Résumé du projet :

"Api" Salle De Sport

Dans le cadre de ma Formation Développeur Full stack en ligne via la plateforme STUDI, j'ai réalisé une application de salle de sport pour mettre en place un panel d'administration des partenaires et structure de la marque FitPark Fitness.

FitPark Fitness est une application WEB permettant une administration complète des Partenaires et Structures liés a la marque. Le back-office est accessible par les administrateurs, Partenaires et structures. Seuls les administrateurs peuvent modifier, créer, activer, désactiver , ajouter des permissions ou supprimer des Partenaires ou Structures. Les partenaires et structures on seulement un accès a la plateforme en mode lecture only.

Pour ce projet, nous avions que la partie administration a créé , mon choix pour la mise en place du projet s'est porté sur le framework REACT 18.2.0. Pour la partie back-end mon choix s'est porté sur Node.js, pour la base de données RGPD j'ai décidé d'utilisé POSTGRESQL hébergée sur ElephantSql .

Pour stylisé le tout j'ai utilise TAILWINDCSS.

Pour effectuer les relations entre la base de données et les requêtes entre le Front-End et le Back-end, j'ai mis en place une API avec Express qui est simple et rapide et qui fournit une performance robuste de fonctionalités pour les applications Web et mobiles . Enfin pour finir j'ai utilisé Firebase pour ça facilité de déploiement de mon application coté Front-End. Pour le back-end le service cloud-function de Firebase.

Dans un second temps, il m'as fallu mettre en place un système d'envoi d'email pour que le partenaire ou la structure puisse recevoir un lien qui leur permettrait de changer leur mot de passe avant leurs toutes premières connexions et qui sera mis à jour en temps réel dans la base de données.Il y avait aussi un email pour souhaiter la bienvenue avec un bouton de redirection vers le panel de connexion et un dernier email étaient envoyer aux partenaires et à la Structure liée lors de la création de celle-ci ainsi que pour toute modification des persmissions.

Enfin, pour sécuriser l'application, j'ai utilisé Bcrypt pour crypter les mots de passe, mis en place d'une connexion par rôle (Admin , Partenaires, Structure) ne pouvant voir que leur contenu ainsi que de la mise en place du JWT Token lors de l'authentification qui pourront donnés l'accès aux ressources.

Puis sur les formulaires une validation des inputs grâce a une libraire qui s'appelle React-Hook-Form

qui permet d'avoir des formulaires performants et une validation facile à utiliser permettant ainsi aussi d"eviter des attaques par injections en limitant le nombre de caractères et en utilisant des regex pour l'email et mot de passe.