CAHIER DES CHARGES du projet Angular 20+

Youth Leadership Tracker

GUIDARA AZIZ & KAROUI ZEINEB

Application de gestion et de suivi des expériences de leadership des membres AIESEC

1. Introduction

Le projet *Youth Leadership Tracker* est une application web avec **Angular 20+**, permettant de gérer les membres d'un comité local AIESEC ainsi que leurs expériences de leadership (Team Member, Team Leader, OC, VP…).
L'objectif est de fournir un outil simple, structuré et conforme aux exigences pédagogiques, tout en reflétant les valeurs d'engagement et progression personnelle.

L'application se compose d'un module d'authentification, de deux modules CRUD complets, d'un tableau de bord analytique, et d'un espace profil.

2. Périmètre et objectifs du projet

L'application doit permettre :

Gestion des membres

- Création, modification, suppression d'un membre
- Consultation de son profil
- Liste des membres filtrable, triable et paginée

Gestion des expériences de leadership

- Création, édition, suppression d'une expérience (TL, OC, VP…)
- Liaison d'une expérience à un membre
- Liste filtrée et triée (par rôle, département, dates…)
- Page de détails

Tableau de bord statistique

- Nombre total de membres
- Nombre total d'expériences
- Pourcentage de leadership par département
- Skills les plus développées
- Expériences en cours / expirées

Profil utilisateur

- Affichage et édition du profil personnel
- Stockage local simulé

Authentification

- Login avec email + mot de passe
- Système de 3 tentatives maximum
- Guard + token simulé
- Verrouillage temporaire du compte

3. Architecture fonctionnelle

3.1 Modules fonctionnels

L'application est organisée en **5 modules principaux** :

1. **AuthModule**
2. **MembersModule** (CRUD #1)
3. **ExperiencesModule** (CRUD #2)
4. **DashboardModule**

5. **ProfileModule**

3.2 Pages obligatoires incluses

1. Page de Login

- Formulaire reactive/template
- Validators : required, email, minLength
- Message d'erreur + blocage après 3 tentatives
- Token simulé via service

2. Page de Liste (CRUD Lecture)

Pour les membres **et** les expériences :

- Table + cartes
- @for
- Recherche + filtres
- Tri (nom, rôle, dates…)
- Pagination
- Service Angular pour fetch local

3. Page de Formulaire (CRUD Create/Update)

- Reactive Forms
- Validators : required, email, minLength, pattern
- Mode création / édition
- Messages utilisateurs propres
- Redirection après succès

4. Page de Détails (CRUD Read-One)

- Routing dynamique /members/:id et /experiences/:id
- ActivatedRoute + service
- Gestion de l’erreur si id invalide ("Not found")

4. Architecture technique

Standalone Components

Tous les composants seront développés en mode standalone.

Services injectés via `inject()`

- AuthService
- MembersService
- ExperienceService
- DashboardService

Interfaces TypeScript

- Member
- Experience
- Skills enum

Routing modulaire

- Lazy loading conseillé (load pages **only when needed** instead of loading all modules when the app starts)
- Routes paramétrées

5. Description des entités

5.1 Member (Utilisateur)

Champ	Type	Description
id	number	Identifiant
fullName	string	Nom complet
email	string	Email AIESEC
department	string	TM / OGX / ICX / ER / etc
age	number	Facultatif
skills	string[]	Soft skills (teamwork, comm…)

5.2 Leadership Experience

Champ	Type	Description
id	number	Identifiant
role	string	TL / OC / VP…
department	string	TM, OGX…
description	string	Missions
startDate	Date	Début
endDate	Date	Fin
skillsGained	string[]	Compétences développées
memberId	number	Lien avec un membre

6. Tableau de bord

Statistiques affichées :

- **Total membres**
- **Total expériences**
- **% de TL / VP / OC/ EST**

- **Top compétences développées**
- **Répartition par département**
- **Expériences actives vs terminées**

Charts dynamiques (Angular + lib simple).

7. Choix techniques

- **Angular 20+** pour standalone components et performance
- **Reactive Forms** pour le CRUD d'expériences (validation complexe)
- **Template-driven** pour un petit formulaire simple (ex: login)
- **Signals** pour synchroniser le profil
- **Local storage** pour les données de session
- **TailwindCSS / Material** pour un design moderne et responsive

8. Qualité UI/UX

- Responsive sur mobile et desktop
- Messages d'erreur clairs
- Feedback utilisateur (loading, success, errors)
- Couleurs douces inspirées d'AIESEC (bleu, blanc, violet)

9. Contraintes et limites

- Pas de vraie API backend : simulation locale
- Authentification non réelle (mock)
- Dashboard basé sur des données locales

10. Conclusion

Le projet *Youth Leadership Tracker* répond pleinement aux exigences du module Angular grâce à son architecture complète, son découpage modulaire, et ses deux CRUDs complets.