

# DOCUMENTAȚIA PROIECTULUI

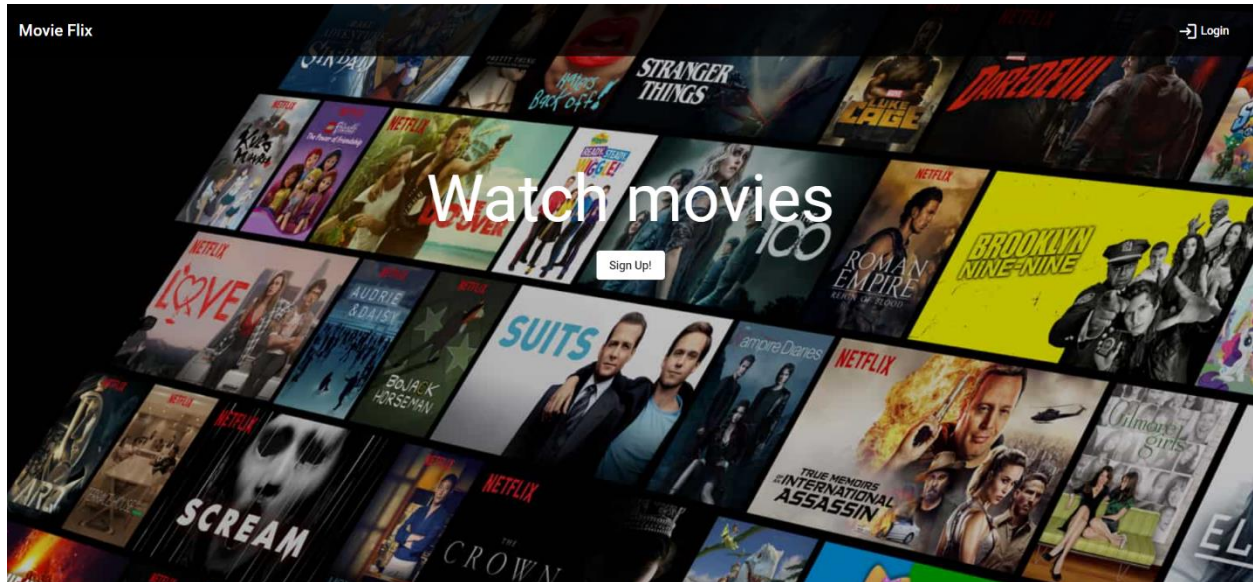
Proiect Movie Flix

STUDENȚI  
Aldes Maria  
Idu Sergiu

# Cuprins

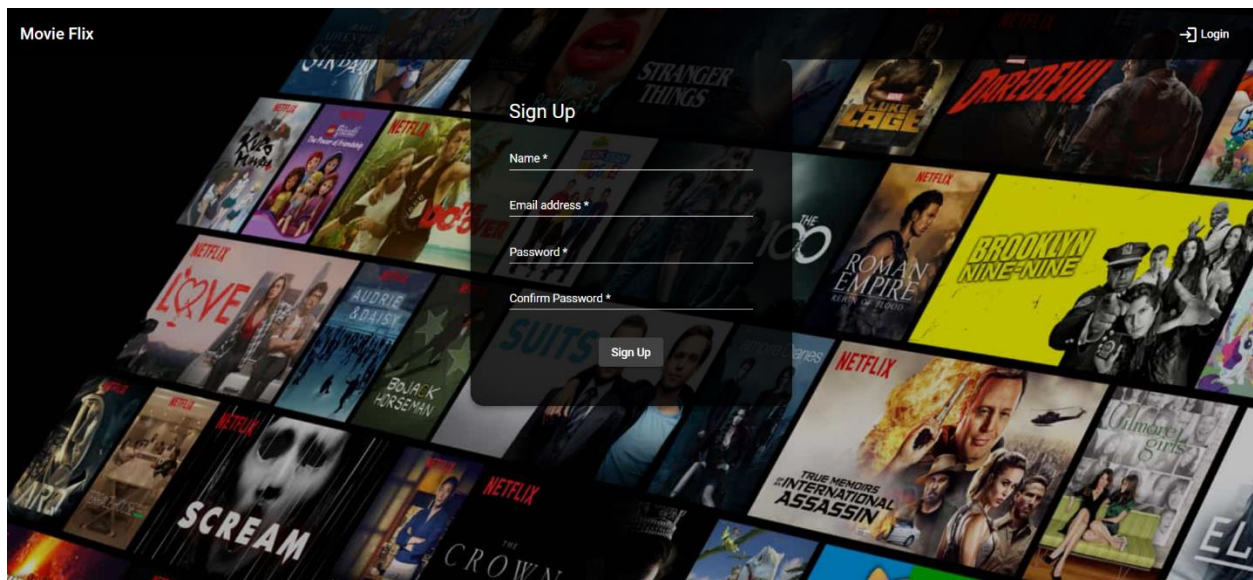
Prezentarea proiectului .....	3
Tehnologiile folosite si partea de “backend” .....	4
Arhitectura .....	6
Cum poate fi utilizata aplicatia .....	8
Concluzii .....	10

## Prezentarea Proiectului



Proiectul Movie Flix consta in dezvoltarea unei pagini web pentru accesarea detaliilor celor mai cunoscute filme.

Proiectul ofera posibilitatea utilizatorului de a-si crea un cont sau de a se autentifica, in cazul in care acesta nu are un cont creat.



Indiferent de preferinte, utilizatorul, de indata ce s-a conectat, poate vizualiza lista filmelor impreuna cu genurile sale. Odata cu selectarea filmului de interes, utilizatorul are acces la mai multe informatii depre acest film: descrierea, anul de lansare al filmului si rateingul acestuia.

## Tehnologiile folosite si partea de Back-end

- FireBase

In proiectul nostru ne-am folosit de FireBase pentru partea de autentificare. FireBase ofera un sistem de autentificare ce necesita e-mailul si parola.

Pentru partea de register, utilizatorul trebuie sa isi introduca numele, e-mail-ul si parola pe care trebuie sa o mai introduca inca o data, pentru confirmarea acesteia. Astfel datele sale sunt introduse in baza de date si retinute pentru ca utilizatorul sa se poata inregistra mai tarziu, astfel avand acces la aplicatie.

Pentru partea de login, utilizatorului i se cere sa introduca e-mail-ul cu care s-a inregistrat si parola pe care a setat-o. Dupa ce sunt verificate datele si se confirma ca acestea sunt in conformitate cu datele de introduse pentru crearea contului, inregistrarea este facuta cu succes, iar in acest moment utilizatorul poate vedea lista de filme.

```
export const environment = {  
  firebase: {  
    projectId: 'login-cbb59',  
    appId: '1:765446300148:web:4ce536095edba20a2799a1',  
    databaseURL: 'https://login-cbb59.firebaseio.com',  
    storageBucket: 'login-cbb59.appspot.com',  
    apiKey: 'AIzaSyA30HuU5Lb8FxlTatdPy4v4oiJ0ATM-jg',  
    authDomain: 'login-cbb59.firebaseio.com',  
    messagingSenderId: '765446300148',  
    measurementId: 'G-NCKNGVFEXKR',  
  },  
  BASE_URL: 'https://movies-app1.p.rapidapi.com/api/movies',  
  production: false  
};
```

Dupa realizarea conexiunii cu succes, am facut requesturi care ne-au ajutat la obtinerea datelor necesare. Astfel, pentru partea de login am putut face request pe baza de date sa aflam email-ul si parola utilizatorilor care si-au creat un cont. Dupa aceasta verificare utilizatorilor cu un cnt creat le putem oferi acces la pagina de filme.

## **Tehnologiile folosite si partea de Back-end**

- RaidAPI

Pentru proiectul nostru am folosit de asemenea platforma RapidAPI. Aici am cautat un API potrivit pentru ceea ce cautam noi.

RapidAPI ne-a pus la dispozitie o lista de care ne-am putut folosi pentru a putea afisa filmele necesare, impreuna cu detaliile specifice pentru fiecare film.

```
setHeaders: {  
  'X-RapidAPI-Key': 'f4ddcbce21msh2b608cb3923ac9ep125378jsn7e5b03d430a6',  
  'X-RapidAPI-Host': 'movies-app1.p.rapidapi.com'  
},
```

Am realizat conexiunea cu API-ul gasit, iar pe baza acestui API am facut request-uri pentru fiecare film in parte, rezultand sa avem acces la detaliile necesare.

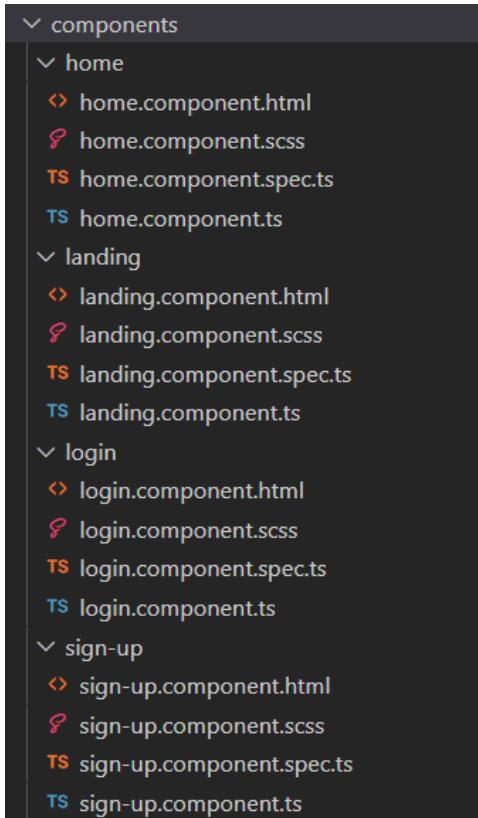
Request-urile realizate ne-au fost utile pentru a afisa lista de filme. Am facut cate un request in parte pentru accesarea tuturor particularitatilor pe care fiecare film il are: titlul acestuia, imaginea de coperta, o lista cu genurile din care face filmul parte.

Pentru afisarea detaliilor fiecarui film in parte am facut request-uri in urma carora am primit ca raspuns anul de lansare al filmului, descrierea si nu in ultimul rand, rating-ul filmului.

## Arhitectura

- Componente si servicii

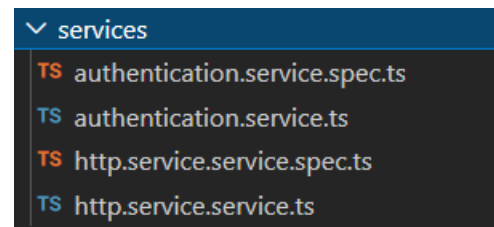
Proiectul nostru este impartit in mai multe componente care ne faciliteaza aplicatia.  
Astfel avem:



- componenta 'home.component' care reprezinta pagina de home, unde avem lista de filme;
- componenta 'landing.component' ce reprezinta pagina principala, cea care are optiunile de autentificare si de login;
- componenta 'login.component' care reprezinta fereastra de login, unde utilizatorul se logeaza
- componenta 'sign-up.component' ce reprezinta pagina de register, acolo unde utilizatorul isi inregistreaza datele pentru a-si crea un cont.

Cu ajutorul acestor componente am reusit sa declaram ce se afiseaza pe paginile aplicatiei, am putut edita continutul si design-ul acestuia.

De asemenea avem servicii care ne ajuta la autentificarea utilizatorului si la obtinerea listei de filme.

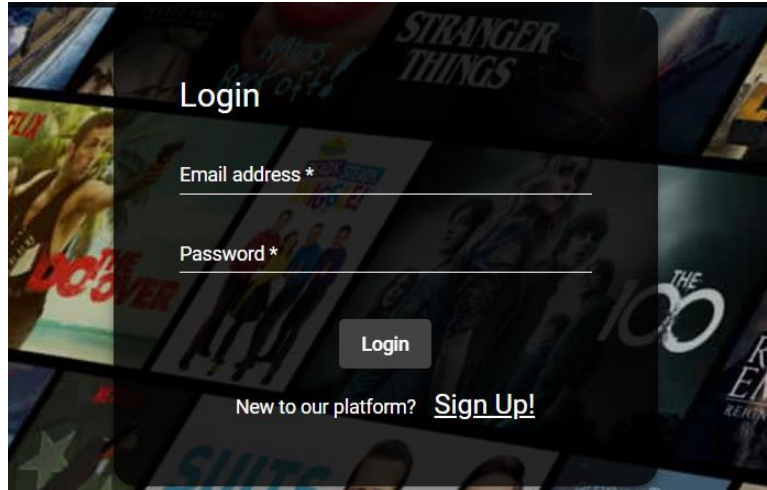


```
export class AuthenticationService {  
  
  currentUser$=authState(this.auth);  
  
  constructor(private auth:Auth ) { }  
  
  login(email: string, password: string): Observable<any> {  
    return from(signInWithEmailAndPassword(this.auth, email, password));  
  }  
  logout()  
  {  
    return from(this.auth.signOut())  
  }  
  
  signUp(email: string, password: string): Observable<UserCredential> {  
    return from(createUserWithEmailAndPassword(this.auth, email, password));  
  }  
}
```

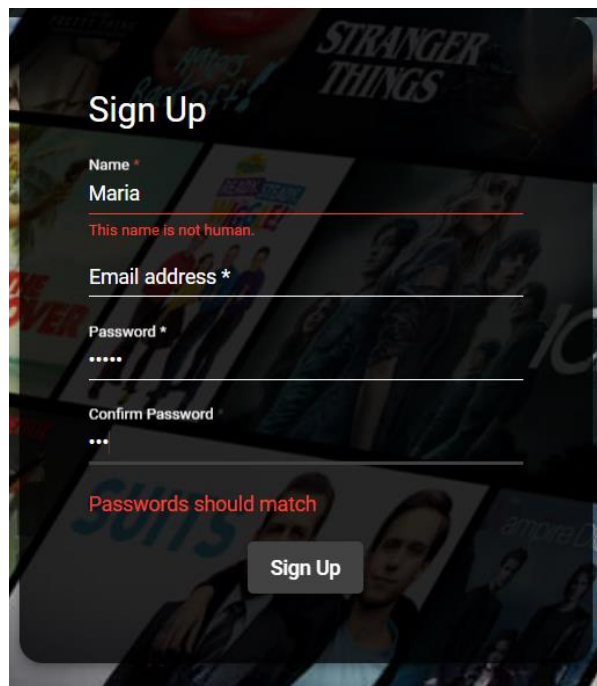
```
export class HttpService {  
  
  constructor(private http: HttpClient) { }  
  
  getMovieList(): Observable<APIResponse<Movie>> {  
    return this.http.get<APIResponse<Movie>>(`${env.BASE_URL}`);  
  }  
}
```

- Validatori Custom

Pentru partea de autentificare ne-am folosit de Material. Cu ajutorul acestei librării am creat formularul de autentificare si de login.

A screenshot of a login form overlaid on a background of movie posters. The form has a title 'Login' at the top. Below it are two input fields: 'Email address \*' and 'Password \*'. A 'Login' button is positioned below the password field. At the bottom, there is a link that says 'New to our platform? [Sign Up!](#)'.

Am creat 2 validatori custom pe form-ul de autentificare. Primul validator este pentru a verifica ca in momentul crearii unui cont, utilizatorul in field-ul 'Name' isi introduce un nume valid, adica un nume si un prenume. Cel de al doilea validator custom verifica ca parola introdusa in field-ul 'Password' sa fie aceeaasi cu cea din field-ul 'Confirm Password'.

A screenshot of a sign-up form overlaid on the same movie poster background. The form has a title 'Sign Up'. It contains four input fields: 'Name \*' with the value 'Maria', 'Email address \*', 'Password \*' with masked characters '.....', and 'Confirm Password \*' with masked characters '...'. There are two red error messages: 'This name is not human.' below the name field and 'Passwords should match' below the password fields. A 'Sign Up' button is at the bottom.



## Cum poate fi utilizata aplicatia ?

Aplicatia porneste cu pagina de start unde utilizatorului i se ofera optiunea de 'sign-up' unde acesta isi creeaza un cont, sau optiunea de 'login', unde se conecteaza cu un cont deja existent.

In momentul apasarii butonului de 'sing-up' se va deschide un formular necesar autentificarii.

In acest formular se gasesc campurile care trebuie obligatoriu completate:

- 'Name', unde utilizatorul isi va introduce numele si prenumele, alfel ii va aparea o eroare cum ca numele nu este unul real;
- 'Email address', camp destinat introducerii email-ului,
- 'Password', unde utilizatorul isi va alege o parola,
- 'Confirm password', camp in care utilizatorul trebuie sa introduca exact aceeasi parola pentru a evita eroarea care ar aparea in cazul in care cele doua parole nu sunt identice.

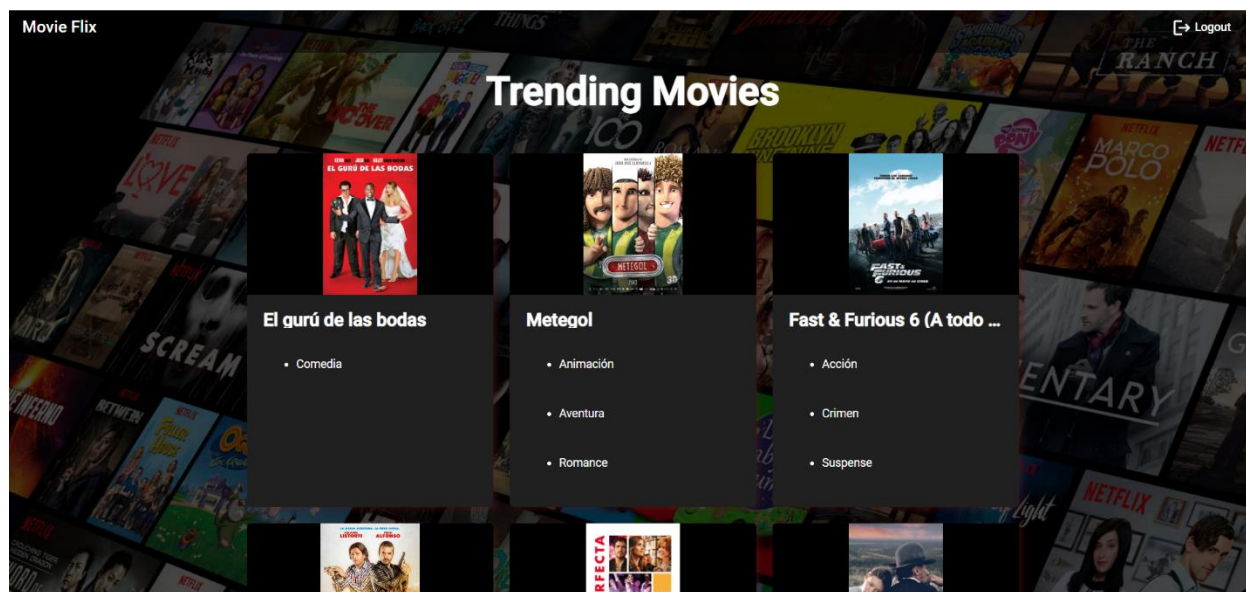
Toate aceste date vor fi retinute in baza de date pentru a fi mai apoi folosite pentru conectarea utilizatorului.

In momentul apasarii butonului de 'login', va aparea formularul pentru introducerea datelor contului creat.

Formularul contine doua campuri:

- 'Email address', acolo unde utilizatorul trebuie sa introduca email-ul inregistrat
- 'Password', unde trebuie introdusa parola asociata email-ului de mai sus.

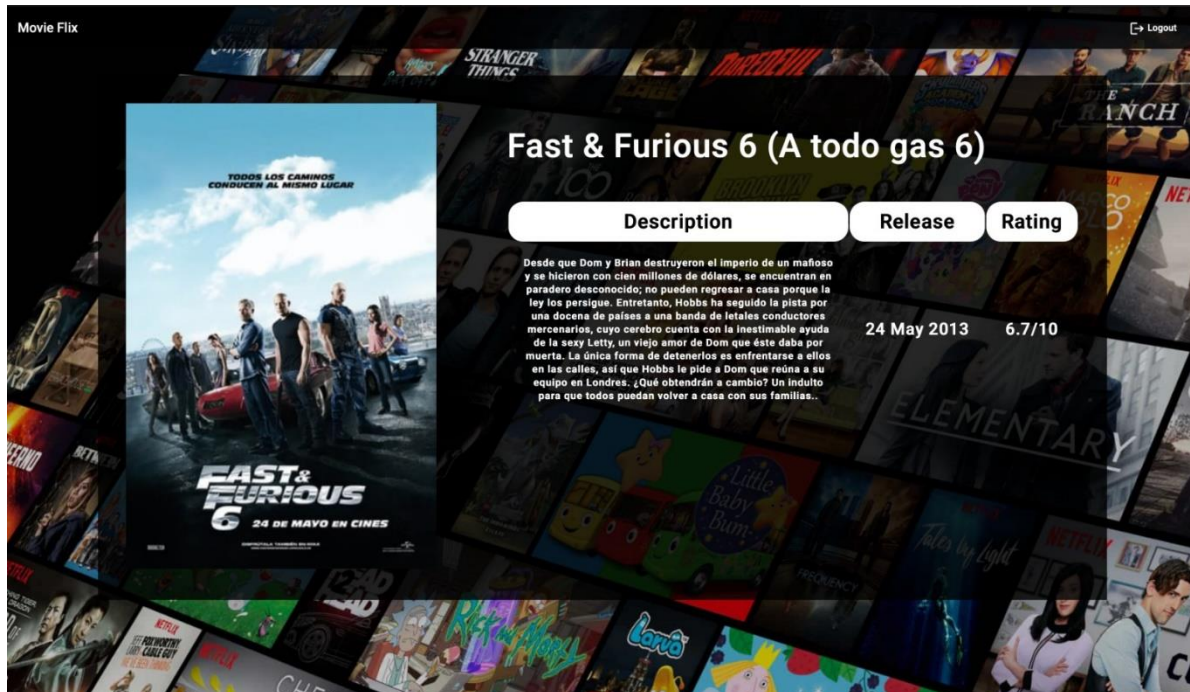
Dupa ce utilizatorul si-a introdus corect datele contului, autentificarea este realizata cu succes, iar acum utilizatorul are acces la pagina de filme.





Pagina de filme contine o lista cu filmele din tranding. Aceste filme sunt organizate sub forma unor carduri care contin imaginea filmului, titlul si genurile acestuia.

In momentul in care utilizatorul selecteaza filmul despre care doreste sa afle mai multe informatii, este redirectionat spre pagina unica a acestui film.



Pe pagina de detalii utilizatorul va regasi pe langa informatiile deja cunoscute de pe pagina anterioara, descrierea filmului, data de lansare si ratingul acestuia.

## **Concluzii**

Datorita acestui proiect am avut ocazia sa lucram cu FireBase, o baza de date in Cloud. De asemenea am descoperit un mod foarte util de a conecta un API prin platforma RapidAPI, aceasta oferind o multitudine de API-uri potrivite pentru orice tip de aplicatie si limbaj de programare.

In concluzie acest proiect ne-a oferit oportunitatea de a lucra in echipa asupra unui proiect pregatindu-ne pentru colaborarea cu alti colegi pe viitor.