

Projet : eLibrary (Bibliothèque Numérique)

Objectif :

Développer une application eLibrary avec **React, Redux Toolkit, React Router et Bootstrap**, permettant de gérer des livres numériques, filtrer, noter et gérer un panier.

Structure d'un livre :

```
{  
  id: number,  
  title: string,  
  author: string,  
  category: string,  
  style: string,  
  description: string,  
  rating: number, // 1 à 5  
  price: number,  
  coverUrl: string  
}
```

Fonctionnalités obligatoires :

1. **Pages / Routes :**
 - / : Liste des livres avec filtres (catégorie, auteur, rating).
 - /book/:id : Détail d'un livre.
 - /cart : Panier avec gestion des quantités.
2. **Redux :**
 - booksSlice : filtrer, modifier rating, ajouter au panier.
 - cartSlice : incrementer, decrementer, supprimer, vider le panier.
3. Analysez votre code avec SonarQube et proposez des améliorations pour la qualité, la sécurité et la maintenabilité.

Projet : eLibrary avec API

Objectif :

Développer une application eLibrary avec React, Redux Toolkit, React Router et Bootstrap, permettant de gérer des livres numériques. Cette version doit consommer une API externe pour récupérer, ajouter ou modifier des livres, et gérer le panier.

Structure d'un livre :

```
{  
  id: number,  
  title: string,  
  author: string,
```

```
category: string,  
style: string,  
description: string,  
rating: number, // 1 à 5  
price: number,  
coverUrl: string  
}
```

Fonctionnalités obligatoires :

1. Pages / Routes :

- / : Liste des livres avec filtres (catégorie, auteur, rating).
- /book/:id : Détail d'un livre.
- /cart : Panier avec gestion des quantités.

2. Redux Toolkit avec createAsyncThunk :

booksSlice

- **Actions asynchrones :**

- fetchBooks → Récupérer la liste des livres depuis l'API.
- updateBookRating → Modifier la note d'un livre via l'API.
- addBookToCart → Ajouter un livre au panier côté backend si nécessaire.

- **Actions synchrones :**

- filterBooks → Filtrer par catégorie, auteur, rating.
- sortBooks → Trier par titre, date ou rating.

cartSlice

- Actions synchrones :

- incrementQuantity(bookId)
- decrementQuantity(bookId)
- removeFromCart(bookId)
- clearCart()

3. Analysez votre code avec **SonarQube** et proposez des améliorations pour :

- La qualité du code
- La sécurité
- La maintenabilité