

Rapport de projet

Site e-commerce

Délices Sucrés



ATHMANE LINA

Etudiante en 3eme année Software
engineering
Année scolaire : 2024-2025

Ce rapport présente la conception et la réalisation d'un site e-commerce dédié à la vente de pâtisseries artisanales, nommé "Délices Sucrés". Le projet intègre à la fois une interface web conviviale et une base de données robuste pour gérer l'ensemble des fonctionnalités nécessaires à une boutique en ligne moderne.

1. Conception générale:

1.1 Architecture du site:

Le site "Délices Sucrés" est conçu selon une architecture modulaire avec séparation des préoccupations :

- **La couche présentation (frontend):** fichiers PHP avec HTML/CSS/JavaScript
- **La couche logique métier :** fonctions PHP et appels aux procédures stockées
- **La couche données :** base de données MySQL avec tables, procédures stockées et triggers

L'organisation des fichiers reflète cette architecture :

- **config :** contient les fichiers de configuration comme database.php
- **includes:** contient les éléments réutilisables (header, footer, fonctions)
- **admin:** interface d'administration pour la gestion des produits et commandes
- **photo:** stockage des images des produits
- **Autres** fichiers PHP pour les différentes pages du site



1.2 Fonctionnalités principales:

Le site offre les fonctionnalités suivantes :

- Affichage des produits par catégories et collections
- Recherche et filtrage des produits selon plusieurs critères
- Gestion du panier d'achat
- Système d'authentification (connexion/inscription)
- Processus de commande complet
- Gestion des avis clients
- gestion de contact clients.
- Interface d'administration
- déconnexion du client

2. Base de données:

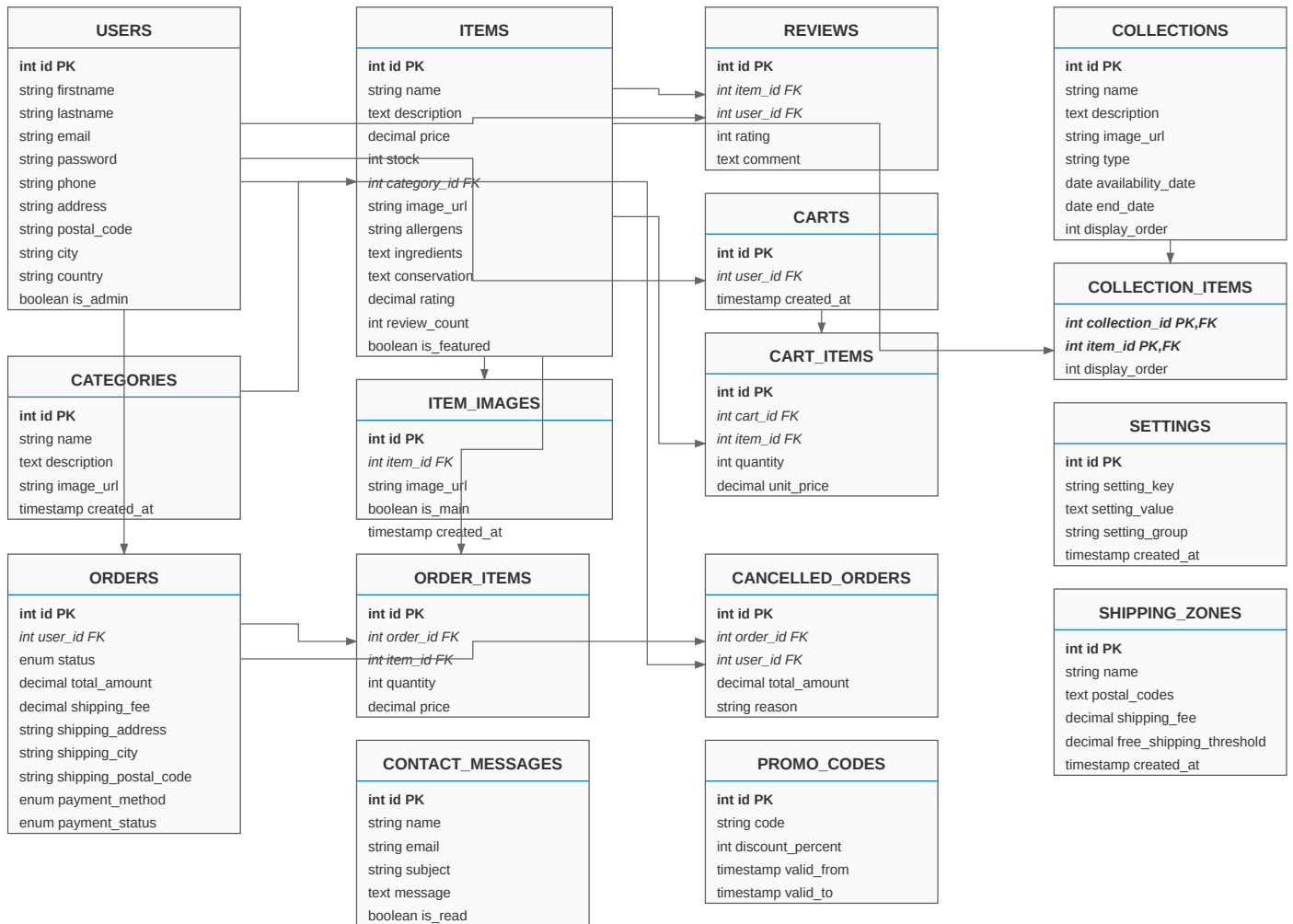
2.1 Schéma relationnel:

La base de données "ecommerce_db" est structurée autour de 16 tables principales :

Tables principales et leurs relations :

- **users** : Stocke les informations des utilisateurs
- **items** : Contient les produits disponibles à la vente
- **categories** : Classifie les produits par catégories
- **collections** : Regroupe les produits dans des collections thématiques
- **collection_items** : Table de liaison entre collections et produits
- **carts** : Panier d'achat des utilisateurs
- **cart_items** : Produits ajoutés au panier
- **orders** : Commandes passées par les utilisateurs
- **order_items** : Produits inclus dans les commandes
- **reviews** : Avis clients sur les produits
- **cancelled_orders** : Historique des commandes annulées
- **promo_codes** : Codes promotionnels
- **settings** : Paramètres généraux du site
- **shipping_zones** : Zones de livraison avec tarifs
- **contact_messages**: permet aux clients de nous contacter
- **item_image**: permet de gérer plusieurs photos pour chaque produit, facilitant la création de galeries d'images.

Schema relationnel de la base de données



2.2 Procédures stockées

Quatre procédures stockées ont été implémentées pour gérer les fonctionnalités clés du site :

2.2.1 Procédure FinalizeOrder:

Cette procédure permet de transformer un panier en commande validée :

```
CREATE PROCEDURE FinalizeOrder(
    IN p_user_id INT,
    IN p_shipping_address VARCHAR(255),
    IN p_shipping_city VARCHAR(50),
    IN p_shipping_postal_code VARCHAR(10),
    IN p_shipping_country VARCHAR(50),
    IN p_payment_method VARCHAR(20),
    IN p_notes TEXT,
    OUT p_order_id INT
)
```

Fonctionnement :

1. Vérification de l'existence du panier et de son contenu
2. Calcul du montant total incluant les frais de livraison
3. Vérification de la disponibilité du stock
4. Création de la commande
5. Transfert des articles du panier vers la commande
6. Vidage du panier

Cette procédure est utilisée dans le fichier paiement.php lorsque l'utilisateur finalise sa commande.

```
// Utiliser la procédure stockée pour finaliser la commande
$stmt = $conn->prepare("CALL FinalizeOrder(?, ?, ?, ?, ?, ?, ?, @p_order_id)");
$stmt->bindParam(1, $_SESSION['user_id'], PDO::PARAM_INT);
$stmt->bindParam(2, $address, PDO::PARAM_STR);
$stmt->bindParam(3, $city, PDO::PARAM_STR);
$stmt->bindParam(4, $postalCode, PDO::PARAM_STR);
$stmt->bindParam(5, $country, PDO::PARAM_STR);
$stmt->bindParam(6, $paymentMethod, PDO::PARAM_STR);
$stmt->bindParam(7, $notes, PDO::PARAM_STR);
$stmt->execute();

// Récupérer l'ID de la commande
$stmt = $conn->query("SELECT @p_order_id as order_id");
$result = $stmt->fetch();
$order_id = $result['order_id'];
```

2.2.2 Procédure GetOrderDetails:

Cette procédure affiche les détails d'une commande spécifique pour un client :

```
CREATE PROCEDURE GetOrderDetails(
    IN p_order_id INT,
    IN p_user_id INT
)
```

```
// Utiliser la procédure stockée pour récupérer les détails de la commande
$stmt = $conn->prepare("CALL GetOrderDetails(?, ?)");
$stmt->bindParam(1, $orderId, PDO::PARAM_INT);
$stmt->bindParam(2, $_SESSION['user_id'], PDO::PARAM_INT);
$stmt->execute();

// Récupérer les informations générales de la commande
$orderInfo = $stmt->fetch();

// Vérifier si la commande existe et appartient à l'utilisateur
if (!$orderInfo) {
    header("Location: mes-commandes.php");
    exit;
}

// Passer au jeu de résultats suivant pour récupérer les détails des produits
$stmt->nextRowset();
$orderItems = $stmt->fetchAll();

// Passer au jeu de résultats suivant pour récupérer les totaux
$stmt->nextRowset();
$orderTotals = $stmt->fetch();
```

Fonctionnement :

1. Vérification que la commande appartient bien à l'utilisateur
2. Affichage des informations générales de la commande
3. Détails des produits commandés
4. Calcul du total à payer

Cette procédure est utilisée dans le fichier commande-details.php pour afficher le récapitulatif d'une commande.

2.2.3 Procédure GetOrderHistory:

Cette procédure récupère l'historique des commandes d'un client :

```
CREATE PROCEDURE GetOrderHistory(
    IN p_user_id INT
)
```

Fonctionnement :

- Affiche toutes les commandes passées par l'utilisateur
- Inclut le statut, le montant, la date et le nombre d'articles

Cette procédure est utilisée dans le fichier mes-commandes.php pour afficher l'historique des commandes.

```
// Récupérer l'historique des commandes
try {
    $conn = connectDB();

    // Utiliser la procédure stockée pour récupérer l'historique des commandes
    $stmt = $conn->prepare("CALL GetOrderHistory(?)");
    $stmt->bindParam(1, $_SESSION['user_id'], PDO::PARAM_INT);
    $stmt->execute();

    $orders = $stmt->fetchAll();
} catch(PDOException $e) {
    die("Erreur lors de la récupération de l'historique des commandes: " . $e->getMessage());
}
?>
```

2.2.4 Procédure update order status

Cette procédure permet à l'administrateur de mettre à jour le statut d'une commande :

```
CREATE PROCEDURE update_order_status(
    IN p_order_id INT,
    IN p_status VARCHAR(20),
    IN p_admin_id INT
)
```

Fonctionnement :

1. Vérification de l'existence de la commande
2. Validation du nouveau statut
3. Mise à jour du statut de la commande

Cette procédure est utilisée dans l'interface d'administration pour gérer les commandes.

2.3 Triggers

Trois triggers ont été implémentés pour maintenir l'intégrité des données :

2.3.1 Trigger check_stock:

Ce trigger vérifie que la quantité commandée ne dépasse pas le stock disponible :

```
CREATE TRIGGER check_stock  
BEFORE INSERT ON order_items  
FOR EACH ROW
```

Ce contrôle prévient les commandes excédant le stock et garantit une expérience client sans frustration.

2.3.2 Trigger order_cancelled_handler:

Ce trigger gère les commandes annulées :

```
CREATE TRIGGER order_cancelled_handler  
AFTER UPDATE ON orders  
FOR EACH ROW
```

Fonctionnement :

1. Détection du changement de statut vers "cancelled"
2. Restauration du stock pour chaque article de la commande
3. Enregistrement de l'annulation dans la table d'historique

2.3.3 Trigger update_stock:

Ce trigger met à jour automatiquement le stock après une commande :

```
CREATE TRIGGER update_stock  
AFTER INSERT ON order_items  
FOR EACH ROW
```

Il assure que le stock est toujours à jour après chaque commande validée.

Notre site "Délices Sucrés" intègre toutes les fonctionnalités essentielles d'un e-commerce moderne, avec une attention particulière à l'expérience utilisateur et à l'intégration avec la base de données.

3.1 Fonctionnalités principales:

Vitrine et navigation :

- La page d'accueil met en avant les collections saisonnières et produits phares
- La boutique présente tous les produits avec filtres dynamiques
- Chaque produit possède une page détaillée avec description, prix et avis

Gestion utilisateur :

- Système d'authentification complet (inscription/connexion)
- possibilités de se déconnecter
- Historique des commandes accessible via la procédure **GetOrderHistory**

Expérience d'achat :

- Panier persistant entre les sessions utilisateur
- Processus de commande en 3 étapes (panier, livraison, paiement)
- Intégration de la procédure **FinalizeOrder** lors de la validation finale

Interactions utilisateurs :

- Système d'avis et de notation des produits
- Formulaire de contact pour le service client
- Le système de recherche permet aux utilisateurs de filtrer les produits selon plusieurs critères comme les catégories, le prix et les allergènes.

Système de sessions et cookies :

La gestion du panier et de l'authentification utilisateur est implémentée grâce aux sessions PHP .

TECHNOLOGIES UTILISÉES :

1. HTML:

Structure les pages en définissant les éléments de base (formulaires, boutons, sections). Sert à concevoir l'interface utilisateur pour les fonctionnalités comme l'inscription, la connexion et la gestion du panier.

2. CSS :

Apporte du style aux pages web : mise en page, couleurs, typographie. Améliore l'apparence et la lisibilité de l'interface utilisateur.

3. JavaScript :

Dynamise les pages web avec des interactions en temps réel sans rechargement de page. Valide les formulaires côté client et gère les interactions dynamiques, comme la gestion du panier.

4. PHP:

Traite les requêtes côté serveur (inscription, connexion, gestion du panier). Connecte et interagit avec la base de données MySQL.

5. MySQL:

Stocke les données essentielles (clients, produits, paniers).

6. WampServer :

Fournit un environnement de développement local permettant d'exécuter le serveur web Apache, PHP et MySQL

CONCLUSION :

En résumé, ce projet e-commerce propose une interface conviviale, et un responsive design qui s'adapte aux différents appareils (mobile, tablette, desktop). Facilitant la navigation des utilisateurs, la gestion de leur panier, de donner leurs avis et de nous contacter. Du côté des administrateurs, les fonctionnalités permettent une gestion simple et efficace des produits, catégories, utilisateurs et commandes. Grâce à l'utilisation de technologies modernes et une structure bien pensée, l'application garantit une expérience fluide et agréable pour tous.