

**Binôme 7 : Boudrouss Réda (0G78YV02AY2) / Mouine Youssef (150391620JF)**

## Choix de modélisation

Notre site web utilise les librairies principales suivantes :

- Typescript (langage)
- Nextjs (en backend)
- React & React server side components (en “frontend”)
- Prisma (Object Relational Mapping)
- Postgresql

### Postgresql

- Pourquoi pas une DB noSQL ?

Nous avons commencé par programmer la backend d’abord au début du semestre vu que cela nous a semblé plus logique. Ayant pris les deux UEs de base de donnée relationnelle et étant beaucoup plus familier avec les bases de données SQL nous avons fait ce choix et avancé avec.

Sans cahier des charges complets, nous avons appris bien plus tard (lors du TME solo) qu’une base de donnée noSQL était demandé, nous avons donc essayer de passer à mongodb mais pas à temps (vous pouvez trouver notre tentative (**non fonctionnelle**) de passer à une database mongo dans la branch `mongodb`).

### Prisma

Choix logique si on part sur une base de donnée relationnelle. Permet de faire le lien entre une base de donnée et notre code, s’occupe de l’initialisation, de la génération, et des requêtes pour nous.

Prisma permet aussi de traduire notre schema intuitif de base de donnée (trouvable sur `prisma/schema.prisma`) en initialisation SQL.

### Typescript

Typescript rajoute un typage fort au Javascript et cela est utile pour l’expérience développeur sur plusieurs points.

Dans un premier temps on ajoute une meilleur gestions des `null` et des `undefined`. Si le typage est fait correctement, nous savons à tout moment quel type est dans chaque variable, surtout que Javascript est connu pour avoir des fonctions au comportement assez imprévisible. Assez utile pour éviter une grande partie des bugs.

Cela permet aussi une meilleur intégration dans l’interface de développement. En effet sachant les types de chaque objet, nous avons des meilleurs recommandation de la part de l’auto-complétion de vscode (en incluant la gestion des `null` et `undefined`).

### Nextjs

Vu que l’on connaissait déjà Expressjs, nous avons décidé d’essayer une nouvelle framework en backend pour apprendre de nouvelles choses.

Nextjs est une framework fullstack, elle nous permet de gérer à la fois la frontend et la backend. Typiquement, une bonne partie du routage est gérée nativement grâce aux chemins d'accès des différents fichiers créés. Sa feature phare ceci dit est la pré-génération de la frontend côté serveur (avec un système de caching) à l'aide des composants server side de React, ce qui donne un aspect beaucoup plus réactif du site (et pleins d'autres avantages).

## Répartition des tâches

Noms : Boudrouss Réda et Youssef Mouine

Nous avons souvent bossé ensemble sur les mêmes features en même temps, y avait pas souvent une distinction de qui faisait quoi. Ceci dit, quand on essayait d'intégrer une nouvelle feature, on essayait de coder le côté frontend et backend en même temps. Youssef s'occupait assez souvent de la frontend et Réda de la backend.

Pour la vidéo, les diapos et le script ont été faits ensemble, Réda a filmé la vidéo.

Nous avons rencontré quelques difficultés pour faire la vidéo ce qui explique le retard du rendu, en effet Réda travaille à temps partiel à côté des études, il a été compliqué de trouver un temps pour travailler ensemble surtout vers cette période chargée.