DOCUMENTAȚIA PROIECTULUI

Movie List – Gestionarea și Descoperire de Filme

STUDENȚI

Munteanu Maria-Magdalena

**Cuprins**

½ - 1 pagina despre prezentarea proiectului, ce isi propune, ce probleme rezolva, solutii

½ - 1 pagina despre tehnologiile folosite

½ - 1 pagina despre partea de “backend” - request-uri folosite

1-3 pagini despre arhitectura (componentele principale, navigare, baza de date etc.)

1-2 pagini despre cum poate fi utilizata aplicatia: tipuri de utilizatori, ce vede fiecare, autentificare etc

½ - 1 pagina concluzii

Prezentarea proiectului

Movie List este o aplicație dezvoltată folosind framework-ul Angular, care își propune să ofere utilizatorilor o experiență simplă și eficientă în gestionarea și descoperirea de filme. Cu ajutorul unei interfețe intuitive și a funcționalităților avansate, utilizatorii pot sa adauge si sa editeze lista de filme, să le urmărească și să descopere noi titluri într-un mod facil și captivant. Pe lângă asta, aplicația oferă și o pagină de autentificare și înregistrare pentru a asigura securitatea și confidențialitatea datelor utilizatorilor.

*Gestionarea listelor de filme:* Cu Movie List, utilizatorii pot organiza și gestiona cu ușurință colecția de filme. Ei pot adăuga și elimina filme si pot de asemenea sa editeze întrările deja existente.

*Descoperirea de noi filme:* Aplicația oferă funcții de căutare și filtrare avansate, ceea ce facilitează descoperirea de noi filme. Utilizatorii pot cauta un film bazat pe titlu, sau în cazul în care doresc sa se uite la filme mai vechi, lista filmelor poate fi sortata în funcție de an pentru o accesare mai ușoară.

*Interfață intuitivă:* Aplicația dispune de o interfață prietenoasă, cu un design modern și funcționalități intuitive. Utilizatorii vor putea naviga ușor prin aplicație și vor avea acces rapid la toate funcțiile oferite.

*Pagină de login/register:* Pentru a asigura securitatea și confidențialitatea datelor utilizatorilor, aplicația include o pagină de autentificare și înregistrare. Utilizatorii vor putea crea conturi personale, unde își pot gestiona listele de filme și pot interacționa cu ceilalți utilizatori.

***Tehnologii folosite***

Pentru dezvoltarea aplicației Movie List, au fost utilizate tehnologii moderne și puternice pentru a crea o experiență de utilizare captivantă și o interfață prietenoasă. Aceste tehnologii includ Angular, HTML, SCSS și TypeScript, care împreună asigură un mediu de dezvoltare flexibil.

*Angular:* este un framework de dezvoltare front-end extrem de popular și puternic. Acesta oferă o arhitectură modulară și un set complet de instrumente pentru dezvoltarea aplicațiilor web complexe. Angular utilizează TypeScript pentru a adăuga tipizare statică la JavaScript, ceea ce ajută la detectarea și rezolvarea erorilor în timpul procesului de dezvoltare. Angular facilitează gestionarea stării aplicației, rutarea, manipularea evenimentelor și crearea de componente reutilizabile.

*HTML:* este limbajul utilizat pentru structurarea și prezentarea conținutului web. În cadrul aplicației Movie List, HTML este utilizat pentru a defini structura și elementele interfeței utilizatorului, cum ar fi butoanele, formularele și listele de filme. Prin intermediul HTML, elementele pot fi organizate într-un mod semnificativ și pot fi stilizate utilizând CSS.

*SCSS:* este o extensie a limbajului CSS, care adaugă funcționalități suplimentare și o sintaxă mai avansată. SCSS permite dezvoltatorilor să folosească variabile, funcții și nestarea selectorilor, ceea ce facilitează organizarea și reutilizarea stilurilor CSS în cadrul aplicației. Prin utilizarea SCSS, stilurile aplicației Movie List pot fi definite într-un mod clar și modular, asigurând o gestionare eficientă și ușurință în menținerea lor pe termen lung.

*TypeScript:* este un superset al limbajului JavaScript care adaugă tipizare statică și caracteristici avansate de programare orientată pe obiect. În dezvoltarea aplicației Movie List, TypeScript este utilizat ca limbaj principal de programare. Aceasta oferă avantaje precum detecția erorilor în timpul compilării, completarea automată a codului și o structură mai puternică a codului. Prin utilizarea TypeScript, putem scrie cod mai ușor de întreținut și mai ușor de extins.

***BackEnd – request-uri***

Partea de "backend" a aplicației Movie List include integrarea cu API-ul de autentificare și înregistrare furnizat de ReqRes (https://reqres.in/) și utilizarea unui fișier JSON pentru a stoca și accesa lista de filme. Aceasta implică utilizarea unor request-uri pentru a comunica cu API-ul și pentru a manipula datele în fișierul JSON. Prin intermediul acestor request-uri, aplicația gestionează funcționalitățile de autentificare, înregistrare și gestionare a listei de filme.

Request-ul de autentificare (POST /login): cand un utilizator încearcă să se autentifice în aplicația Movie List, se realizează un request POST către URL-ul /login al API-ului ReqRes. Acest request include un obiect care conține credențialele utilizatorului, cum ar fi adresa de email și parola. Serverul API-ului verifică aceste informații și returnează un răspuns care conține detalii despre utilizatorul autentificat sau un mesaj de eroare în cazul în care autentificarea a eșuat. Această interacțiune asigură securitatea și autentificarea corectă a utilizatorilor în aplicație. În mod foarte diminuarea o sa lucreze și request-ul de înregistrare.

Accesarea listei de filme (GET /movies): pentru a afișa lista de filme în aplicația Movie List, aceasta realizează un request GET către o rută specifică (cum ar fi /movies). În cadrul backend-ului, acest request este gestionat și rezultatul este obținut din fișierul JSON care conține lista de filme. Serverul furnizează un răspuns care conține datele din fișierul JSON, iar apoi aceste date pot fi prelucrate și afișate în interfața utilizatorului.

***Arhitectură***

*Arhitectura aplicatiei* mele implica doua componente principale: autentificarea (mainauth) si panoul de control (maindashboard), pe care urmeaza sa le detaliem in continuare:

*Autentificarea* (folder-ul mainauth) contine componente si servicii relevante pentru autentificarea utilizatorului cum ar fi: componenta auth-login-form, această componentă furnizează interfața pentru formularul de autentificare și înregistrare. Utilizatorii pot completa informațiile necesare pentru autentificare sau crearea unui cont nou. Autentificarea se bazeaza pe un fake API.

In continuare in acesta mai exista si logica pentru validatori custom, acestia fiind: Validatorul pentru parola puternica care ii cere utilizatorului sa introduca o parola ce contine cel putin o litera mare, o litera mica si un numar; de asemenea mai avem validatorul pentru confirmarea parolei , ce o sa verifice daca parola introdusa in campul “Confirm password” din “SignUp” este aceiasi cu parola introdusa in campul “Password”.

Avem de asemenea si un fisier ts ce se ocupa de serviciile specifice autentificarii, precum comunicarea cu Fake API-ul nostru mentionat mai devreme si gestionarea token-urilor de autentificare.

*Panoul de Control* (folder-ul maindashboard) conține componente și servicii relevante pentru afișarea și gestionarea listei de filme. Acesta la randului lui are create foldere pentru a putea avea o aplicatie bine organizata: Foldere existente sunt components, interface si services pe care o sa le detaliem.

In primul rand, în folder-ul *“components”* am împărțit aplicația în trei subfoldere

* *"dashboard":* Această componentă este responsabilă pentru afișarea paginii principale a panoului de control. Aceasta include apelarea componentei "header" și a componentei "tabela".
* *"header":* Componenta "header" este făcută pentru a sta în mod simplu deasupra componentei "table" și aceasta conține un searchbar pentru căutarea de filme și o funcție (apelata la apăsarea unui buton "Add") pentru adăugarea de filme noi.
* *"table":* Această componentă reprezintă o tabelă în care sunt afișate informațiile despre filme într-un mod plăcut vizual. Aceasta se bazează pe o interfață definită în folderul "interfaces" și utilizează date dintr-un fișier JSON pentru a popula tabela.

Urmează folderul *“interface”* în care am adăugat interfața care definește structura datelor pentru filme, ce va fi utilizata în “table”.

În mod normal în *“services”* exista fișierul ts ce se ocupa de serviciile specifice gestionării filmelor. Aici se pot găsi funcții pentru adăugarea, editarea, ștergerea și căutarea filmelor în baza de date. Tot în acest fișier am adăugat și fișierul JSON cu datele din tabele.

*Rutarea în cadrul aplicației* Angular are un rol important în gestionarea navigării utilizatorului între diferitele pagini sau componente, precum pagina de autentificare și pagina de panou de control (dashboard).

Pentru a implementa această rutare, am utilizat modulul de rutare din Angular, care oferă funcționalități puternice pentru a gestiona navigarea între diferitele componente și pentru a afișa componentele corespunzătoare în funcție de ruta accesată.

*Configurarea rutării:* În fișierul "app-routing.module.ts", am configura rutele aplicației. Aici am definit rutele pentru paginile de autentificare și panou de control. Rută principala o sa fie pagina de autentificare/înregistrare, pe lângă aceasta avem și ruta pentru dashboard și mai avem un wildcard ce o sa redirecționeze utilizatorului la o pagina “page-not-found”.

Pentru a proteja ruta "dashboard" și a permite accesul doar utilizatorilor autentificați, am verificat lista de token-uri (local și session), în cazul în care acestea exista (ceea ce înseamnă că utilizatirul este , sau a fost, înregistrat) atunci o sa se poată face redirecționarea la dashboard, în caz contrar (nu se găsesc tokeni) atunci utilizatorul va fi redirecționat către pagina principala de login/signup.

*Baza de date* a acestei aplicații a fost făcută local folosind un fisier JSON în care am stocat informațiile necesare despre filme pe care am dorit sa le afișam în dashboard.

***Utilizare aplicatie***

Utilizarea aplicației poate fi descrisă în funcție de diferitele tipuri de utilizatori și interacțiunile lor specifice. Mai jos, am pregătit o descriere pentru două tipuri de utilizatori: utilizatorul neautentificat și utilizatorul autentificat, pentru a evidenția diferențele în funcționalitățile și vizualizările disponibile.

*Utilizatorul neautentificat:* Atunci când un utilizator neautentificat accesează aplicația, acesta va avea la dispoziție pagina de autentificare și înregistrare. Utilizatorul poate introduce datele de autentificare (email și parolă) pentru a accesa funcționalitățile aplicației. Dacă nu are un cont, poate utiliza opțiunea de înregistrare pentru a crea un cont nou. După completarea formularului de autentificare sau înregistrare, utilizatorul este redirecționat automat către pagina de panou de control (dashboard).

*Utilizatorul autentificat:* După autentificare, utilizatorul are acces la întreaga gamă de funcționalități oferite de aplicație (vizualizare, editare, stergere, adaugare). Acesta este redirecționat către pagina de panou de control (dashboard), unde poate interacționa cu următoarele elemente:

* *Vizualizarea listei de filme:* Utilizatorul poate vedea o listă de filme disponibile. Această listă include informații precum titlul, genul, descrierea, directori, actori și altele. Filmele sunt prezentate sub forma unui tabel (cate 5 intrari pe o pagina). Acestia o sa aiba la dispozitie si un poster al filmului in partea stanga a titlului.
* *Căutarea de filme:* Utilizatorul poate utiliza funcția de căutare pentru a găsi un film specific în lista de filme disponibile. Utilizatorul va introduce numele filmului in bara de cautare dupa care acesta va primi un raspuns daca filmul a putut fi gasit sau nu.
* *Adăugarea de filme noi:* Utilizatorul poate adăuga filme noi în lista afisata pe dashboard. Poate completa informațiile despre film, cum ar fi titlul, genul, descrierea și altele, într-un formular dedicat. După adăugarea cu succes a unui film, acesta va fi inclus în lista de filme disponibile.
* Editarea și ștergerea filmelor: Utilizatorul poate edita informațiile existente ale unui film sau le poate șterge complet din lista de filme. Prin intermediul interfeței aplicației, utilizatorul poate accesa opțiunile de editare și ștergere a unui film, permițându-i să actualizeze sau să elimine informațiile dorite. Aceste actiuni sunt puse in dreptul filmului in partea dreapta a tabelului ca niste iconite pentru un still mai compact al aplicatiei.
* *Sortarea filmelor:* Utilizatorul are posibilitatea de a sorta filmele dupa anul in care acestea au fost publicate

***Concluzie***

În concluzie, aplicația propusă bazată pe Angular oferă utilizatorilor o platformă web eficientă și prietenoasă pentru gestionarea unei liste de filme. Cu funcționalități precum autentificare, vizualizare a listei de filme, căutare, adăugare, editare și ștergere, aplicația aduce un plus de comoditate utilizatorilor săi.

Prin intermediul unei arhitecturi bine structurate, aplicația asigură o navigare clară și ușoară între diferitele componente și funcționalități. Componenta de autentificare permite utilizatorilor să-și creeze conturi noi sau să se autentifice cu datele existente, asigurând astfel confidențialitatea și securitatea datelor. După autentificare, utilizatorii sunt redirecționați către pagina de panou de control, unde pot interacționa cu lista de filme.

O funcționalitate importantă este adăugarea, editarea și ștergerea filmelor. Utilizatorii pot personaliza lista de filme prin adăugarea de noi titluri, actualizarea informațiilor existente și eliminarea filmelor nedorite. Aceste funcționalități permit o gestionare flexibilă și completă a listei de filme, asigurându-se că utilizatorii au control total asupra conținutului.

Aplicația oferă o interfață intuitivă și ușor de utilizat, facilitând navigarea și accesarea funcționalităților dorite. Componentele bine definite și organizate în cadrul structurii aplicației permit o experiență coerentă și plăcută utilizatorilor.

De asemenea, mecanismul de autentificare asigură că doar utilizatorii autorizați au acces la aplicație, protejând astfel informațiile personale și evitând accesul neautorizat.