TITRE PROFESSIONNEL DÉVELOPPEUR WEB ET WEB MOBILE

Soutenance

Kanthio DOUCOURE – JOIE-EPICE

Paris -09/02/2025

Qui suis-je ?

Parcours

-Baccalauréat en Science

- 2019-2020

-1ère année universitaire en ISI (ingénierie Des systèmes informatique)

- 2021-2022

- Centre Européen de Formation

- 2023-2025
- Spécialisation en back-end et front-end

Pourquoi ?

- Acquérir une base scientifique solide
- Développer mes compétences en informatique
- Me spécialiser en développement

SOMMAIRE

PRÉSENTATION DU PROJET

Joie Épice est une plateforme e-commerce spécialisée dans la vente d'épices de qualité. L'objectif du projet est de proposer une expérience fluide et intuitive aux utilisateurs, leur permettant de parcourir les produits, d'effectuer des achats en ligne.

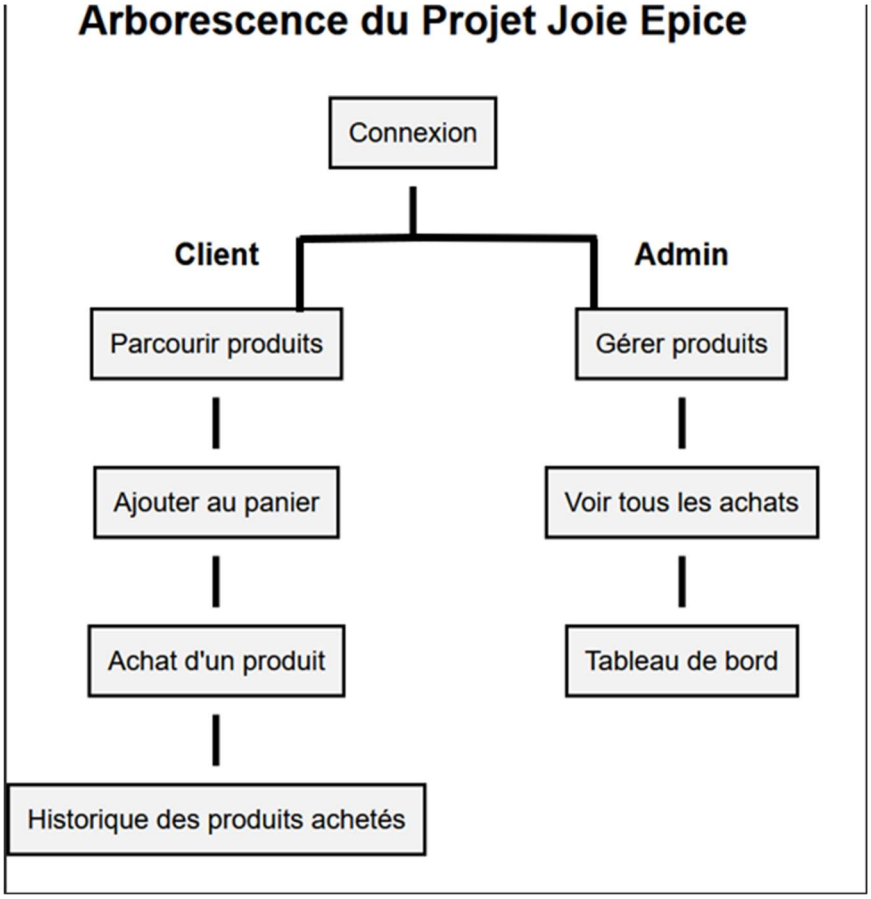
Ce projet a été conçu et développé en **solo**, en mettant l'accent sur une architecture robuste et une interface utilisateur moderne.

CONCEPTION

- **2 types d'utilisateurs :**
  - ➢ *Clients*
  - ➢ *Administrateurs*

En tant que	Je souhaite	Afin de
Client		
Client	Parcourir produits	Découvrir et sélectionner des produits
Client	Ajouter au panier	Sélectionner et sauvegarder les produits pour l'achat
Client	Achat d'un produit	Réaliser un achat en ligne facilement
Client	Historique des produits achetés	Consulter mes achats passés
Admin		
Admin	Gérer produits	Mettre à jour le catalogue et gérer les produits
Admin	Voir tous les achats	Surveiller les transactions et analyser les ventes
Admin	Tableau de bord	Ajouter ou modifier des produits

Arborescence



MCD

Modèle Conceptuel de Données (MCD)

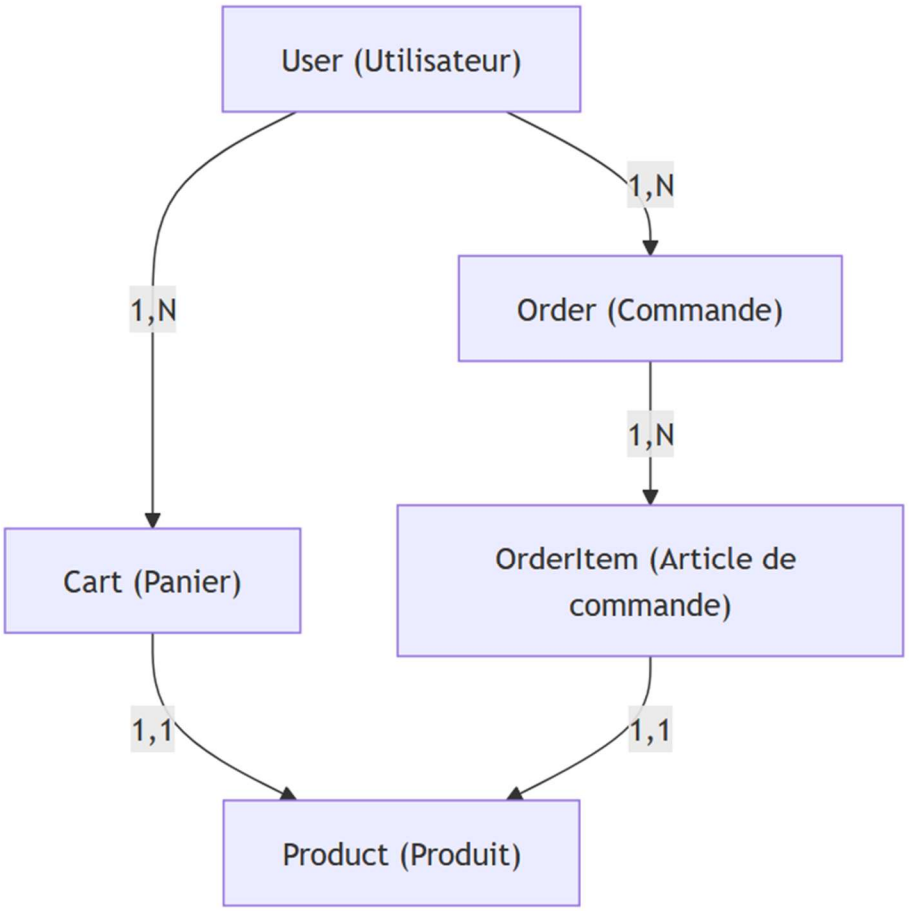


Table Utilisateur

Attribut	Type	Clé
id	int	Clé primaire
nom_utilisateur	varchar(255)	Unique
email	varchar(255)	Unique
mot_de_passe	varchar(255)	
date_creation	datetime	
date_modification	datetime	

Table Profil

Attribut	Type	Clé
id	int	Clé primaire
utilisateur_id	int	Clé étrangère
prenom	varchar(255)	
nom	varchar(255)	
date_naissance	date	
bio	text	

Table Commande

Attribut	Type	Clé
id	int	Clé primaire
utilisateur_id	int	Clé étrangère
date_commande	datetime	
montant_total	decimal(10,2)	
statut	varchar(50)	

MLD

User (id, username, email, roles, password, is_verified)

Product (id, name, price, image, description, stock)

Cart (id, user_id, product_id, quantity)

Order (id, user_id, order_date)

OrderItem (id, order_id, product_id, quantity)

DÉVELOPPEMENT

Création de la base de données

1. Entité User :

- **Rôle** : L'entité User représente un utilisateur du système. Un utilisateur peut avoir plusieurs paniers (Cart) et plusieurs commandes (Order).
- **Attributs** :
    - id : Identifiant unique.
    - username, email : Informations d'identification de l'utilisateur.
    - roles : Attribut définissant les rôles de l'utilisateur (ex : ROLE_CLIENT).
    - password : Mot de passe de l'utilisateur.
    - isVerified : Indicateur si l'utilisateur a vérifié son compte.
- **Relations** :
    - **Carts** : Un utilisateur peut avoir plusieurs paniers, chacun étant lié à un ou plusieurs produits.
    - **Orders** : Un utilisateur peut passer plusieurs commandes, chaque commande étant liée à un ou plusieurs produits via des OrderItem.

2. Entité Product :

- **Rôle** : L'entité Product représente un produit dans le magasin.
- **Attributs** :
    - id : Identifiant unique.
    - name : Nom du produit.
    - price : Prix du produit.
    - image : Image du produit.
    - description : Description du produit.
    - stock : Quantité de stock disponible pour ce produit.
- **Relations** :
    - **Carts** : Un produit peut être ajouté à plusieurs paniers différents (via Cart).
    - **OrderItems** : Un produit peut être inclus dans plusieurs éléments de commande (via OrderItem).

3. Entité Cart :

- **Rôle** : L'entité Cart représente un panier d'achat où l'utilisateur peut ajouter des produits.

- **Attributs** :
    - id : Identifiant unique du panier.
    - user : L'utilisateur auquel ce panier appartient.
    - product : Le produit ajouté au panier.
    - quantity : La quantité de produit dans le panier.
- **Relations** :
    - **User** : Un panier est associé à un utilisateur unique.
    - **Product** : Un panier est associé à un produit spécifique.
- **Rôle dans le processus** : Lorsqu'un utilisateur ajoute des produits au panier, cela permet de suivre la quantité des produits avant la commande. Lors de la validation de la commande, les produits du panier sont déplacés vers la commande.

4. Entité Order :

- **Rôle** : L'entité Order représente une commande passée par un utilisateur, incluant plusieurs articles de commande.
- **Attributs** :
    - id : Identifiant unique de la commande.
    - user : L'utilisateur qui a passé la commande.
    - orderDate : La date à laquelle la commande a été passée.
    - orderItems : Les articles de la commande (représentés par l'entité OrderItem).
- **Relations** :
    - **User** : Une commande appartient à un utilisateur.
    - **OrderItems** : Une commande contient plusieurs articles de commande (via OrderItem).
- **Rôle dans le processus** : Lorsqu'un utilisateur finalise ses achats, un panier est converti en commande. La commande est associée à plusieurs articles de commande qui correspondent aux produits achetés et à leurs quantités.

5. Entité OrderItem :

- **Rôle** : L'entité OrderItem représente un article dans une commande, c'est-à-dire un produit spécifique avec une quantité associée dans une commande.
- **Attributs** :
    - id : Identifiant unique de l'article de commande.
    - order : La commande à laquelle cet article appartient.
    - product : Le produit spécifique ajouté à la commande.

    - quantity : La quantité de produit achetée.
- **Relations** :
    - **Order** : Un article de commande appartient à une commande spécifique.
    - **Product** : Un article de commande fait référence à un produit acheté.
- **Rôle dans le processus** : Lors de la finalisation de la commande, chaque produit ajouté au panier devient un OrderItem. Les OrderItems associent les produits aux commandes avec les quantités spécifiques de chaque produit.

Interaction entre les entités :

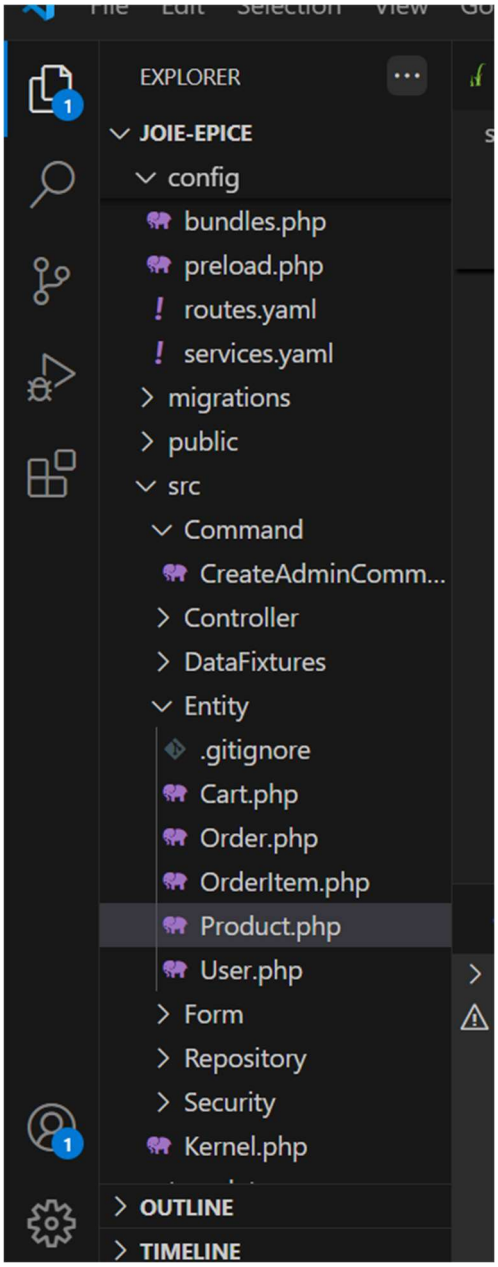
- Lorsqu'un utilisateur ajoute des produits au panier (Cart), chaque produit est associé à un panier spécifique avec une quantité donnée.
- Une fois que l'utilisateur passe une commande, le panier est transféré vers la commande, où chaque produit devient un OrderItem dans l'entité Order.
- Les produits sont associés aux paniers et aux commandes via des relations ManyToOne pour assurer une gestion cohérente des stocks et des commandes.
- Le stock d'un produit est mis à jour lorsque la commande est confirmée.

Schéma relationnel :

- **User** a une relation **OneToMany** avec **Cart** et **Order**.
- **Product** a une relation **OneToMany** avec **Cart** et **OrderItem**.
- **Cart** a une relation **ManyToOne** avec **User** et **Product**.
- **Order** a une relation **ManyToOne** avec **User** et une relation **OneToMany** avec **OrderItem**.
- **OrderItem** a une relation **ManyToOne** avec **Order** et **Product.**

Réalisation du front-end

partie « dynamique »



1. RegistrationController

- Fonctionnalité : Gère l'inscription des utilisateurs.
- Méthodes :
    - register : Crée un formulaire d'inscription. Si le formulaire est soumis et valide, le mot de passe de l'utilisateur est haché, un rôle client est assigné, l'utilisateur est enregistré dans la base de données, un email de confirmation est envoyé, puis l'utilisateur est redirigé vers la page d'accueil.
    - verifyUserEmail : Vérifie le lien de confirmation de l'email. Si le lien est valide, l'utilisateur est marqué comme vérifié ; sinon, un message d'erreur est affiché. L'utilisateur est ensuite redirigé vers la page d'inscription.

2. SecurityController

- Fonctionnalité : Gère l'authentification des utilisateurs.
- Méthodes :
    - login : Affiche la page de connexion et gère les erreurs potentielles.
    - logout : Gère la déconnexion des utilisateurs. Cette méthode est généralement interceptée par le pare-feu de sécurité.

3. ProductController

- Propriétés : Utilise EntityManagerInterface pour interagir avec la base de données.
- Méthodes :

- productDetails($id) : Récupère un produit par son identifiant. Si le produit est trouvé, la vue correspondante est rendue ; sinon, une exception est lancée.

4. AuthController

- Propriétés : Utilise EntityManagerInterface pour les opérations liées aux utilisateurs.
- Méthodes :
  - forgotPassword(Request $request) : Gère les demandes de réinitialisation de mot de passe. Si l'email existe, redirige vers une page de confirmation ; sinon, affiche un message d'erreur.
  - confirmResetPassword(Request $request, string $email, UserPasswordHasherInterface $userPasswordHasher) : Permet à l'utilisateur de définir un nouveau mot de passe après vérification de l'email.

5. CartController

- Propriétés : Utilise EntityManagerInterface pour interagir avec la base de données.
- Méthodes :
  - addToCart(int $productId, ProductRepository $productRepository) : Ajoute un produit au panier après vérification du stock.
  - viewCart(Request $request) : Affiche le contenu du panier de l'utilisateur.
  - removeFromCart(int $id) : Supprime un article du panier et restaure le stock correspondant.
  - increaseQuantity(int $id) : Augmente la quantité d'un article dans le panier après vérification du stock.
  - decreaseQuantity(int $id) : Diminue la quantité d'un article ou le supprime si la quantité atteint 1.

6. StripeController

- Propriétés : Utilise EntityManagerInterface pour gérer les entités.
- Méthodes :
  - index(Request $request) : Affiche la page de paiement Stripe avec le total de la session.
  - createCharge(Request $request) : Traite le paiement via Stripe. Si le paiement est réussi, le panier est vidé, une nouvelle commande est créée et enregistrée, puis un message de succès est affiché.

7. OrderController

- Propriétés : Utilise EntityManagerInterface pour interagir avec la base de données.
- Méthodes :

- history() : Affiche l'historique des commandes de l'utilisateur. Si l'utilisateur est administrateur, toutes les commandes sont affichées ; sinon, seules les commandes de l'utilisateur connecté sont montrées.

EXEMPLE DE RECHERCHE

1. **Sources consultées** :
    - **YouTube** : Visionnage de tutoriels sur la conception de maquettes de sites e-commerce.
    - **Google** : Recherche d'articles et d'exemples de designs de sites e-commerce.
    - **Sites e-commerce** : Analyse de designs et fonctionnalités de sites existants.
2. **Informations clés** :
    - **YouTube** : Apprentissage des meilleures pratiques en design de maquettes.
    - **Google** : Identification des éléments essentiels pour une expérience utilisateur optimale.
    - **Sites e-commerce** : Observation des tendances actuelles en matière de design et de navigation.
3. **Réflexions personnelles** :
    - Les tutoriels vidéo ont facilité la compréhension des concepts de design.
    - Les recherches en ligne ont aidé à identifier les éléments clés d'un site e-commerce réussi.
    - L'analyse de sites existants a permis de repérer des pratiques exemplaires à adopter.

CONCLUSION

- Satisfaction du travail accompli
- Autonomie
- Expérience enrichissante
- Me motiver à approfondir et à développer davantage le projet à l'avenir

A vos questions !