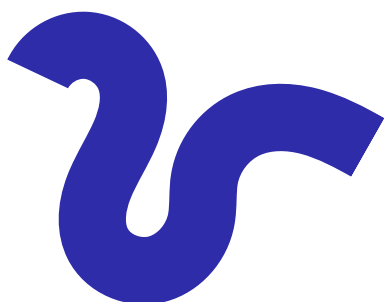




# CAHIER DES CHARGES

PRÉSENTÉ PAR LE GROUPE 5



## SOMMAIRE

**01** ICO!

**02** Notre équipe

**03** Benchmark technique

**04** Gestion de projet et organisation

**05** Développement et mise en production


**06** Mise en production et tests



# A QUOI SERT LE PROJET ICO!



PRÉSENTER LE  
CONCEPT ET  
LES RÈGLES DU  
JEU.



OFFRIR UN  
BACKOFFICE  
COMPLET  
POUR LA  
GESTION DES  
CONTENUS ET  
DES  
COMMANDES.



PROPOSER UN  
SYSTÈME DE  
VENTE EN LIGNE.

## NOTRE ÉQUIPE



**MAREVA**  
cheffe de  
projet digital



**THOMAS**  
frontend  
developer



**NOAH**  
backend  
developer



**YOANN**  
backend  
developer

# LE BENCHMARK TECH

Critères	Facilité d'apprentissage	Performance	Sécurité	Scalabilité	SEO-friendly	Temps de développement
Stack retenue (HTML, CSS, Twig, JS, Bootstrap, Symfony)	Facile pour les développeurs back-end PHP ayant des bases front-end	Bonne optimisation grâce à Symfony (cache, Doctrine ORM)	Symfony offre des protections robustes (CSRF, XSS, SQLi)	Facile à maintenir et à faire évoluer avec une architecture modulaire	Excellent avec Twig et Bootstrap (HTML/CSS structurés)	Optimisé grâce aux outils fournis par Symfony (CRUD, templates Twig)
Alternatives (React, Angular, Node.js, Express)	Plus complexe à maîtriser (JS framework complet)	Haute performance avec SPAs, mais plus gourmand	Bonne sécurité, nécessite plus de configurations	Haute scalabilité, mais nécessite plus de gestion	SEO correct mais nécessite SSR (Next.js par ex.)	Plus long en raison du setup API+Front séparé
Alternatives (WordPress, WooCommerce)	Très simple mais peu flexible	Moins performant à grande échelle	Vulnérable sans plugins de sécurité	Moins flexible pour des évolutions	SEO performant avec des plugins	Rapide mais avec des limites en personnalisation

## POLITIQUE DE GESTION DES DONNÉES

### Anonyme

- Accès Vitrine
- Accéder au shop
- Passer une commande
- Créer un compte

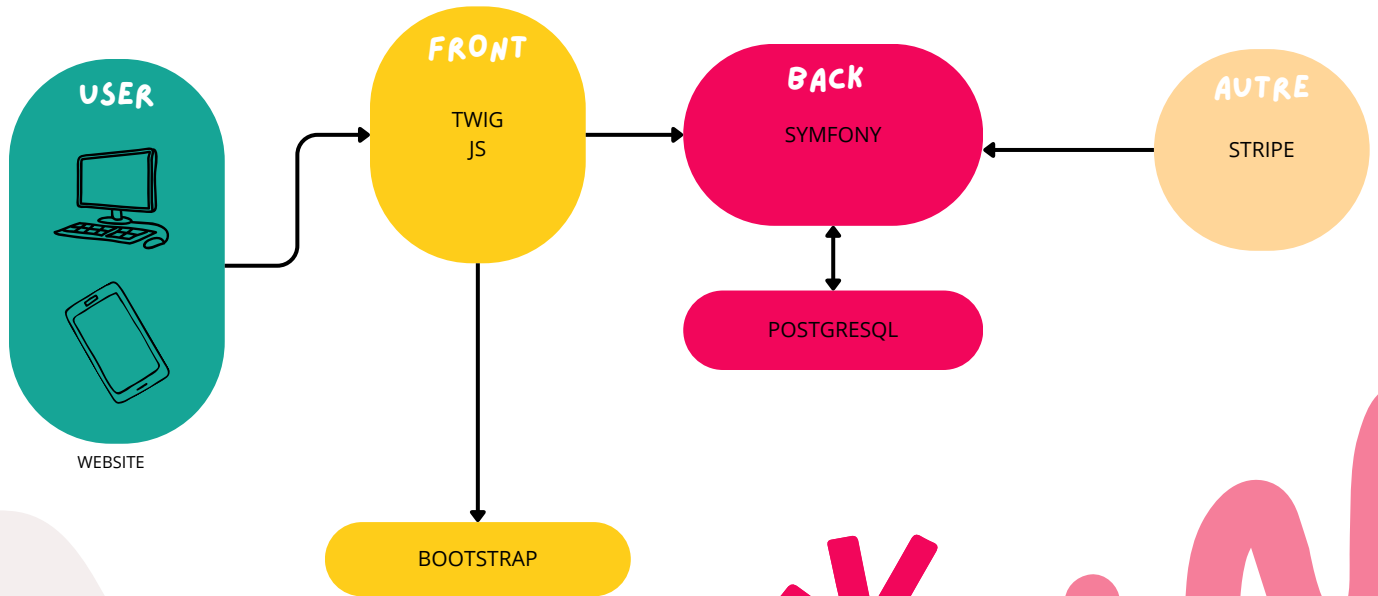
### Utilisateur

- Accès Vitrine
- Accéder au shop
- Passer une commande
- Créer un compte
- Voir l'historique des commandes
- Modifier le profil
- Voir ma commande

### Administrateur

- Accès Vitrine
- Accéder au shop
- Passer une commande
- Créer un compte
- Voir l'historique des commandes
- Modifier le profil
- Voir ma commande
- Accès au BO
- Gérer les utilisateurs
- Gérer les produits
- Gérer les commandes
- Gérer les promotions ou remises
- Accéder au dashboard
- Voir toutes les commandes

# ARCHITECTURE HARDWARE ET SOFTWARE



## JUSTIFS DES CHOIX TECHNOLOGIQUES

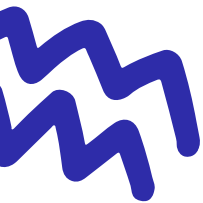
HTML, CSS,  
BOOTSTRAP

JAVASCRIPT

SYMFONY

TWIG

MÉTHODOLOGIE  
KANBAN AVEC GIT



HTML, CSS,  
BOOTSTRAP

HTML, CSS, Bootstrap (Frontend)

Standards du web garantissant une compatibilité optimale.  
Bootstrap offre une approche responsive et facilite le développement UI.

TWIG

Twig (Moteur de template)

Séparation claire entre logique et présentation.  
Sécurité renforcée grâce à l'échappement automatique.

JAVASCRIPT

JavaScript (Dynamisme et interactivité)

Ajout de fonctionnalités interactives sans rechargement de page.  
Compatibilité avec les bibliothèques modernes.

SYMFONY

Symfony (Backend robuste)

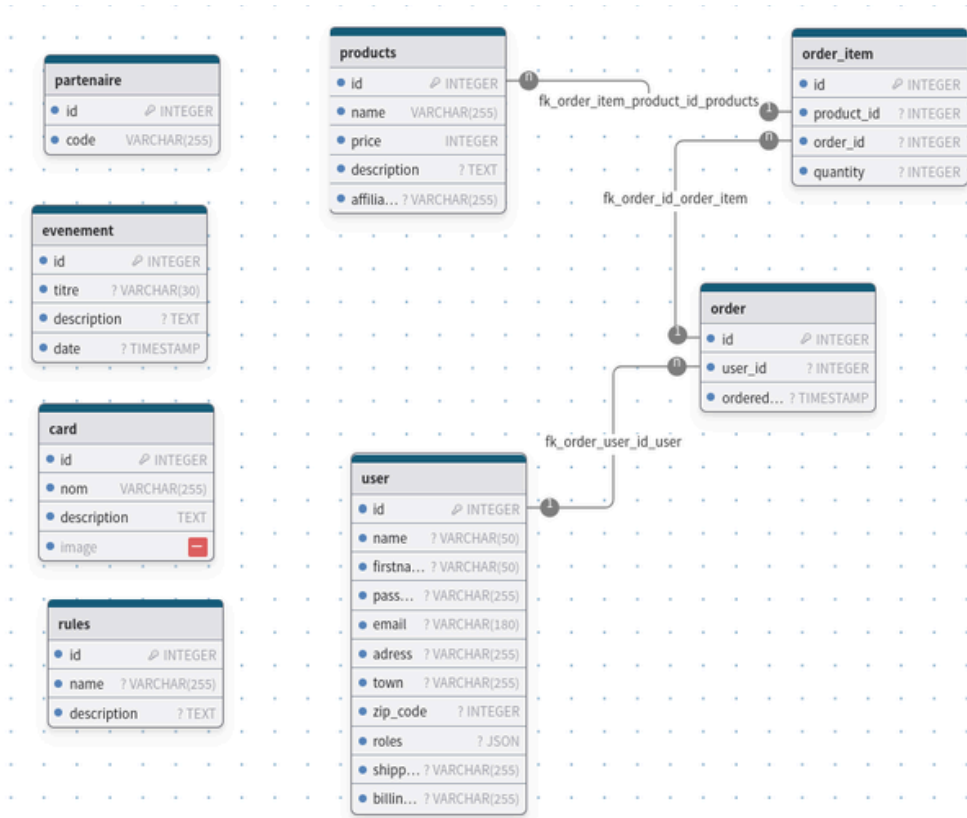
Framework PHP offrant des outils de sécurité et de gestion avancés.  
Respect des bonnes pratiques MVC pour une meilleure évolutivité.

MÉTHODOLOGIE  
KANBAN AVEC GIT

Git & Kanban (Gestion de projet)

Suivi des tâches en temps réel via GitHub Projects.  
Kanban permet une gestion visuelle et itérative des priorités.

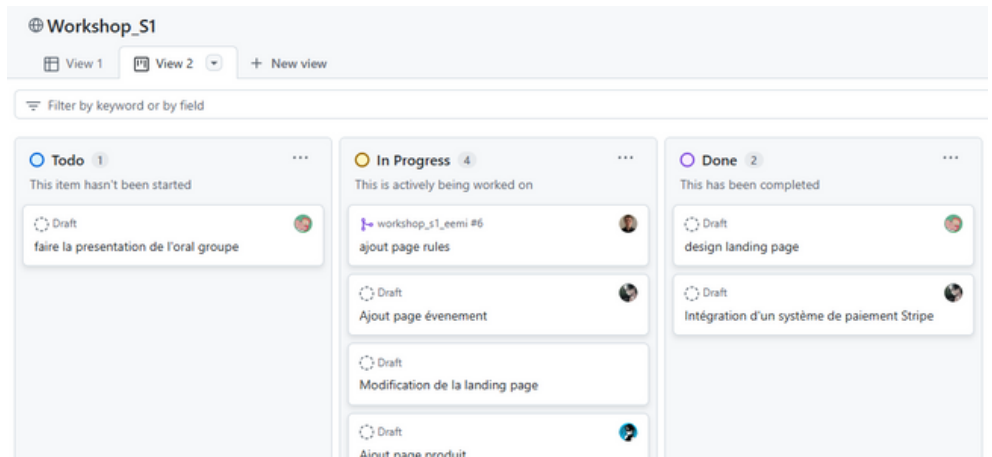
# MCD/MLD



## GESTION DE PROJET ET ORGANISATION

### Méthodologie Kanban

Visualisation du travail en colonnes (To Do, In Progress, Done).



**VISUALISATION DU FLUX DE TRAVAIL**

**FLEXIBILITÉ ET ADAPTATION CONTINUE**

**RÉDUCTION DES TEMPS D'ATTENTE ET AMÉLIORATION DE LA PRODUCTIVITÉ**