FormakBac Adresse : Montplaisir Imm. Espace Tunis, Bloc H, 5ème étage.	**Devoir de contrôle n°2** **(ExpressJs NextJs typescript)** Durée : 2 **heures**	**Enseignant :** Younes Manita **Classe :** Niveau 5 **Date : 03**/09/2024

Objectif : Développement d'une application Type Script avec Express.js pour gérer une liste de produits

Votre mission est de développer une application web en utilisant Express.js avec Type Script pour gérer une liste de produits dans une base de données PostgreSQL. L'application devra permettre d'ajouter, de lister, de filtrer, de mettre à jour et de manipuler les données des produits en utilisant les méthodes JavaScript `map` et `filter`.

Étapes du projet :

1. Configuration de l'environnement :

- Créez une nouvelle application Express.js avec TypeScript.
- Configurez la connexion à une base de données PostgreSQL nommée **`test`**.
- Créez une table nommée **`products_db`** pour stocker les informations des produits.

2. Création de la table `products_db` :

- La table `products_db` doit avoir les champs suivants :
  - **`id`** : entier, clé primaire, auto-incrémenté.
  - **`name`** : chaîne de caractères, non nul.
  - **`price`** : nombre décimal, non nul.
  - **`category`** : chaîne de caractères, non nul.
  - **`inStock`** : booléen, indiquant si le produit est en stock ou non.
  - **`quantity`** : entier, représentant la quantité disponible en stock, non nul.

3. Implémentation des Routes :

Routes à implémenter :

1. **Ajouter un produit :**
    - **Route :** `POST /products`
    - Permet d'ajouter un nouveau produit à la base de données.
2. **Lister tous les produits :**
    - **Route :** `GET /products`
    - Affiche tous les produits enregistrés dans la base de données.
3. **Filtrer les produits par catégorie :**
    - **Route :** `GET /products/category/:category`
    - Retourne les produits appartenant à une catégorie spécifique en utilisant `filter`.
4. **Lister les noms de produits :**
    - **Route :** `GET /products/names`
    - Retourne une liste des noms de tous les produits en utilisant `map`.
5. **Filtrer les produits par prix minimum :**
    - **Route :** `GET /products/above/:price`
    - Retourne les produits dont le prix est supérieur à une valeur spécifiée en utilisant `filter`.
6. **Mettre à jour la quantité en stock :**
    - **Route :** `Put /products/:id/update-stock`
    - Si la quantité est supérieure à 0, le champ `inStock` doit être automatiquement mis à `true`.
    - Si la quantité est égale à 0 , le champ `inStock` doit être mis à `false`.

7. **Supprimer un produit :**
    - **Route :** `DELETE /products/:id`
    - Supprime un produit par son ID de la base de données.

Démarrage du serveur :

- Initialisez un projet Node.js avec `npm init -y`.
- Installez les dépendances nécessaires : `express`, `pg` (client PostgreSQL),`@types/pg`, et `dotenv` pour la gestion des variables d'environnement.
- Installez les types et outils TypeScript : `typescript`, `ts-node`, `@types/node`, `@types/express`.

- Configurez TypeScript en créant un fichier `tsconfig.json` avec les configurations de base pour le développement TypeScript
- Ajoutez un script de démarrage dans votre `package.json` pour lancer l'application en utilisant `ts-node`.

Bon Courage