Documentation technique

Sommaire

[1) Outils utilisés 1](#_Toc162297907)

[2) Présentation du code 2](#_Toc162297908)

[3) La base de données 6](#_Toc162297909)

# **Outils utilisés**

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.

# **Présentation du code**

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 rentrés dans le formulaire de login. Par la suite, si les deux sont équivalents un token est généré, contenant 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, puis on vérifie si cet utilisateur existe déjà grâce à son mail. Son mot de passe va ensuite être crypté (10 fois) à l’aide du module bcrypt , puis on va effectuer une requête « Insert into » en générant un uuid, ce qui permet d’obtenir des id aléatoires et sécurisés.

Fonctions de gestion des produits

Une image contenant texte, capture d’écran

Description générée automatiquement

Cette fonction permet d’ajouter un nouveau produit. On récupère les données voulues depuis le formulaire puis on créer un produit, en envoyant ces informations dans la base de données.

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

Cette fonction permet la modification d’un produit. Elle envoie une mise à jour du produit, prenant en compte les informations saisies dans les champs et met à jour la base de données une fois le formulaire envoyé.

Une image contenant texte, capture d’écran, logiciel

Description générée automatiquement

Cette fonction permet de supprimer un produit, en récupérant ce dernier par son id puis va le supprimer.

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

Connexion à la base de données

Le fichier server.js permet de créer des raccourcis pour la connexion à la BDD.

Une image contenant texte, capture d’écran

Description générée automatiquement

Les données de la constante pool sont stockées dans le fichier .env, et les routes font appel à d’autres routes qui vont chercher les différentes fonctions du CRUD dans les controllers

Fonctions d’état

Une image contenant texte, capture d’écran, logiciel

Description générée automatiquement

La base de données contient deux tables, Produits et Utilisateurs. La table utilisateur contient tous les clients ainsi que les administrateurs, avec un rôle permettant de gérer leurs accès, notamment grâce aux fonctions isLoggedIn et isAdmin dans le composant « Navbar » :

On vérifie si un token a été généré, ce qui signifie qu’un utilisateur est connecté. Pour l’autre cas on vérifie, grâce au cookie généré précédemment s’il est admin (il s’agit d’une valeur binaire, 1 s’il est admin et 0 s’il ne l’est pas. Lors de la creation de compte un 0 est assigné en tant que valeur par défaut, les rôles d’administrateur sont créés manuellement.

# **La base de données**

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Dans la table Utilisateur il y a les champs id, nom, prenom, mail, mdp et isAdmin.

Dans la table Produit il y a les champs id, nom, prix, image, description et stock.

Dans la table Commande il y a les champs id\_produit, id\_utilisateur et quantité.