

# **DOCUMENTAȚIE PROIECT DAWM (ANGULAR)**

## **MOVIE PLATFORM**

-realizat de Brânzei Radu și Codreanu Alexandru, grupa 10LF211-

Aplicația este o platformă de vizualizare a informațiilor și rating-urilor personale despre diverse filme și seriale TV, dezvoltată în Angular.

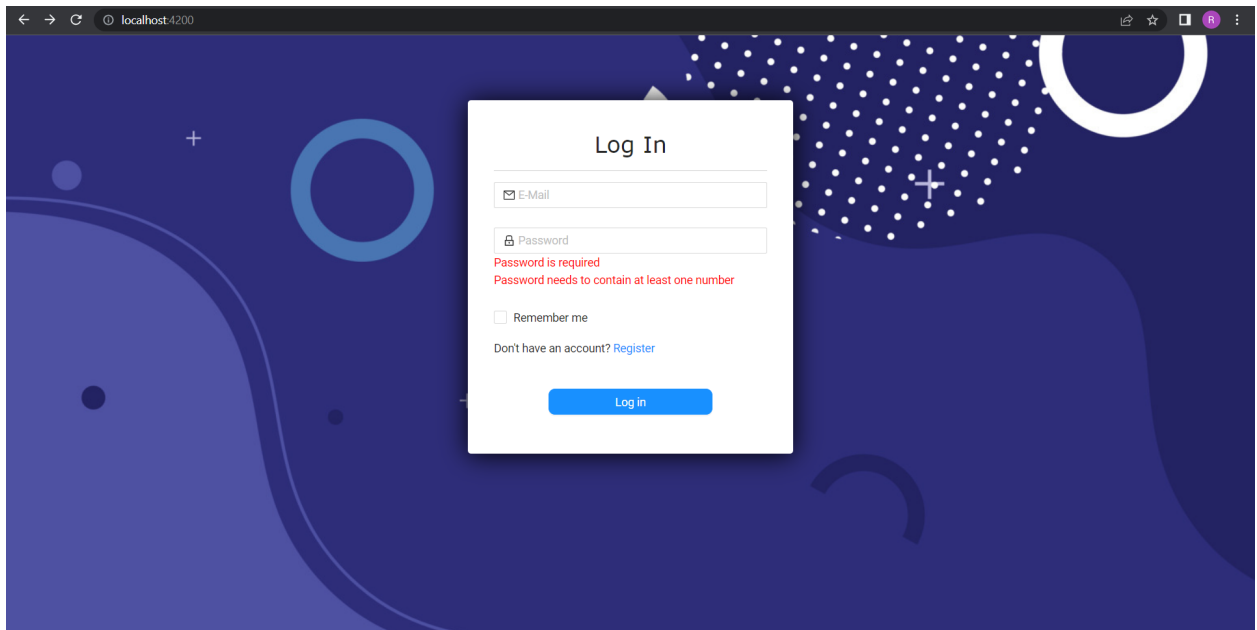
Aceasta este structurată pe 2 segmente principale – segmentul de autentificare (log in / register) și segmentul de vizualizare a unui tabel cu filmele selectate de fiecare utilizator.

### **STRUCTURA CODULUI**

- 3 module (pentru log in, register și main-table)
- Module încărcate prin lazy-loading
- Autentificare prin fișiere .json
- Secțiuni individuale (foldere) pentru servicii, guard-uri, interfețe, validatori, pipes și fișiere media

## LOG IN PAGE

Pagina de start ce reprezintă ruta nulă, va fi cea unde utilizatorul va trebui să introducă datele unui cont (email + parolă). Cu ajutorul unor validatori existenți în Angular și a unor validatori custom, am restricționat email-ul și parola după anumite condiții (email-ul trebuie să aibă o formă acceptată, fiind nevoie de un format [name@domain.tld](#), iar parola trebuie să conțină minim 8 caractere, dintre care minim un număr).



Pentru design-ul ilustrat, dar și pentru anumite funcționalități, precum validatoarele, am folosit librăria de UI, NgZorro.

Log In-ul se face prin intermediul verificării dintr-un fișier .json, unde avem salvați mai mulți User.

```
[
  {
    "id": 1,
    "email": "george.bluth@reqres.in",
    "username": "George Bluth",
    "password": "password123",
    "scores": [
      {
        "idMovie": 1,
        "Score": 10
      },
      {
        "idMovie": 2,
        "Score": 9
      },
    ],
  },
]
```

Pentru verificarea datelor, vom folosi un service ce parcurge acest Users.json atunci când e nevoie de logarea sau înregistrarea unui nou utilizator.

Deși nu lucrăm cu un Fake API, vom simula oricum folosirea unui token, pentru a reține informații despre sesiunea curentă a utilizatorului logat. La momentul logării, token-ul va reține id-ul utilizatorului conectat, până la finalul sesiunii, sau până la momentul Log Out-ului. În cazul în care utilizatorul selectează opțiunea de Remember Me, acesta va rămâne conectat doar până în momentul log out-ului.

```
private token: string | null = '';
private users: User[] = userData;

constructor(private router: Router) {
  if (sessionStorage.getItem('userToken')) {
    this.token = sessionStorage.getItem('userToken');
  }

  if (localStorage.getItem('userToken')) {
    this.token = localStorage.getItem('userToken');
  }
}
```

```

successLogin(payload: LoginPayload): boolean {
  for(let user of this.users) {
    if(user.email == payload.email && user.password == payload.password) {
      this.token = user.id.toString();
      if(payload.rememberMe) {
        localStorage.setItem('userToken', this.token);
      }
      else {
        sessionStorage.setItem('userToken', this.token);
      }
      return true;
    }
  }
  return false;
}

```

Prin intermediul Guard-urilor, interzicem accesul la tabela principală utilizatorilor neînregistrați. Acest Guard trebuie aplicat componentei principale, în app-routing.module.ts.

```

export class AuthGuard implements CanActivate {
  constructor(private router: Router) {}

  canActivate():
    | Observable<boolean | UrlTree>
    | Promise<boolean | UrlTree>
    | boolean
    | UrlTree {
    if (
      localStorage.getItem('userToken') ||
      sessionStorage.getItem('userToken')
    ) {
      return true;
    }

    this.router.navigateByUrl('/');
    return false;
  }
}

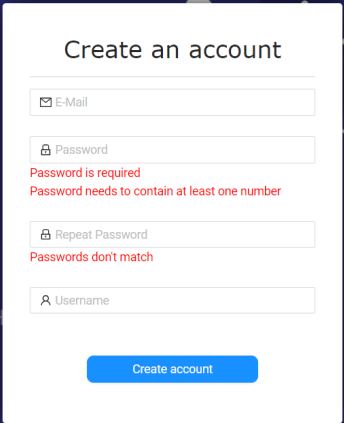
```

În momentul log out-ului vom goli localStorage și sessionStorage de token-ul existent, și vom naviga înapoi la pagina de start, unde vom putea reintroduce date de intrare.

```
logout() {  
  this.token = null;  
  sessionStorage.clear();  
  localStorage.clear();  
  this.router.navigateByUrl('/');  
}
```

## REGISTER PAGE

Dacă din form-ul de Log In selectăm că dorim să creăm un cont nou, atunci vom fi trimiși către form-ul de register, pe ruta '/register'.



The image shows a 'Create an account' form on a dark blue background with abstract white and light blue shapes. The form is white and contains the following fields and messages:

- E-Mail**: A text input field.
- Password**: A text input field with a red error message below it: "Password is required" and "Password needs to contain at least one number".
- Repeat Password**: A text input field with a red error message below it: "Passwords don't match".
- Username**: A text input field.
- Create account**: A blue button at the bottom of the form.

Folosim din nou validatoare pentru parolă, confirmarea parolei (trebuie să fie identică cu cea introdusă inițial). În momentul apăsării pe Create Account, service-ul folosit pentru tot ce ține de autentificare va verifica dacă email-ul este deja folosit de alt cont, caz în care crearea nu va avea loc. După o creare reușită, user-ul va fi din nou redirectionat către form-ul de log in, unde va trebui să introducă noile date.

```

createAccount(): void {

  var newUser: User = {
    id: this.authService.getUsers().length + 1,
    email: this.registerEmail,
    password: this.registerPassword,
    username: this.registerUsername,
    scores: []
  }

  if(this.registerForm.valid) {
    if(this.authService.addUser(newUser)) {
      this.router.navigateByUrl('/');
    }
    else {
      this.incorrectRegister = true;
    }
  }
}
}

```

## MAIN PAGE

A doua parte este cea de vizualizare a listei de seriale/filme aferentă fiecărui user.

Accesarea acestei secțiuni se face după ce se înregistrează userul folosindu-și email-ul și parola. Folosind ruta cu parametrul de id al userului se v-a încarca folosind un service toate filmele salvate de user.

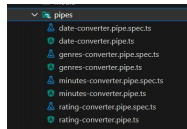
Filmele conțin două câmpuri care v-or fi transformate prin pipe-uri pentru vizualizarea mai plăcută a user-ului. Aceste câmpuri sunt genurile filmului, păstrate ca un array de Id-uri care vor fi mapate unor obiecte de tip Genre corespunzătoare și al doilea câmp este rating-ul păstrat ca un id care tot așa v-a fi mapat unui obiect de tip Rating.

```

{
  "id": 22,
  "image": "https://cdn.myanimelist.net/images/anim/1668/188792.jpg",
  "title": "The Stranger by the Shore",
  "runtime": 54,
  "releaseDate": "2020-09-11",
  "score": 7.89,
  "series": 1,
  "genres": [5, 13],
  "rating": 3
}

```

Se mai folosesc de asemenea pipe-uri pentru: convertirea datei dintr-un format de tipul 'dd-mm-yyyy' într-unul de tip 'dd(cardinal) of Month, yyyy', convertirea unui număr de minute la un număr de ore și minutes cu 'h' reprezentând 'hour' și 'm' reprezentând 'minutes', convertirea id-uri păstrate în Movie la denumirea lor cum ar fi '1' pentru genuri ar fi 'Action' și '4' pentru rating ar fi 'R'.



```
export class GenresConverterPipe implements PipeTransform {
  constructor(private service: MiscServicesService) {}

  transform(genres: number[]): string {
    let actualGenres = this.service.getGenresByIds(genres);
    let output: string = '';
    actualGenres.forEach((element, index) => {
      output += element.name;
      if (index !== actualGenres.length - 1) {
        output += ', ';
      }
    });
    // You, yesterday + worked on main table, services and ts
    return output;
  }
}
```

```
export class MinutesConverterPipe implements PipeTransform {
  transform(minutes: number): string {
    if (minutes < 0) {
      return 'N/A';
    }

    const hours = Math.floor(minutes / 60);
    const remainingMinutes = minutes % 60;

    const formattedTime = `${hours}h ${remainingMinutes}m`;
    return formattedTime;
  }
}
```

```
transform(dateString: string): string {
  const dateParts = dateString.split('-');
  const year = dateParts[0];
  const month = this.getMonthName(parseInt(dateParts[1]));
  const day = this.getFormattedDay(parseInt(dateParts[2]));

  const formattedDate = `${day} of ${month}, ${year}`; // You, yesterday
  return formattedDate;
}

private getMonthName(month: number): string {
  const monthNames = [
    'January', 'February', 'March', 'April', 'May', 'June',
    'July', 'August', 'September', 'October', 'November', 'December'
  ];
  return monthNames[month - 1] || '';
}

private getFormattedDay(day: number): string {
  let formattedDay = day.toString();

  if (formattedDay.length === 1) {
    formattedDay = day.toString().replace('0', '');
  }



















  if (day >= 11 && day <= 13) {
    formattedDay += 'th';
  } else {
    formattedDay += 'th';
  }

  return formattedDay;
}
```

Tabelul in sine este făcut să arate doar câte 6 filme/seriale deodată, folosindu-se de un paginator pentru a naviga printre ele.

Search bar-ul este un input box in care fiecare eveniment de tip keystroke (keyup) activează o funcție care afișează o listă de butoane, butoane care conțin titlurile cele mai apropiate de ce a scris user-ul și o imagine aferentă filmului/serialului. Acele butoane odată apăsate setează pagina tabelului pe pagina din care fac parte și afișează detaliile in mijlocul ecranului deasupra tabelului. Afișarea detaliată se poate face și folosind unul din butoanele din coloana Actions a tabelului.

Despre tabel in sine; are mai multe coloane : imaginea, titlul, durata, data lansării, scorul (luat de la MyAnimeList) și scorul dat de utilizator, genurile, rating-ul și acțiuni. În afară de prima și ultima coloană restul pot fi apăsate pentru a fi sortate în funcție de acel criteriu.

Movies									
Image	Title	Runtime	Release Date	Score	Your Score	Genres	Rating	Actions	
	Steins;Gate	9h 36m	6th of April, 2011	6.07	10	Drama, Sci-Fi, Suspense	PG-13		
	Kaguya-sama: Love is War - Ultra Romantic	4h 59m	9th of April, 2022	9.05	8	Comedy, Romance	R		
	Fruits Basket	24h 9m	6th of April, 2019	9	9	Drama, Romance, Supernatural	PG-13		
	[Shin No Ki]	5h 30m	12th of April, 2023	6.97	9	Drama, Supernatural	PG-13		
	Your Name.	1h 48m	26th of August, 2016	8.85	10	Award Winning, Drama, Supernatural	PG-13		
	Bocchi the Rock!	4h 48m	9th of October, 2022	8.85	8	Comedy	PG-13		

1

2

3

>

Movies...

9.05

Kaguya-sama: Love is War - Ultra Romantic

Runtime: 4h 59m

Released: 9th of April, 2022

Genres: Comedy, Romance

Rating: R













Image	Title	Runtime	Release Date	Score	Your Score	Genres	Rating	Actions
	Steins;Gate	9h 36m	6th of April, 2011	9.07	10	Drama, Sci-Fi, Suspense	PG-13	  
	Kaguya-sama: Love is War - Ultra Romantic	4h 59m	9th of April, 2022	9.05	8	Comedy, Romance	R	  
	Kaguya-sama: Love is War - Ultra Romantic	4h 59m	9th of April, 2022	9.05	8	Comedy, Romance	R	