE CALLING THE ENDPOINT ALLOWING TO ADD FAVORITE MOVIES

```
userAddFavoriteMovie(Uusername: string, MovieID: string): Observable<any> {  
  const token = localStorage.getItem('token');  
  const headers = new HttpHeaders({  
    Authorization: 'Bearer ' + token,  
  });  
  const requestBody = {};  
  return this.http.post(apiUrl + 'users/' + Uusername + '/movies/' + MovieID, requestBody, { headers })  
    .pipe(  
      map(this.extractResponseData),  
      catchError(this.handleError)  
    );  
}
```

# CHALLENGES OR SPECIAL POINTS OF CONSIDERATION

At some point, I realized that there was a missing endpoint in the backend, considering the list of calls from the frontend that needed to be implemented. It was therefore impossible to call this specific missing endpoint from the frontend.

To fix this, I went back to the backend code, and I created a new endpoint. I carried out various tests to ensure that it was working properly. I was then able to connect my frontend service to this new endpoint, thus ensuring to keep meeting all the project requirements.



**Creating** user registration  
and user log in components /  
UIs

---

## Skills used

---

Research  
Problem solving  
Code writing  
Debugging

# WHY WAS THIS STEP IMPORTANT

Implementing and rendering a sign up and a log in logic / view was important to ensure users have a mean to access the content displayed inside the web app (and reserved for authenticated users).

# WHAT WAS THE GOAL

Creating a sign up and log in component using *\$ ng generate component user-registration-form* and *\$ ng generate component user-login-form*.

Implementing the codes in the newly TypeScript (.ts) file created for the sign up and log in process respectively.

(suite on next slide)



# WHAT WAS THE GOAL (SUITE)

Updating the HTML file for the sign up and log in respectively, to ensure a simple and clear view is rendered in users' welcome page for both components.

- Angular's binding syntax and directives, such as *NgModel*, have also been added at this point in the HTMLs to establish communication between the TypeScript files and the HTML files.

Polishing up the UIs for the welcome page / sign up view / log in view with Angular Material.

- For example, sign up form and log in form in UIs are shown using a dialog box from Angular Material (similar to a modal in Bootstrap) (see next images).

Running tests to ensure sign up and log in process are working as expected and that, once logged-in, current user and related token are stored correctly in localStorage.

# FIRST RENDERED VIEW - WELCOME PAGE (NO STYLING)

Welcome to myFlix!

Sign Up

Login

# FIRST RENDERED VIEW - SIGN UP FORM (NO STYLING)

Welcome to myFlix!

Sign Up

Login

Sign Up!

Username

Password

Email

mm/dd/yyyy

Sign Up

# FIRST RENDERED VIEW - LOG IN FORM (NO STYLING)

Welcome to myFlix!

Sign Up

Login

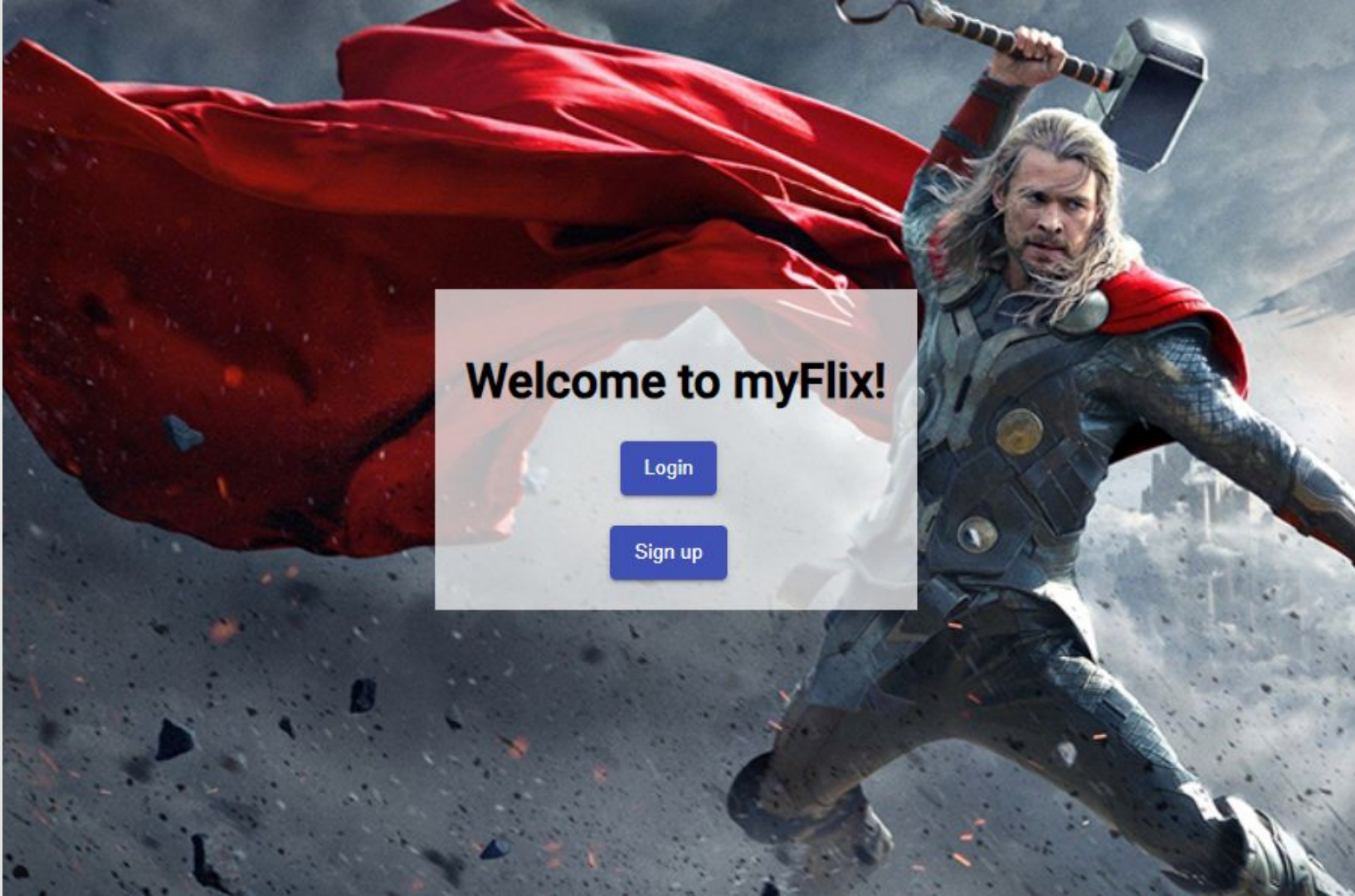
Log in!

Username

Password

Log in

**WELCOME  
PAGE**  
(WITH  
STYLING)



**Welcome to myFlix!**

Login

Sign up

# SIGN UP FORM (WITH STYLING)

Sign up!

Username

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

Password

Password can contain alphanumeric and  
non-alphanumeric characters.

Email

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

Birthday

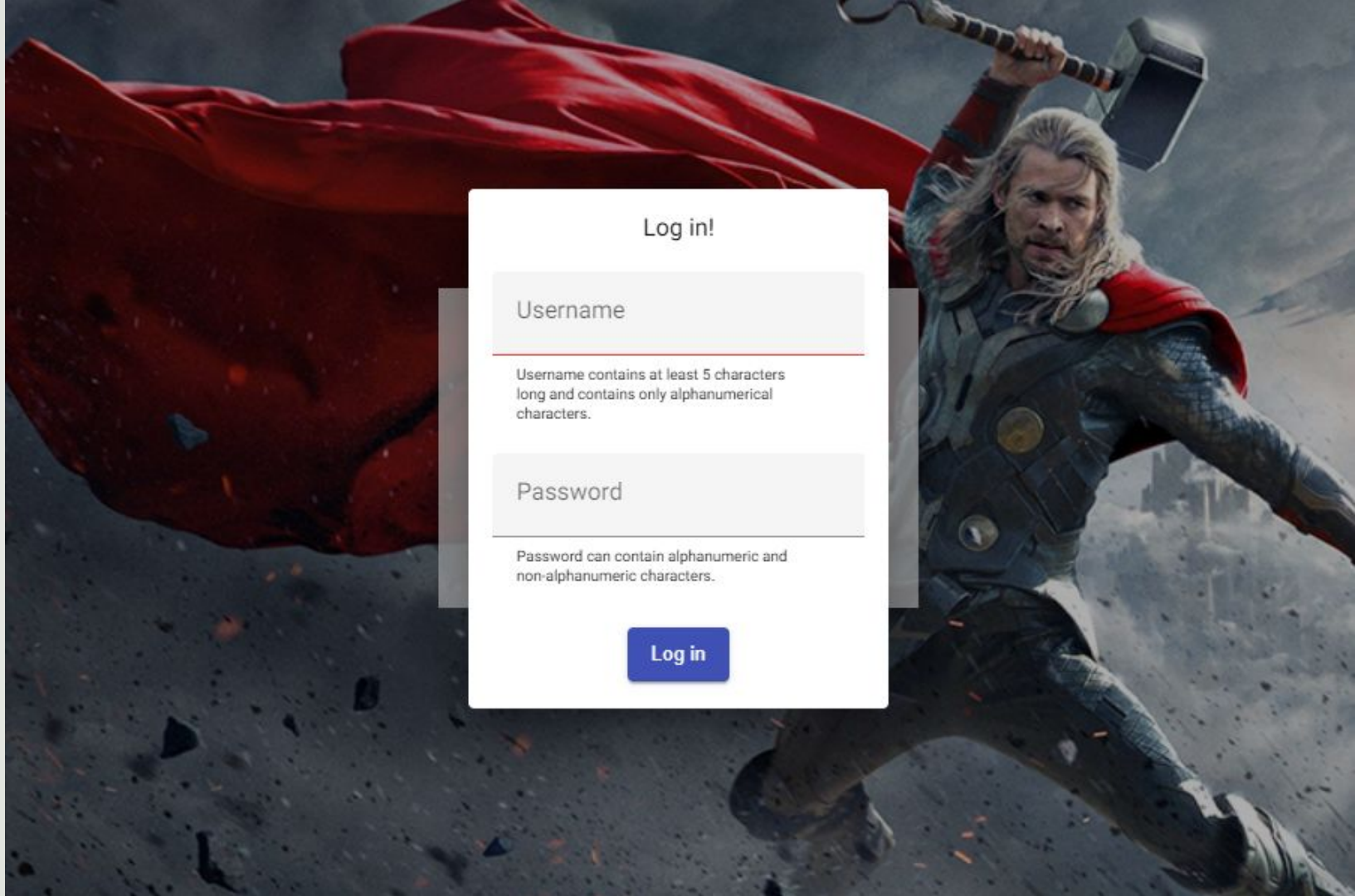


Birthday is optional. If you want to add it,  
pick your birthday from the calendar icon.

Sign up



# LOG IN FORM (WITH STYLING)



Log in!

Username

Username contains at least 5 characters long and contains only alphanumerical characters.

Password

Password can contain alphanumeric and non-alphanumeric characters.

Log in



**Implementing** the web app  
main view and profile page  
(for logged-in users), and  
routing between pages

---

## Skills used

---

Research  
Problem solving  
Code writing  
Debugging



# WHY WAS THIS STEP IMPORTANT

This step was crucial for implementing the actual core codes and views of the web app. Without it, myFlix wouldn't have had any movies to display and no profile page on which users could have checked their information and updated it, making the final product pointless.

`movie-cards`

`director-details`

`genre-details`

`synopsis-details`

`user-profile`

# WHAT WAS THE GOAL

Creating the main view (displaying all movie cards) on which users land after logging in successfully. To do so, I:

- Created the main view component
- Wrote codes in the main view TypeScript file to fetch the movies from the API
- Wrote codes in the main view HTML file to display the movies fetched from the API
- Added an action bar below each movie card with different links to display, in dialogs and based on users' interaction, movie director details, movie genre details and movie synopsis details. A favorite button has also been added (see images on next slides)
- Ensured responsiveness and consistent styling / color palette with Angular Material for a great experience on any device sizes

Creating the navigation bar at the top of the website. To do so, I:

- Added the myFlix logo
- Listed all available pages to navigate
- Added the logout function

# FIRST RENDERED VIEW - MAIN VIEW ON LARGE SCREEN (NO STYLING)

myFlix

Menu

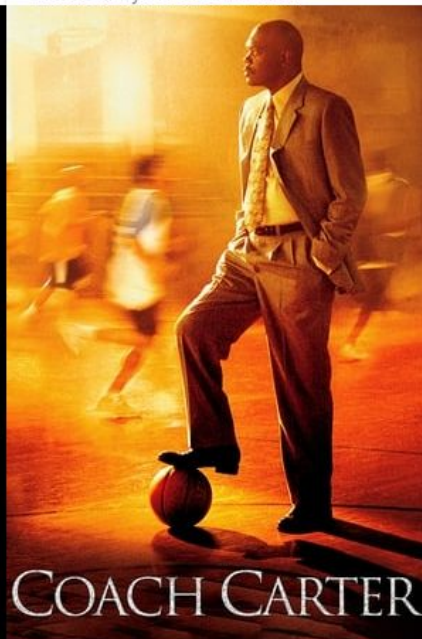
Jaws  
Directed by: Steven Spielberg



Jurassic Park  
Directed by: Steven Spielberg



Coach Carter  
Directed by: Thomas Carter



Gladiator  
Directed by: Ridley Scott



Silence of the Lambs  
Directed by: Jonathan Demme



[Genre](#) [Director](#) [Synopsis](#) [♥](#)

[Genre](#) [Director](#) [Synopsis](#) [♥](#)

[Genre](#) [Director](#) [Synopsis](#) [♥](#)

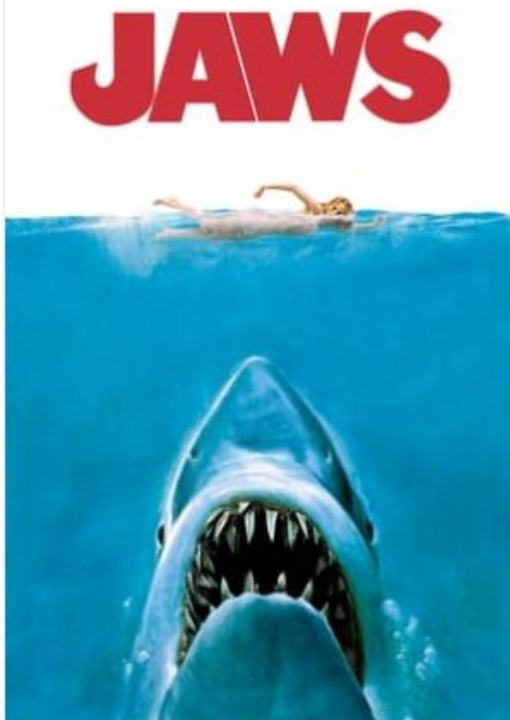
[Genre](#) [Director](#) [Synopsis](#) [♥](#)

[Genre](#) [Director](#) [Synopsis](#) [♥](#)

MAIN  
VIEW ON  
LARGE  
SCREEN  
(WITH  
STYLING)

### Jaws

Directed by: Steven Spielberg



Genre

Director

Synopsis



### Jurassic Park

Directed by: Steven Spielberg



Genre

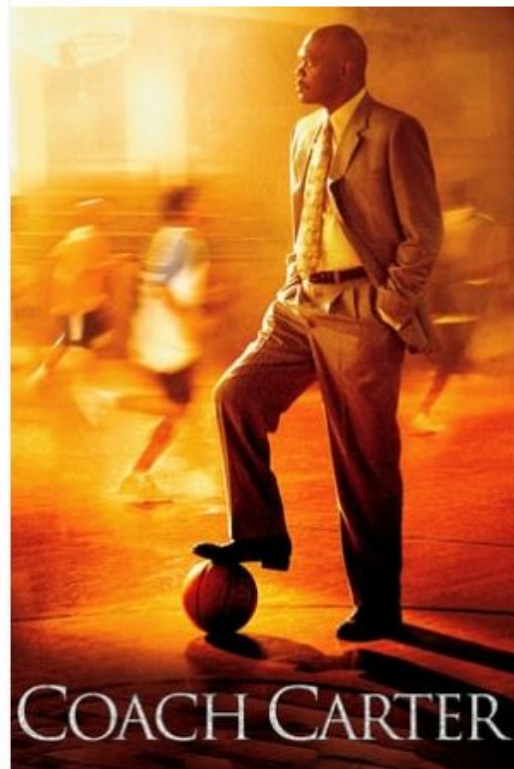
Director

Synopsis



### Coach Carter

Directed by: Thomas Carter



Genre

Director

Synopsis





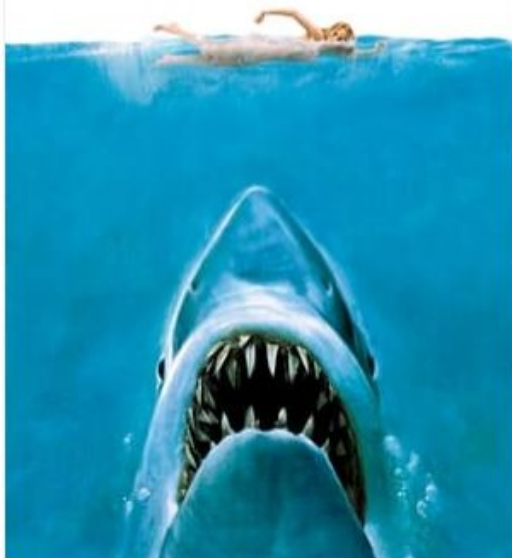
Home

Profile

Logout

Jaws

Directed by: Steven Spielberg

**JAWS**

Genre

Director

Synopsis



Jurassic Park

Directed by: Steven Spielberg



Genre

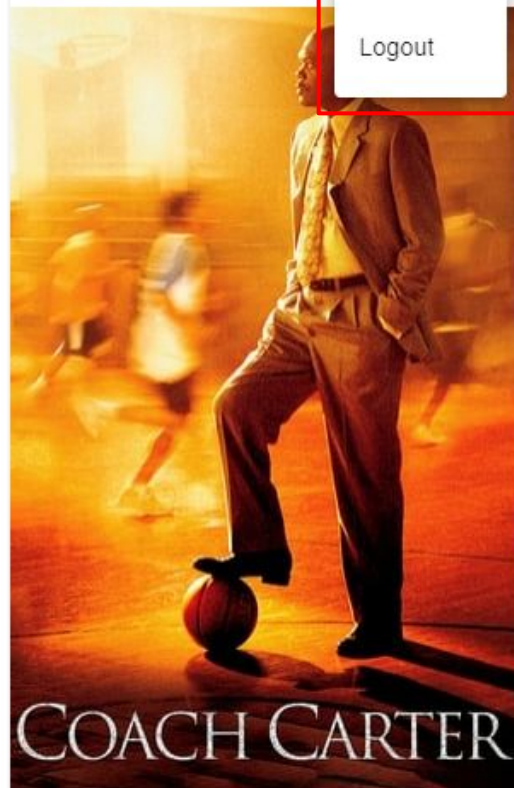
Director

Synopsis



Coach Carter

Directed by: Thom



Genre

Director

Synopsis

**MAIN  
VIEW**

(WITH

STYLING)

**Jaws**

Directed by: Steven Spielberg

**JAWS**

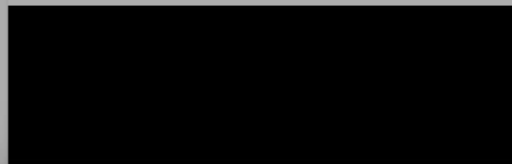
Genre

Director

Synopsis

**Jurassic Park**

Directed by: Steven Spielberg



Genre

Director

Synopsis

**Coach Carter**

Directed by: Thomas Carter



Genre

Director

Synopsis

**Science fiction**

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.

MAIN  
VIEW  
-  
MOVIE  
GENRE  
DETAILS

**Jaws**

Directed by: Steven Spielberg

**JAWS****Jurassic Park**

Directed by: Steven Spielberg

**Coach Carter**

Directed by: Thomas Carter

**Steven Spielberg**

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

Date of birth: 12/18/46

Date of death: NA

Genre

Director

Synopsis



Genre

Director

Synopsis



Genre

Director

Synopsis

**MAIN  
VIEW**

-

**MOVIE  
DIRECTOR  
DETAILS**

**Jaws**

Directed by: Steven Spielberg

**JAWS****Jurassic Park**

Directed by: Steven Spielberg

**Coach Carter**

Directed by: Thomas Carter

**Synopsis**

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

Director

Synopsis



Genre

Director

Synopsis



Genre

Director

Synopsis

**MAIN  
VIEW**

-

**MOVIE  
SYNOPSIS  
DETAILS**



MAIN  
VIEW

-

ADD TO  
FAVORITE

Jaws

Directed by: Steven Spielberg

JAWS



Genre

Director

Synopsis



Jurassic Park

Directed by: Steven Spielberg



Genre

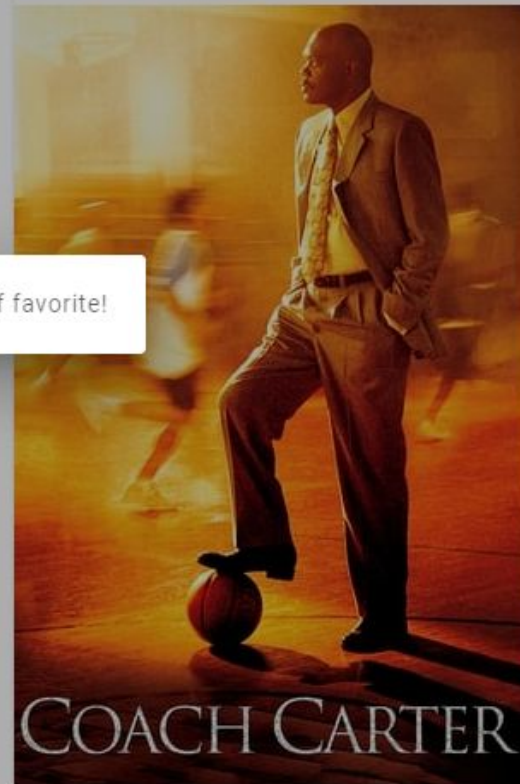
Director

Synopsis



Coach Carter

Directed by: Thomas Carter



Genre

Director

Synopsis



Coach Carter has been successfully **added** to your list of favorite!

MAIN  
VIEW

REMOVE  
FROM  
FAVORITE

Jaws

Directed by: Steven Spielberg

JAWS



Genre

Director

Synopsis



Jurassic Park

Directed by: Steven Spielberg



Genre

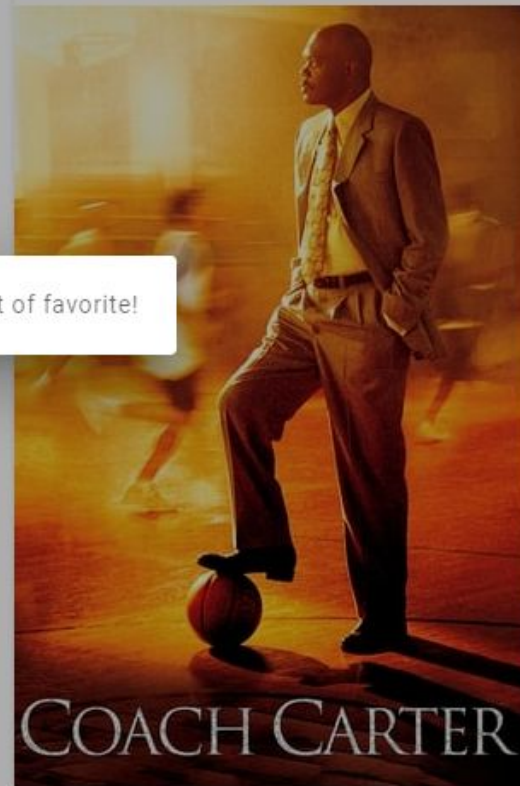
Director

Synopsis



Coach Carter

Directed by: Thomas Carter



Genre

Director

Synopsis



Coach Carter has been successfully **removed** from your list of favorite!

# MAIN VIEW

-

RESPONSIVE FOR  
MEDIUM SCREENS



# MAIN VIEW

-

RESPONSIVE FOR  
SMALL SCREENS



# WHAT WAS THE GOAL (SUITE)

Creating the profile page for users to be able to consult their info and update their account. To do so, I:

- Repeated some steps done in the main view UI creation (creation of a component and wrote codes in the TypeScript and HTML files)
- Implemented logic to display users information, allow users to update their account (username, password, email and birthday) or delete it, and show movies that have been added to their favorite. I also added some codes to ensure users can remove movies from their favorite from their profile page, and update their list of favorite movies accordingly in real time.
- Implemented logic to ensure that users' actions (e.g: movie added to favorite) remains between sessions, so that users can go back to their account after logout and always see / access the movies they've listed as favorite in previous sessions.
- Ensuring responsiveness and consistent styling / color palette with Angular Material for a great experience on any device sizes.

# FIRST RENDERED VIEW - PROFILE PAGE (NO STYLING)

## User Profile

Name: Testuser

Email: Testuser@gmail.com

Favorite movies:

# FIRST RENDERED VIEW - PROFILE PAGE (NO STYLING)

## User Profile

Name: Testuser

Email: Testuser@gmail.com

Favorite movies:

Jurassic Park

Directed by: Steven Spielberg



[remove from favorite](#)



# FIRST RENDERED VIEW - PROFILE PAGE (NO STYLING)

myFlix

Menu

## User Profile

### User information

Name: Testuser

Email: Testuser@gmail.com

### Edit information

Username

Username must be at least 5 characters long and contain only alphanumeric 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. If you want to add it, pick your birthday from the calendar icon.

Update

### Delete information

Delete account

### Favorite movies:



# PROFILE PAGE ON LARGE SCREENS (WITH STYLING)

myFlix

Menu

## User profile

### User information

**Name:** Youpi  
**Email:** youpi@hotmail.com

If you want 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).

### Edit information

Username

At least 5 characters long and only alphanumeric characters.

Password

Can contain alphanumeric / non-alphanumeric characters.

Email

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

Birthday



Birthday is optional. If you want to add it, use the calendar icon.

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 account

## Favorite movies

Move from side to side to see all movies (when many movies in favorite)

Jaws

Directed by: Steven Spielberg

Jurassic Park

Directed by: Steven Spielberg

Gladiator

Directed by: Ridley Scott

Silence of the Lambs

Directed by: Jonathan Demme

Blood diamond

Directed by: Edward

# PROFILE PAGE

—

## RESPONSIVE FOR MEDIUM SCREENS (PART 1)

### User profile

#### User information

Name: Youpi

Email: youpi@hotmail.com

If you want 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).

#### Edit information

Username

At least 5 characters long and only alphanumerical characters.

Password

Can contain alphanumeric / non-alphanumeric characters.

Email

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

Birthday



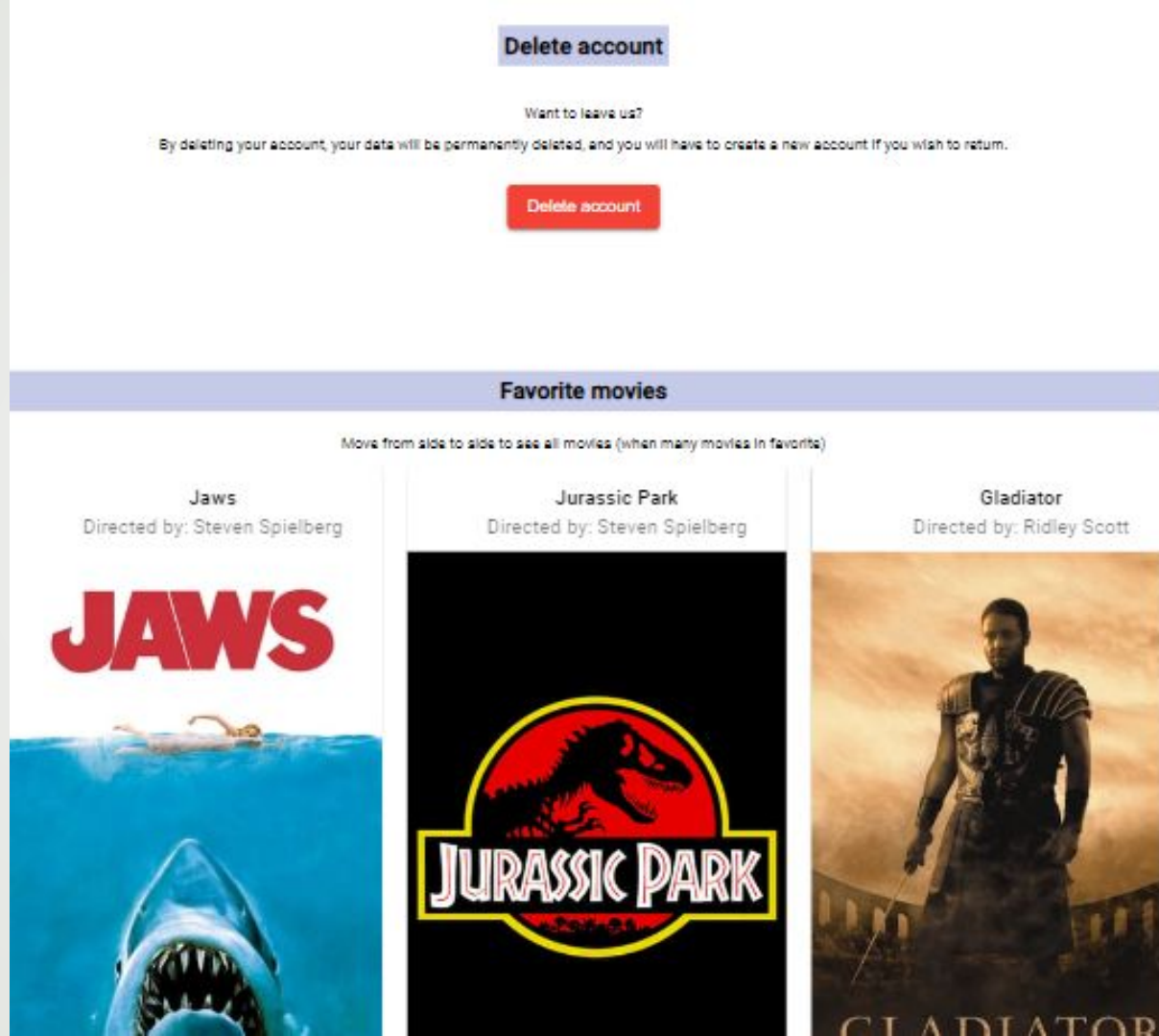
Birthday is optional. If you want to add it, use the calendar icon.

Update

# PROFILE PAGE

-

## RESPONSIVE FOR MEDIUM SCREENS (PART 2)



# PROFILE PAGE

—

## RESPONSIVE FOR SMALL SCREENS (PART 1)

myFlixMenu

User profile

User information

Name: Youpi  
Email: youpi@hotmail.com

If you want 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).

Edit information

Scroll down to complete all the form

Username

At least 3 characters long and only alphanumerical characters.

Password

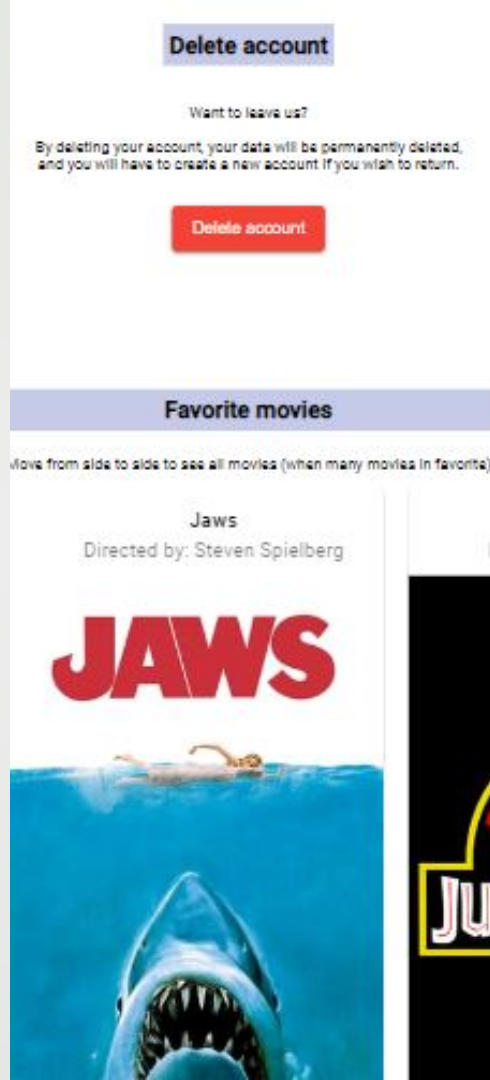
Can contain alphanumerics / non-alphanumerics characters.

Update

# PROFILE PAGE

—

## RESPONSIVE FOR SMALL SCREENS (PART 2)



# WHAT WAS THE GOAL (SUITE)

Implementing routing between the different views (with *RouterModule*) to redirect users when certain actions are triggered (e.g: users are redirected to the welcome view when they delete their account). The three routes implemented are:

- /welcome
- /movies
- /user-profile

# DECISIONS MADE

I chose how to structure and present the visuals (fonts, colors, buttons, dialogs, etc.) of the welcome view, the main view and the profile page, for all screen sizes.







Final code **revision**, testing,  
and **design last adjustments**

---

Skills used

---

Critical thinking  
Synthetization

# 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.

# **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.



github-pages

**Deploying** the web app

---

Skills used

---

N.O.

## **WHY WAS THIS STEP IMPORTANT**

This step was essential to ensure that the web app can be used and accessed by everyone.

## **WHAT WAS THE GOAL**

Deploying the app to gh-pages and make it available for everyone.

## **CHALLENGES OR SPECIAL POINTS OF CONSIDERATION**

Since I had previously implemented a CORS protection measure in the backend code, I had to add the new gh-pages domain of myFlix Angular in the backend's list of authorized domains, in order to ensure that the API would authorize the requests routed to it instead of blocking them.



README .

md

**Creating** all the necessary  
documentation

---

Skills used

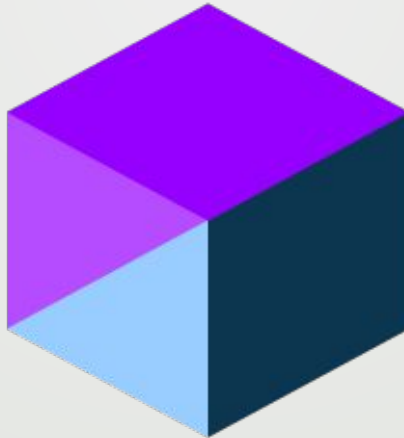
---

Technical writing  
Communication



# WHY WAS THIS STEP IMPORTANT

Adequate and complete documentation on a project, from the comments inside the codes to the creation of a README resource, greatly facilitate future updates, better and shared understanding within teams, and easier project appropriation by new members, among other things. It was therefore important to carry out such task.



# WHAT WAS THE GOAL

Preparing two of my projects for handoff to other developers - my myFlix backend API and my myFlix frontend Angular. To do so, I:

- Inserted meaningful comments where necessary in my projects files
- Generated documentation for myFlix backend using JSDoc (see image next slides)
- Generated documentation for myFlix Angular frontend using TypeDoc (see image next slides)
- Created a README on the project Github repository (see image next slides)

```
/**
 * @fileoverview passport.js
 * @description This file contains the logic that handles two authentication strategies: local (username and password) and jwt (JSON Web Token)
 */
```

```
const passport = require('passport'),
      LocalStrategy = require('passport-local').Strategy,
      Models = require('./models.js'),
      passportJWT = require('passport-jwt');
```

```
let Users = Models.User,
    JWTStrategy = passportJWT.Strategy,
    ExtractJWT = passportJWT.ExtractJwt;
```

```
passport.use(new LocalStrategy(
  {
    usernameField: 'Username',
    passwordField: 'Password'
  },
  (username, password, callback) => {
    console.log(username + ' ' + password);
    Users.findOne({ Username: username })
      .then((user) => {
```

### myFlix backend

COMMENTS WRITTEN IN JSDOC  
FORMAT TO GENERATE JSDOC  
DOCUMENTATION AUTOMATICALLY  
(SEE NEXT IMAGE)

# myFlix JSDoc DOCUMENTATION AUTOMATICALLY GENERATED

## Home

### auth.js

This file contains the API logic for user login, including authentication using Passport and emission of JWT tokens upon users successful login. It has one POST endpoint : /login.

Source: [auth.js, line 1](#)

### index.js

This file constitutes the main API element. All endpoints are defined there, as well as their specific parameters, what they require and what information they return. Endpoints summary (12 in total) :

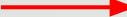
-6 GET  
-1 PUT  
-3 POST  
-2 DELETE

This file also contains the codes to import other project files (eg: auth.js which contains the logic for log in or passport.js which contains the logic related to JWT) as well as codes that configure certain additional elements, such as cross-origin resource sharing (CORS).

## Home

### Global

addFavoriteMovies  
createUser  
deleteFavoriteMovies  
deleteUser  
getDirector  
getGenre  
getMovies  
getOneMovie  
getUserInfo  
rootDirectory  
updateUser  
userLogin



```
/**
 * @module fetch-api-service
 * @remarks
 * This component regroups all the logic and function to call the different API's endpoints. Specific
 *
 * This component include several requests/functions:
 *
 * -6 GET requests
 * -1 PUT request
 * -3 POST requests
 * -2 DELETE requests
 */
```

```
import { Injectable } from '@angular/core';
import { catchError } from 'rxjs/operators';
import { HttpClient, HttpHeaders, HttpResponse } from '@angular/common/http';
import { Observable, throwError } from 'rxjs';
import { map } from 'rxjs/operators';
```

```
/**
 * Declaring the api url that will provide data for the client
 */
const apiUrl = 'https://my-weekend-movie-app-53a46e3377d7.herokuapp.com/api';
@Injectable({
  providedIn: 'root'
})
```

## myFlix frontend

COMMENTS WRITTEN IN TYPEDOC  
FORMAT TO GENERATE TYPEDOC  
DOCUMENTATION AUTOMATICALLY  
(SEE NEXT IMAGE)

# myFlix TYPEDOC DOCUMENTATION AUTOMATICALLY GENERATED

## my-flix-angular-client

- my-flix-angular-client
  - app-root
    - AppComponent
  - favorite-movie-add
    - AddFavoriteDetailsCompo
  - favorite-movie-remove
    - RemoveFavoriteDetailsCor
  - fetch-api-service
    - FetchApiDataService
  - get-all-movies
    - MovieCardComponent
  - get-information-on-director
    - DirectorDetailsComponent
  - get-information-on-genre
    - GenreDetailsComponent
  - get-information-on-synopsis
    - SynopsisDetailsComponen

## Module get-information-on-director

[Settings](#)

### Remarks

This component allows to display information about the director of each movie in a dialog prompt (when clicked by users). To do so, it takes the information about the director of X movie object, and renders it in the html.

### Example

In the TypeScript file, this is the data extracted from a movie object and required for this component to work:

```
data: {  
  Name: string,  
  Bio: string,  
  Birth: string,  
  Death: string  
}
```

These are then accessed within the html (e.g: `{{ data.Name }}` to display director's name).

### Param



# README SAMPLE - FULL VERSION ON GITHUB

## *myFlix* web app documentation (frontend - Angular) 🔗

### Content

- Project description
- User interface
- Technical aspects (overview)
- Technical aspect (development)
- 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 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 Angular for the best user-experience whenever they are accessing *myFlix* to read details about different movies.

*myFlix* frontend development can be breakdown in the five following points:

- **Who** — The users of *myFlix* web app, so movie enthusiasts who enjoy reading information about different

# **CHALLENGES** OR SPECIAL POINTS OF CONSIDERATION

Finding the right balance between giving the right level of information, while remaining as synthetic as possible.

## **DECISIONS** MADE

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