

Objectif : Création d'une application de gestion de produits avec système de notation, utilisant une base de données MariaDB et une architecture en services et répertoires

 Propriétaire	 Marine
 Étiquettes	

Contexte

Vous allez créer une application de gestion de produits qui permet de gérer des produits et leurs notations. Chaque produit peut avoir une note qui est calculée sur la base de plusieurs critères comme la qualité, le prix et l'utilité. Vous devrez interagir avec une base de données MariaDB pour stocker et récupérer ces données.

Tâches à réaliser

1. Création de la base de données et des tables :

- Vous devez créer une base de données MariaDB nommée **Ratings** si elle n'existe pas déjà.
- Ensuite, vous devez créer deux tables principales :
 1. **Table **products**** : Cette table contiendra des informations sur les produits (nom, description, etc.).
 2. **Table **ratings**** : Cette table stockera les évaluations des produits, avec des colonnes pour la qualité, le prix et l'utilité.

- Chaque produit aura un identifiant unique (`product_id`), et chaque évaluation sera liée à un produit via une clé étrangère (`product_id`).

2. Création de l'architecture de l'application :

```
- src/
  - controllers/
    - product.controller.js
  - services/
    - product.service.js
    - rating.service.js
  - repositories/
    - product.repository.js
    - rating.repository.js
  - db/
    - connection.js
    - migrations/
      - create_tables.sql
  - utils/
    - logger.js
  - routes/
    - product.routes.js
  - index.js
  - .env
  - package.json
```

3. Création de la connexion à la base de données :

- Utilisez un pool de connexions pour établir une connexion efficace et sécurisée à votre base de données MariaDB. Ce pool doit être configuré avec des informations de connexion telles que l'hôte, le nom d'utilisateur, le mot de passe, et le nom de la base de données. Le pool n'est défini qu'une seule fois et exporté.

4. Gestion des erreurs :

- Vous devez gérer les erreurs de connexion à la base de données, en affichant un message d'erreur compréhensible en cas de problème.
- Assurez-vous de vérifier la validité des identifiants avant d'exécuter des requêtes, pour éviter des erreurs lorsque vous tentez d'interagir avec

des produits qui n'existent pas.

5. Insertion de données initiales :

- Pour tester votre application, vous devrez insérer des données dans la base de données.
- Bonus : vous pouvez utiliser une transaction pour insérer des produits et des évaluations de manière globale : si une des insertions échoue, l'ensemble des modifications doit être annulé.
- Assurez-vous que vos données sont cohérentes (par exemple, un produit ne doit pas avoir de notation avant d'être créé).

6. Fonctionnalités à implémenter :

- **Affichage des produits** : Créez une fonctionnalité qui permet de lister tous les produits dans la base de données.
- Ajout d'un produit
- **Ajout d'une évaluation** : Implémentez une fonctionnalité pour ajouter une évaluation à un produit donné, avec les critères de qualité, prix et utilité.
- **Calcul du score pondéré** : Créez une fonctionnalité qui calcule un score pondéré pour chaque produit basé sur ses évaluations. Vous devrez utiliser un service pour cette logique, qui prendra en compte les pondérations de chaque critère.

7. Tests et validation :

- Une fois que tout est mis en place, vous devez tester l'application pour vous assurer que chaque fonctionnalité fonctionne correctement.
- Effectuez des tests manuels pour vérifier si les produits sont bien ajoutés, si les évaluations sont correctement associées aux produits et si les scores pondérés sont calculés correctement.