



Stubborn Shop

Documentation technique & utilisateur

Framework	Symfony (PHP)
Front	Twig + Bootstrap 5 + Webpack Encore
DB	MySQL / MariaDB (Doctrine ORM)
Email	Mailer (Mailtrap Sandbox en dev)
Sécurité	Authentification + UserChecker + rôles

Résumé :

Application e-commerce de démonstration (produits, panier par tailles, commande, compte, vérification d'e-mail).

Documentation du site **Stubborn Shop**



1. Introduction

Cette application a été développée avec **Symfony 6/7**, Doctrine ORM et Twig.

Elle constitue une base de boutique en ligne minimaliste permettant de :

- lister des produits,
- gérer un **panier avec tailles** (S, M, L, XL),
- valider une commande,
- administrer les produits (ajout, mise à jour, suppression).

2. Architecture de l'application

2.1 Technologies

Symfony (Framework PHP moderne MVC)

Doctrine ORM (gestion base de données MySQL/PostgreSQL)

Twig (moteur de templates)

Bootstrap 5 (design et responsivité)

Session Symfony (stockage du panier)

2.2 Organisation des dossiers

- **config/** # Config Symfony et Doctrine
- **public/** # Point d'entrée web (index.php)
- **src/**
 - **Controller/** # Contrôleurs MVC (Cart, Product, Order...)
 - **Entity/** # Entités Doctrine (Product, User...)
 - **Repository/** # Requêtes Doctrine custo
 - **Security/** # Authentification
- **templates/** # Vues Twig (cart, product, base...)
- **assets/css/** # Feuilles de style (base.css, thème)
- **migrations/** # Scripts DB



3. Modèle de données

3.1 Entité Product

- ✓ id (int, PK)
- ✓ name (string)
- ✓ description (text)
- ✓ price (float)
- ✓ stock_s, stock_m, stock_l, stock_xl (int)

3.2 Entité User

- ✓ id (int, PK)
- ✓ email (string)
- ✓ roles (array)
- ✓ password (hashed string)

3.3 Entité Order

- ✓ id (int, PK)
- ✓ user (relation ManyToOne)
- ✓ items (relation OneToMany avec OrderItem)
- ✓ total (float)
- ✓ createdAt (datetime)

3.4 Entité OrderItem

- ✓ product (ManyToOne)
- ✓ size (string)
- ✓ quantity (int)
- ✓ price (float)



4. Fonctionnalités principales

4.1 Gestion du panier

- ✓ Ajout d'un produit avec une taille choisie (/cart/add/{id})
- ✓ Mise à jour des quantités (/cart/update/{id}/{size})
- ✓ Suppression d'un article (/cart/remove/{id}/{size})
- ✓ Vider le panier (/cart/clear)
- ✓ Vérification stricte du stock par taille

4.2 Interface utilisateur

- ✓ Page d'accueil + boutique (/products)
- ✓ Panier (/cart)
- ✓ Page commande (/order)
- ✓ Système de flash messages pour les retours utilisateurs

4.3 Sécurité

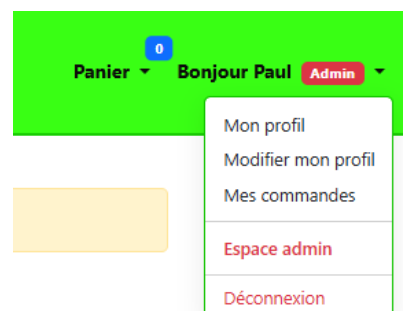
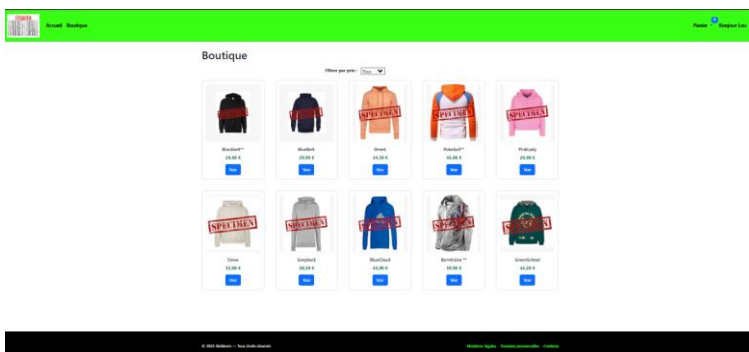
- ✓ Authentification via symfony/security
- ✓ Création d'un administrateur :

```
php bin/console app:create-admin
```

- ✓ Accès restreint pour l'administration

5. Interface graphique

- ✓ Navbar verte (fluo, thème Stubborn)
- ✓ Footer noir avec liens légaux (mentions légales, RGPD, contacts)
- ✓ Boutons normalisés grâce à la classe CSS .btn-eq pour alignement
- ✓ Dropdown de Bootstrap 5. Il déclenche l'ouverture du menu.





6. Hébergement et exécution

6.1 Hébergement local

1. Installer PHP 8.2+, Composer, Symfony CLI
2. Cloner le projet :

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

3. Installer les dépendances :

```
composer install  
npm install && npm run build
```

4. Configurer .env avec la base de données (ex. MySQL)
5. Créer la base :

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

6. Lancer le serveur :

```
symfony serve -d
```

L'application est disponible sur <http://127.0.0.1:8000>



7. Évolutions possibles

- Paiement en ligne (Stripe / PayPal)
- Gestion des stocks en temps réel
- Espace membre (historique commandes)
- Administration complète avec EasyAdminBundle

8. Conclusion

L'application Stubborn Shop fournit une base fonctionnelle de boutique e-commerce Symfony.

Elle met en place les bonnes pratiques :

- séparation MVC,
- gestion sécurisée des sessions,
- vérification stricte des stocks,
- UX claire avec Bootstrap.

Ce projet est évolutif et il peut servir de socle à un site e-commerce complet.