

Test Symfony

Objectif :

Développer une application contenant un catalogue de produits et une gestion de panier basique.

Indications :

- L'application propose 12 produits au total.
- Les produits contiennent les champs suivants : id, nom, description, prix.
- Les produits seront stockés en base de donnée.
- Le panier sera stocké en session.

Écrans :

- Page d'accueil : afficher la totalité des produits ordonnés par nom, un bouton "voir la fiche" permet d'aller sur la page de détails du produit.
- Page de détails d'un produit : afficher les informations du produit ainsi qu'un bouton permettant l'ajout au panier, avec le choix de la quantité (lors de l'ajout d'un produit au panier, si celui-ci est déjà présent il faudra mettre à jour sa quantité, et non pas le dupliquer).
- Page du panier : détailler le contenu du panier, afficher son montant total et un bouton permettant de vider le panier.
- Sur toutes les pages : afficher le nombre d'articles présents dans le panier dans un encart dédié.

Exigences :

- Projet Symfony 4/5
- PHP >= 7.3
- SQLite
- ---
- Conforme aux bonnes pratiques et standards de code de Symfony
- Mise en forme simple avec le framework Bootstrap 4 et gestion des assets avec Symfony Encore
- ---
- Fixtures pour charger les produits en base de données
- Migrations de base de données
- ---
- Test unitaire du panier : tester le calcul du montant total
- Test fonctionnel de la page d'accueil : tester le code de la réponse et le nombre de produits affichés

- ---
 - Rédiger et fournir une documentation sous la forme d'un fichier README indiquant comment :
 - installer le projet
 - charger les fixtures
 - compiler les assets
 - lancer le projet
 - jouer les tests
 - ...
 - Héberger le code sur Github et nous envoyer le lien du repository
-

Fonctionnalités avancées :

- Fournir une configuration Docker / Docker Compose pour faire tourner le projet
- ---
- Installer un espace d'administration permettant de gérer les produits (*easyadmin*)
- Sécuriser cet espace d'administration (utilisateur "admin" avec pour mot de passe "password" de type *in_memory*)
- ---
- Produit : ajouter une photo visible sur les pages d'accueil et de détails
- Produit : ajouter un slug qui servira à afficher la page de détails
- ---
- Page panier : pouvoir modifier les quantités
- Page panier : pouvoir supprimer un produit
- ---
- Proposer un point d'API renvoyant l'intégralité des produits au format JSON
- ---
- Proposer un export au format CSV de l'intégralité des produits depuis une commande console dédiée
- ---
- Permettre au visiteur de choisir la langue du site : fr ou en (les produits et les urls ne sont pas à traduire, seule l'interface est à adapter)
- ---
- Tests plus complets