MySQL :

MySQL est un système de gestion de base de données relationnelle open source, réputé pour ses performances, sa fiabilité, sa sécurité, et sa communauté active. Il offre une solution économique avec une documentation abondante, une compatibilité étendue, et une évolutivité pour gérer des volumes de données croissants.

Node.js :  
Node.js est un environnement d'exécution JavaScript côté serveur, apprécié pour ses performances élevées, son évolutivité, son écosystème vaste avec npm, et sa popularité dans le développement rapide. Il excelle dans les applications en temps réel et dispose d'une large communauté de développeurs.

React :

React, une bibliothèque JavaScript, favorise la construction d'interfaces utilisateur avec des composants réutilisables. Ses avantages incluent un Virtual DOM pour des performances optimisées, une gestion unidirectionnelle des données, et un écosystème dynamique avec une forte communauté de support. Utilisé largement, il est aussi compatible avec React Native pour le développement d'applications mobiles.

Fonction de connexion/inscription

Une image contenant texte, capture d’écran, logiciel

Description générée automatiquement

Cette fonction envoie une requête qui compare le mail et le mot de passe rentré dans le formulaire de login, et suite à cela si les deux correspondent on génère un token avec les informations de l’utilisateur ( sauf le mot de passe)  
  
Une image contenant texte, capture d’écran

Description générée automatiquement

Pour l’inscription la fonction récupère les informations du formulaire d’inscription, on vérifie si cet utilisateur existe déjà a partir de son mail, on va ensuite crypter son mot de passe (10 fois) à l’aide du module bcrypt et on va effectuer une requête insert Into en générant un uuid qui permet de générer des id aléatoires sécurisés.

Le site présente aussi une gestion de produits. On a par exemple l’ajout de celui-ci   
Une image contenant texte, capture d’écran

Description générée automatiquement

On récupère du formulaire les données voulus et on créé un produit en envoyant ces nouvelles informations dans la Base de données

Modification d’un produit :   
Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

Cette fonction envoie un update du produit qui prend en compte les informations des produits, et qui les modifies ensuite une fois le formulaire envoyé

Le delete :   
Une image contenant texte, capture d’écran, logiciel

Description générée automatiquement

Le delete lui va récupérer un id par son produit et supprimer celui-ci.

Dans les fonctions précédente on peut aussi voir la présence de gestions d’erreurs si quelque chose n’aboutit pas correctement.

Le fichier server.js lui permet de créer des raccourcis pour la connexion à la BDD   
Une image contenant texte, capture d’écran

Description générée automatiquement

Les données de pool sont stockés dans le .env et les routes font appels à d’autres routes qui vont chercher les différentes fonctions du CRUD dans les controllers

Pour la BDD on a deux tables, Produits et Clients qui comprend en fait tout les utilisateurs ainsi que les admins, le rôle permet de trier leurs accès, notamment grâce aux fonctions isLoggedIn et isAdmin dans le components Navbar :

Une image contenant texte, capture d’écran, logiciel

Description générée automatiquement

On verifie si un token a été generé ce qui signifie que un utilisateur est connecté et pour l’autre cas on verifie, grâce au cookie generé précédemment si il est admin (c’est une valeur binaire 1 estAdmin, 0 ne l’est pas mais lors de la creation de compte 0 est assigné, les admins sont créés manuellement.

Partie BDD :



Dans la table client, on a nom, prenom, mail, mdp, isAdmin

Dans la classe produit on a nom, prix, image, description, stock