

1. Introduction

Présentation du Projet

Stubborn Shop est une plateforme e-commerce permettant aux utilisateurs d'acheter des vêtements et accessoires en ligne. L'application propose un espace administrateur pour gérer les produits et commandes, ainsi qu'une interface utilisateur intuitive pour naviguer et acheter des articles.

Objectifs de l'Application

- Permettre aux utilisateurs de consulter et acheter des produits.
- Offrir un espace administrateur pour gérer les produits et commandes.
- Assurer une gestion fluide du panier et des paiements.

Public Cible

- Utilisateurs souhaitant acheter des vêtements en ligne.
 - Administrateurs et gestionnaires de boutique e-commerce.
-

2. Installation & Configuration

Prérequis

- PHP 8.1 ou supérieur
- Symfony 6
- Composer
- MySQL ou MariaDB
- Node.js et NPM (pour l'intégration front-end)

Installation du Projet

1. Cloner le dépôt :

```
git clone https://github.com/mon-projet/stubborn-shop.git
cd stubborn-shop
```

2. Installer les dépendances PHP :

```
composer install
```

3. Installer les dépendances front-end :

```
npm install
npm run build
```

4. Configurer l'environnement :

- Copier le fichier `.env.example` en `.env`
- Modifier les variables de connexion à la base de données.

5. Créer la base de données et exécuter les migrations :

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

6. Lancer le serveur Symfony :

```
symfony serve
```

Ou avec PHP :

```
php -S 127.0.0.1:8000 -t public
```

3. Architecture du Projet

Structure des Dossiers

```
project-root/
├── src/
│   ├── Controller/
│   ├── Entity/
│   ├── Form/
│   ├── Repository/
│   ├── Security/
│   ├── Service/
│   ├── Templates/
│   ├── Public/
│   └── Assets/
├── migrations/
├── config/
├── public/
├── var/
└── vendor/
```

- **Controller/** : Contient les fichiers gérant les routes et la logique métier.
- **Entity/** : Contient les modèles de base de données.
- **Form/** : Contient les formulaires Symfony.
- **Repository/** : Gère l'accès aux données.
- **Templates/** : Contient les vues Twig.
- **Public/** : Contient les fichiers accessibles publiquement (images, CSS, JS).

4. Utilisation de l'Application

Navigation et Fonctionnalités

- **Page d'accueil** : Présente les produits en vedette.

- **Page Boutique** : Affiche tous les produits disponibles.
- **Page Panier** : Liste les produits ajoutés au panier avec un bouton de paiement.
- **Page Commandes** : Affiche l'historique des commandes.

Gestion des Utilisateurs

- **Connexion / Inscription.**
- **Gestion des rôles (Administrateur, Utilisateur).**

Gestion des Produits

- Ajout, modification et suppression des produits.
- Gestion des stocks et mise en avant.

Gestion du Panier

- Ajout / Suppression de produits.
 - Validation et paiement sécurisé.
-

5. Développement & API

Routes Principales

- **Utilisateur**
 - `/register` : Inscription
 - `/login` : Connexion
 - `/logout` : Déconnexion
- **Produits**
 - `/product/` : Liste des produits
 - `/product/{id}` : Détail d'un produit
 - `/cart/` : Gestion du panier
- **Administration**
 - `/admin/products/` : Gestion des produits
 - `/admin/orders/` : Gestion des commandes

Services Utilisés

- **Doctrine** : Gestion de la base de données.
- **Twig** : Moteur de templating.

- **Security Bundle** : Gestion des utilisateurs et des rôles.
-

6. Déploiement

Préparation pour la mise en production

- Configurer les variables d'environnement en `.env.prod`.
- Exécuter `composer install --no-dev`.
- Générer les assets front-end avec `npm run build`.
- Configurer un serveur web (Apache/Nginx) pour pointer vers `public/`.

Sécurité

- HTTPS obligatoire.
 - Configuration des permissions des fichiers sensibles.
 - Protection contre les injections SQL et XSS.
-

7. Support & Contact

Liens Utiles

- Documentation Symfony : <https://symfony.com/doc/current/>
- Documentation Doctrine : <https://www.doctrine-project.org/>

Contact

- Email : stubborn@blabla.com
- Adresse : Piccadilly Circus, London W1J 0DA, Royaume-Uni