

Projet : Boutique en Ligne - ShopEase

Contexte

Vous êtes chargé de développer une application de boutique en ligne moderne et sécurisée permettant à des utilisateurs de consulter des produits, de les ajouter à un panier et de passer une commande. L'application sera composée d'une API développée avec Symfony (back-end) et d'un front-end interactif réalisé avec React. **Le panier sera géré localement dans le navigateur grâce au LocalStorage ou via Redux.**

Objectifs

1. Back-end avec Symfony

Créer une API RESTful qui gère :

- **Produits :**
 - Création, modification, suppression et consultation des produits.
- **Catégories :**
 - Gestion des catégories de produits.
- **Utilisateurs :**
 - Inscription, connexion (authentification via JWT ou OAuth2), et gestion des informations personnelles.
- **Commandes :**
 - Enregistrement et suivi des commandes (les données du panier seront envoyées à l'API lors de la validation de la commande).

Autres fonctionnalités :

- Implémenter un système de validation des données avec Symfony Validator.
 - Utiliser Doctrine ORM pour gérer les entités et leurs relations (Produit, Catégorie, Utilisateur, Commande).
 - Gestion des erreurs avec des codes HTTP appropriés.
 - Documentation API avec **Swagger** ou **API Platform Docs**.
-

2. Front-end avec React

Créer une interface utilisateur ergonomique permettant :

- **Produits :**
 - Consultation des produits et recherche par catégorie, nom ou prix.
 - Affichage des détails d'un produit.
- **Panier :**
 - Ajout de produits au panier.
 - Supprimer des produits du panier.
 - Supprimer le panier.

- **Gestion avec LocalStorage Ou via Redux** : sauvegarde des articles, quantités et informations directement dans le navigateur.
- Modification et suppression d'articles du panier avec mise à jour dans le LocalStorage.
- **Utilisateur** :
 - Création de compte et connexion.
- **Commandes** :
 - Validation d'une commande avec envoi des données du panier à l'API.

Autres fonctionnalités :

- Navigation avec **React Router**.
 - Gestion des appels API avec **Axios** ou **fetch**.
-

3. Base de données

Utiliser une base de données relationnelle (MySQL ou PostgreSQL) pour modéliser les entités suivantes :

- **Utilisateur** :
 - email, mot de passe, etc.
 - **Produit** :
 - nom, description, prix, stock, image, catégorie.
 - **Catégorie** :
 - nom.
 - **Commande** :
 - produits, utilisateur, statut, date.
-