ORION



Justification des choix techniques ***Projet MDD***



Auteur : [Hugo Charon]

Version 0.0.1

[Aperçu / Synthèse 3](#_Toc197699087)

[Choix techniques 3](#_Toc197699088)

[Choix 1 Angular 3](#_Toc197699089)

[Choix 2 Angular Material 4](#_Toc197699090)

[Choix 3 ngx-cookie-service 4](#_Toc197699091)

[Choix 4 RxJS 4](#_Toc197699092)

[Choix 5 TypeScript 5](#_Toc197699093)

# 

# Aperçu / Synthèse

*Dans le cadre de ce projet, nous avons pris des décisions clés concernant les technologies et outils à utiliser pour la construction de l’application front-end. Nous avons opté pour Angular comme framework principal, en raison de sa robustesse et de son large écosystème. Angular fournit des outils pour la gestion des composants, des formulaires, et de l’interaction avec des services backend. Il s'aligne avec les contraintes techniques imposées, telles que la nécessité d'une application modulaire et maintenable.*

*Nous avons également intégré Angular Material pour la gestion de l’interface utilisateur, ce qui permet de gagner en rapidité dans la création d'interfaces modernes et réactives, tout en respectant les principes de design Material. Pour la gestion des cookies, nous avons choisi la librairie ngx-cookie-service afin de faciliter la manipulation des cookies côté client.*

*Enfin, pour l’intégration continue et les tests unitaires, nous avons opté pour des outils bien établis comme Jasmine et Karma.*

# Choix techniques

## Choix 1 Angular

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | **Choix technique** |  |  | | --- | |  | | |  | | --- | | **Lien vers la documentation / ressource** |  |  |  | | --- | --- | |  |  | | |  | | --- | | **But du choix** |  |  | | --- | |  | |
| Angular (Framework principal) | <https://angular.dev/guide/routing/router-tutorial> | |  | | --- | |  |  |  | | --- | | Architecture de l’application, gestion des composants, routage, gestion des formulaires | |

| **Justification du choix technique** |
| --- |
|  |

Angular est un framework robuste et largement adopté, parfaitement adapté pour construire des applications web modernes et performantes. Nous l'avons choisi pour sa modularité, son support natif pour TypeScript, et sa large communauté qui garantit une bonne documentation et des mises à jour régulières.

## Choix 2 Angular Material

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | **Choix technique** |  |  | | --- | |  | | |  | | --- | | **Lien vers la documentation / ressource** |  |  |  | | --- | --- | |  |  | | |  | | --- | | **But du choix** |  |  | | --- | |  | |
| Angular Material (UI Library) | <https://material.angular.io/> | |  | | --- | |  |  |  | | --- | | Design de l’interface utilisateur, composants UI préfabriqués | |

| **Justification du choix technique** |
| --- |

|  |
| --- |
|  |

Angular Material permet de créer des interfaces modernes en respectant les principes de Material Design. Il propose une large gamme de composants réutilisables, permettant de gagner du temps sur la conception visuelle et de garantir une interface cohérente et responsive.

## Choix 3 ngx-cookie-service

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | **Choix technique** |  |  | | --- | |  | | |  | | --- | | **Lien vers la documentation / ressource** |  |  |  | | --- | --- | |  |  | | |  | | --- | | **But du choix** |  |  | | --- | |  | |
| ngx-cookie-service (Gestion des cookies) | <https://www.npmjs.com/package/ngx-cookie-service> | |  | | --- | |  |  |  | | --- | | Gestion des cookies côté client, stocker et récupérer des informations sur l’utilisateur | |

| **Justification du choix technique** |
| --- |

ngx-cookie-service est une librairie légère et facile à utiliser, permettant de manipuler les cookies dans Angular. Elle offre des méthodes simples pour gérer les cookies de manière sécurisée, ce qui est crucial pour la gestion des sessions utilisateur.

## Choix 4 RxJS

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | **Choix technique** |  |  | | --- | |  | | |  | | --- | | **Lien vers la documentation / ressource** |  |  |  | | --- | --- | |  |  | | |  | | --- | | **But du choix** |  |  | | --- | |  | |
| RxJS (Gestion des flux asynchrones) | <https://rxjs.dev/> | Gestion des événements asynchrones, des flux de donnéesréactifs |

| **Justification du choix technique** |
| --- |

RxJS permet de gérer des flux de données de manière réactive et asynchrone, facilitant ainsi la gestion des appels HTTP et des états de l’application grâce aux observables.

## Choix 5 TypeScript

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | **Choix technique** |  |  | | --- | |  | | |  | | --- | | **Lien vers la documentation / ressource** |  |  |  | | --- | --- | |  |  | | |  | | --- | | **But du choix** |  |  | | --- | |  | |
| TypeScript (Langage principal) | <https://www.typescriptlang.org/docs/> | |  | | --- | |  |  |  | | --- | | Renforcer la sécurité du code, amélioration de la lisibilité et maintenabilité | |

| **Justification du choix technique** |
| --- |
|  |

TypeScript ajoute des types statiques à JavaScript, ce qui améliore la sécurité du code et aide à éviter de nombreuses erreurs au moment de la compilation. Il permet de détecter les erreurs plus tôt et offre des fonctionnalités d’autocomplétion dans les IDE.