


# Plateforme E-Commerce de Sweatshirts

 GitHub Repository : [marine1512/symfonymm](https://github.com/marine1512/symfonymm)

---



## Présentation

Ce projet propose une **boutique en ligne spécialisée dans les sweatshirts**. Il permet aux utilisateurs de :





- parcourir les produits disponibles,
- les ajouter à un **panier dynamique**,
- finaliser leur achat via un **paiement sécurisé** avec **Stripe Checkout**.

Le projet repose sur **Symfony 6**, une base de données **MySQL**, et intègre **Stripe** pour la gestion des paiements.

---



## Objectifs

-  Navigation fluide pour les utilisateurs connectés
  -  Page d'accueil avec produits vedettes
  -  Système de panier interactif
  -  Paiement sécurisé via **Stripe Checkout**
- 



## Fonctionnalités



### Connexion / Inscription

- Création de compte
- Connexion sécurisée avec vérification des identifiants



### Page d'accueil

- Affichage de produits en promotion (image, description, prix)



### Boutique

- Liste complète des sweatshirts

- Filtrage par **plage de prix**

## **Palement Stripe**

- Validation du panier avec redirection vers Stripe Checkout
- Palement simulé avec **clés de test Stripe**

---








## **Modèle Conceptuel de Données (MCD)**

### **Principales entités**

- **User** : ID, email, mot de passe
- **Product** : nom, description, prix, image
- **Order** : date, statut, utilisateur
- **OrderLine** : produit, quantité, commande

---

## **Technologies utilisées**

Composant	Technologie
 Langage	PHP 8.1+
 Framework	Symfony 6
 Base de données	MySQL 8.0
 Palement	Stripe
 Service Mail	Mailtrap
 Tests	PHPUnit
 Serveur Dev	Symfony CLI intégré

---

## **Installation**

### **1 Cloner le projet**

```
git clone https://github.com/marine1512/symfonymm.git
cd symfonymm
```

### **2 Installer les dépendances**

```
composer install
```

### **3 Configurer les variables d'environnement**

Créer un fichier .env.local :

```
APP_ENV=dev
APP_SECRET=...
```

```
DATABASE_URL="mysql://utilisateur:motdepasse@127.0.0.1:3306/nom_base"
```

```
STRIPE_API_SECRET="sk_test_..."
STRIPE_API_PUBLIC="pk_test_..."
```

## 4 Créer et migrer la base de données


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

## 5 Charger les fixtures (produits, utilisateurs, etc.)

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

## 6 Lancer le serveur Symfony

```
symfony server:start
```

 Application disponible sur : <https://127.0.0.1:8000>

---

## Tester un paiement Stripe

Utilisez les informations de test suivantes :

- **Carte** : 4242 4242 4242 4242
- **Expiration** : 12/34
- **CVC** : 123

**STRIPE\_SECRET\_KEY**=sk\_test\_51ROfCc2a6UM81evrhHbUMI5tdXcoroX0WZSnXmLOLJn3uTfEyR  
XxYELbZ3f5OcNUZ2E5y3FI304Qupc4z1GIPWp900Htq0nD74

**STRIPE\_PUBLIC\_KEY**=pk\_test\_51ROfCc2a6UM81evrHoC8F4IK7B2oqFe8BqNWGJFItsFYUHCeMyr  
9YAyAjAlrHMaMs0SdKDUHboY2BUwGvnhGJluL00vEslhrs

---

## Tests unitaires avec PHPUnit

Installer PHPUnit :

```
composer require --dev phpunit/phpunit
```

### **Lancer un test spécifique :**

```
./vendor/bin/phpunit tests/Service/CartServiceTest.php
```

---