

# Exercice : Application de Gestion de Bibliothèque en ligne

## Contexte :

Vous avez été embauché pour développer une application de gestion de bibliothèque en ligne. L'application devrait permettre d'ajouter des livres à la base de données, de les afficher, de les mettre à jour et de les supprimer.

## Étapes :

### 1. Configuration de la base de données MySQL :

- Créez une base de données nommée `bibliotheque`.
- Ajoutez une table `livres` avec les colonnes suivantes :
  - `id` (auto-incrémenté, clé primaire)
  - `titre` (VARCHAR)
  - `auteur` (VARCHAR)
  - `annee_publication` (INT)
  - `disponible` (BOOL)

### 2. Création du modèle de livre :

- Créez un fichier `models/livre.js`.
- Définissez un modèle de livre qui utilise la connexion à la base de données et expose des méthodes pour effectuer les opérations CRUD (Create, Read, Update, Delete) sur la table `livres`.

### 3. Création du contrôleur de livres :

- Créez un fichier `controllers/livreController.js`.
- Implémentez des fonctions de contrôleur pour gérer les actions CRUD en utilisant le modèle de livre.

### 4. Configuration d'Express.js :

- Dans le fichier principal `app.js`, configurez une application Express.js.

- Utilisez le middleware `body-parser` pour pouvoir traiter les données JSON.
- Importez le routeur des livres (que vous créerez à l'étape suivante).

## 5. Création du routeur des livres :

- Créez un fichier `routes/livreRoutes.js`.
- Configurez les routes CRUD pour les livres en utilisant le contrôleur de livres.

## 6. Test de l'application :

- Ajoutez des livres à la base de données en utilisant la route d'ajout.
- Affichez tous les livres en utilisant la route de lecture.
- Mettez à jour un livre en utilisant la route de mise à jour.
- Supprimez un livre en utilisant la route de suppression.

## 7. Validation :

- Assurez-vous que toutes les fonctionnalités sont opérationnelles.
  - Testez la robustesse de l'application en gérant les erreurs possibles (par exemple, livre non trouvé).
-