Documentation du site e-commerce Stubborn réalisé avec Symfony.

Sommaire :

Installation :

Cloner le dépôt :

```
git clone https://github.com/eihtac/Stubborn
cd stubborn
```

Installer les dépendances :

```
composer install
```

Créer et configurer le fichier .env.local :

Créer un fichier .env.local à la racine du projet et y définir les paramètres suivants :

```
DATABASE_URL=
MAILER_DSN=
STRIPE_SECRET_KEY=
STRIPE_PUBLIC_KEY=
```

Créer la base de données :

```
php bin/console doctrine:database:create
php bin/console doctrine:migrations:migrate
```

Charger les données test :

```
php bin/console doctrine:fixtures:load
```

Lancer le serveur local :

```
symfony server:start
```

Authentification :

Le site utilise le système de sécurité de Symfony avec deux rôles d'utilisateurs :

- ROLE_CLIENT : utilisateurs clients du site
- ROLE_ADMIN : administrateurs ayant accès au back-office

Inscription

L'inscription se fait via un formulaire accessible depuis la page d'accueil. Une fois inscrit, l'utilisateur reçoit un email de confirmation. Il doit cliquer sur le lien contenu dans cet email pour activer son compte. Tant que l'adresse n'est pas confirmée, il ne peut pas se connecter.

Sécurité

L'accès aux routes sensibles est protégé par des règles définies dans security.yaml (rôle client ou admin)

Produits :

Sur la page d'accueil, les sweats-shirts mis en avant sont visibles par tous les visiteurs, même non connectés.
Une fois connecté, l'utilisateur peut accéder à une page produits, listant tous les sweat-shirts disponibles et permettant de les filtrer par fourchettes de prix. Chaque produit a également une page de détails, permettant de choisir une taille parmi celles disponibles en stock et de l'ajouter au panier.

Panier :

Chaque utilisateur possède un panier actif avec le statut cart.
Une fois la commande réglée, le statut passe à order.
Le prix total du panier est calculé dynamiquement via la méthode getTotal() de l'entité Cart.

Commande :

Le paiement est géré via Stripe Checkout en mode test.
Lors du passage de commande, une session Stripe est créée automatiquement à partir du contenu du panier.
Une fois le paiement validé, le panier passe du statut cart à order et met à jour les stocks des produits commandés.

Administration :

L'interface d'administration n'est accessible qu'aux utilisateurs ayant le ROLE_ADMIN.
Elle permet d'ajouter un nouveau produit, ou de modifier ou supprimer un produit existant.

Tests :

Deux tests unitaires ont été réalisés avec PHPUnit pour vérifier le bon fonctionnement du panier et du processus de commande :

Test Panier :

Crée deux produits fictifs avec des prix différents
Les ajoute au panier via des CartItems
Assertion sur le total attendu.

Test Paiement :

Simulation de paiement
Crée un panier contenant un article
Modifie le stock du produit dans la taille commandée
Passe du statut cart à order
Vérifie le stock décrémenté et le statut mis à jour.