



Universitatea *Transilvania* din Braşov
Facultatea de Matematică şi Informatică
Departamentul de Matematică şi Informatică

Proiect Angular

Autori: Bondoc Daniel, Faliboga Dimitrie
Informatică Aplicată, Anul II
Grupa 10LF331
2025

My Trips

Introducere

Proiectul *My Trips* a fost dezvoltat în cadrul cursului de Angular din anul 2 de studiu al programului de Informatică Aplicată de la Universitatea Transilvania Braşov.

Tema aleasă pentru aplicaţie se concentrează pe explorarea şi organizarea de excursii şi trasee pentru drumeţii sau mers cu bicicleta, oferind utilizatorilor un mediu interactiv pentru planificarea aventurilor în natură.

Într-o societate tot mai digitalizată, aplicaţiile care facilitează activităţile în natură devin tot mai populare, încurajând un stil de viaţă activ şi conectat cu mediul înconjurător.

My Trips vine în întâmpinarea acestei nevoi, permiţând utilizatorilor să creeze, editeze, sorteze şi exploreze trasee montane sau cicloturistice, într-un mod intuitiv şi accesibil.

Aplicaţia oferă funcţionalităţi esenţiale precum autentificarea utilizatorilor (login/register), validarea datelor introduse, salvarea sesiunilor şi accesul diferenţiat în funcţie de autentificare. În plus, prin intermediul unei interfeţe prietenoase bazate pe biblioteca UI *NgZorro*, aplicaţia oferă o experienţă plăcută şi coerentă de utilizare. Scopul principal al proiectului este de a îmbina elementele fundamentale ale dezvoltării în Angular cu o idee practică şi atractivă din viaţa reală.

Proiectul are şi o componentă educaţională importantă, prin aplicarea conceptelor de lazy loading, servicii, validatori personalizaţi, @Input/@Output, manipularea unei liste de date şi utilizarea semnalelor. Astfel, *My Trips* nu este doar o aplicaţie web funcţională, ci şi o demonstraţie practică a principiilor moderne de arhitectură şi dezvoltare frontend.

Autentificare şi Înregistrare

Una dintre componentele centrale ale aplicaţiei *My Trips* este sistemul de autentificare, care include funcţionalităţi de tip *Login* şi *Register*. Această funcţionalitate oferă utilizatorilor o modalitate securizată şi intuitivă de a accesa aplicaţia şi a-şi gestiona traseele.

Componenta de login permite autentificarea pe baza unei adrese de email şi a unei parole. Un element opţional, dar important, este checkbox-ul *Remember Me*, care asigură păstrarea sesiunii active prin salvarea token-ului în *localStorage* sau *sessionStorage*, în funcţie de alegerea utilizatorului. Formularul validează datele introduse şi oferă feedback în cazul în care structura parolei nu respectă cerinţele minime.

Formularul de înregistrare este compus din câmpuri pentru prenume, nume, email, parolă și confirmarea parolei. S-a implementat un validator personalizat pentru complexitatea parolei, asigurând existența unei litere mari, a unei litere mici, a unei cifre și a unui caracter special, precum și un validator pentru confirmarea corectă a parolei.

Pentru persistarea datelor utilizatorilor, aplicația se conectează la un API personalizat creat pe platforma `mockapi.io`, accesibil la adresa: `https://683870c12c55e01d184d5142.mockapi.io/mytripsapi/Users`. Cererile de tip POST sunt trimise pentru înregistrare și autentificare, iar în răspuns este returnat un token, care este apoi stocat și utilizat în aplicație pentru gestionarea accesului.

Interfața vizuală este construită folosind biblioteca *NgZorro*, ceea ce contribuie la o experiență consistentă, modernă și responsivă. Formularele sunt reactive, construite cu `FormBuilder`, și oferă validări în timp real.

Prin această abordare, aplicația asigură o bază solidă de autentificare, adaptabilă și ușor de extins cu funcționalități suplimentare precum resetarea parolei sau profilul utilizatorului.

Protecția Accesului prin Auth Guards

Accesul la diverse rute ale aplicației *My Trips* este restricționat printr-un mecanism de **route guarding**, utilizând interfața `CanActivate` oferită de Angular. Această abordare este esențială pentru menținerea unui control strict asupra componentelor accesibile utilizatorilor autentificați vs. vizitatorilor.

AuthOnlyGuard

`AuthOnlyGuard` este responsabil pentru protejarea rutelor rezervate utilizatorilor autentificați. Acesta verifică existența unui token de autentificare în `localStorage` sau `sessionStorage`. Dacă tokenul este absent, utilizatorul este redirecționat către pagina de înregistrare:

```
const token = localStorage.getItem('auth_token') || sessionStorage.getItem('auth_token');
```

Astfel, rutele critice precum `/features/home` sau `/features/explore` sunt inaccesibile fără autentificare validă.

GuestOnlyGuard

Complementar, `GuestOnlyGuard` permite accesul la paginile de *login* și *register* doar în absența unei sesiuni active. În cazul în care un utilizator deja autentificat accesează manual o astfel de rută, acesta este redirecționat automat către secțiunea principală a aplicației:

```
this.router.navigate(['/features/explore']);
```

Integrarea în sistemul de rutare

Rutele sunt definite în `app.routes.ts`, folosind **lazy loading** pentru a îmbunătăți performanța aplicației și a reduce timpul inițial de încărcare. Fiecare rută importantă este asociată cu un `guard` specific:

- `/auth/login` și `/auth/register` sunt protejate de `GuestOnlyGuard`.
- `/features/explore` și `/features/home` sunt protejate de `AuthOnlyGuard`.

Astfel, securitatea aplicației este gestionată direct la nivel de rutare, prevenind accesul neautorizat și reducând riscurile legate de afișarea accidentală a componentelor sensibile.

Utilitate

Prin utilizarea unor **guards** bine definite, aplicația aplică principiile de **separație a accesului**, **securitate la nivel de rută** și **minimizare a suprafeței de expunere**, oferind în același timp o experiență controlată și predictibilă pentru utilizatori, indiferent de starea sesiunii lor.

Serviciul TrailService

Serviciul responsabil cu logica de gestionare a datelor despre trasee este **TrailService**. Acesta oferă funcționalități de bază pentru manipularea unei liste de trasee, stocate local în aplicație (nu în baza de date). Serviciul include metode pentru generarea de ID-uri unice, adăugarea unui traseu nou, obținerea unui traseu după ID, ștergerea și actualizarea unui traseu existent. Datele sunt gestionate reactiv, folosind un mecanism de tip *signal*, care permite actualizarea automată a componentelor care consumă această listă.

Formularul de adăugare/editare traseu

Componenta **AddTrailFormComponent** gestionează formularul pentru adăugarea sau editarea unui traseu. Formularul este construit folosind **FormGroup** și **FormControl** din Angular, cu validări pentru fiecare câmp (de exemplu, titlul și locația trebuie să aibă minim 3 caractere, descrierile au limite minime de caractere, iar coordonatele trebuie să fie în intervale geografice valide).

La trimiterea formularului, dacă datele sunt valide, se apelează metodele din serviciu pentru a salva sau actualiza traseul. Dacă formularul este invalid, toate câmpurile sunt marcate ca *dirty* pentru a evidenția erorile utilizatorului.

Modalul de detalii traseu

Componenta **TrailDetailsModalComponent** este folosită pentru afișarea detaliilor unui traseu într-o fereastră modală. Aceasta primește un obiect de tip **Trail** și afișează informații detaliate, precum imaginea, locația, tipul și descrierea traseului. Modalul poate fi închis de utilizator, iar acțiunea este logată în consolă.

Pagina principală de management al traseelor

Componenta **TrailsPageComponent** este pagina principală care orchestrează afișarea, căutarea, sortarea și manipularea traseelor. Aceasta:

- Încarcă lista de trasee din serviciu și o păstrează atât în forma completă, cât și filtrată (pentru căutare și sortare).
- Permite căutarea rapidă după titlu, locație sau tip de traseu.
- Oferă sortare ascendentă sau descendentă după diverse câmpuri.
- Deschide modale pentru vizualizarea detaliilor, adăugarea sau editarea unui traseu, folosind componentele dedicate.
- Permite ștergerea unui traseu cu confirmare prealabilă.
- Actualizează automat lista afișată după orice operație de CRUD.

Interfața utilizator

Interfața este construită folosind componente din **ng-zorro-antd** pentru tabele, formulare, butoane și modale, ceea ce asigură o experiență vizuală modernă și responsivă. Utilizatorul poate adăuga un traseu nou, poate căuta și sorta traseele, poate vizualiza detalii suplimentare, edita sau șterge un traseu, toate aceste acțiuni fiind disponibile direct din interfața principală.

Validare și feedback

Formularul de adăugare/editare oferă feedback instant utilizatorului privind validitatea datelor introduse, afișând mesaje de eroare clare pentru fiecare câmp invalid. Acest lucru asigură introducerea unor date corecte și complete, reducând riscul de erori.