

# Documentation technique du projet

## Réflexions initiale technologique sur le sujet

Pour le développement du frontend, j'ai choisi d'utiliser JavaScript, HTML, SCSS ainsi qu'un framework CSS, Bootstrap. Ce dernier a été un atout précieux pour accélérer le développement, grâce à ses composants préconçus et ses outils de mise en page responsive.

En ce qui concerne le backend, j'ai opté pour le framework NestJS, car au moment où je devais commencer l'ECF, je ne maîtrisais pas encore suffisamment le framework Symfony. Ayant déjà terminé les modules de JavaScript et de TypeScript et étant libre quant à la technologie à adopter, le choix de NestJS m'a semblé idéal. Cette décision se justifie également par le fait que NestJS est un framework Node.js simple à utiliser, rapide à apprendre et facile à installer avec une excellente documentation. Il s'appuie sur TypeScript, mais reste utilisable en JavaScript.

## Configuration de l'environnement de travail

Voici les principaux outils et technologies utilisés dans le cadre de ce projet, répartis entre les parties frontend et backend :

### Frontend :

- **Visual Studio Code (VS Code)** : IDE utilisé pour le développement du code.
- **JavaScript (JS)** : Langage de programmation utilisé pour la logique côté client.
- **HTML** : Langage de balisage utilisé pour structurer le contenu des pages web.
- **SCSS** : Préprocesseur CSS utilisé pour styliser l'interface frontend avec des fonctionnalités avancées.
- **Bootstrap** : Framework CSS utilisé pour le développement de l'interface frontend.
- **Node.js** : Environnement d'exécution en JavaScript et gestion des dépendances avec npm.

### Backend :

- **Visual Studio Code (VS Code)** : IDE utilisé pour le développement du code.
- **Node.js** : Environnement d'exécution en JavaScript et gestion des dépendances avec npm.
- **NestJS** : Framework Node.js utilisé pour le développement du backend.
- **TypeORM** : ORM (Object-Relational Mapping) utilisé pour interagir avec la base de données relationnelle
- **XAMPP** : Environnement local pour héberger la base de données MySQL.
- **MongoDB** : Base de données NoSQL pour la gestion des données non relationnelles.
- **MongoDB Compass** : Interface graphique pour interagir avec MongoDB.

- **Nodemailer** : Module Node.js utilisé pour l'envoi d'emails depuis le backend.

## Les dispositions prises sur la sécurité

Pour assurer une sécurité renforcée sur l'application, plusieurs dispositions ont été mises en place. Tout d'abord, les API sont sécurisées à l'aide de **tokens d'accès** (access tokens), garantissant ainsi que seules les requêtes authentifiées peuvent interagir avec les ressources protégées. De plus, pour éviter les attaques par devinette ou modification d'ID, j'ai opté pour l'utilisation de **UUIDs (identifiants uniques universels)** comme clés primaires dans les bases de données, remplaçant ainsi les clés auto-incrémentées traditionnelles. Cela rend les identifiants moins prévisibles et plus difficiles à manipuler. Enfin, côté frontend, j'ai mis en place une gestion stricte des **rôles utilisateurs**, garantissant que chaque utilisateur n'a accès qu'aux pages et ressources qui correspondent à son rôle.