

Documentation Technique – L'Univers des Jeux

1. Introduction

Brève présentation du projet :

- Nom du projet : **L'Univers des Jeux**
- Type : Site e-commerce de vente de jeux vidéo
- Technologies utilisées : HTML, CSS, JavaScript, PHP, MySQL
- Objectif : Permettre aux utilisateurs d'acheter, noter et donner leur avis sur des jeux vidéo récemment sortis.

2. Installation et Configuration

2.1 Prérequis

Avant d'installer et d'exécuter le projet, assurez-vous d'avoir :

- Un serveur local (XAMPP, WAMP ou autre)
- PHP 7+
- MySQL
- Visual Studio Code ou tout autre éditeur de code
- Composer (si vous utilisez des bibliothèques PHP externes)

2.2 Installation du projet

Cloner le projet :

git clone <https://github.com/Marion101200/grandprojet>

Configurer la base de données :

- Importer le fichier `db` dans MySQL

3. Lancer le serveur local :

- Si vous utilisez XAMPP : démarrer Apache et MySQL
- Accéder au projet via
<http://localhost/grandprojet-main/grandprojet/accueil.php>

4. Mode d'emploi pour l'Admin

L'administrateur a accès à une interface dédiée pour gérer le contenu du site :

- **Gestion des jeux** : Ajouter, modifier, supprimer des jeux depuis le back-office.
- **Gestion des commandes** : Suivi des commandes, validation et modification des statuts.
- **Gestion des utilisateurs** : Bloquer ou débloquer des comptes utilisateurs.
- **Modération des avis** : Suppression des avis inappropriés ou non conformes.
- **Gestion des catégories** : Ajout et modification des catégories de jeux.

5. Gestion des Emails avec PHP Mailer

L'envoi d'emails (confirmation d'inscription, validation de commande, etc.) est géré avec **PHP Mailer**.

Installation de PHP Mailer

composer require phpmailer/phpmailer

Envoi d'un email de confirmation d'inscription

Fichier : email.php

```
use PHPMailer\PHPMailer\PHPMailer;
```

```
use PHPMailer\PHPMailer\Exception;
```

```
require 'vendor/autoload.php';
```

```
$mail = new PHPMailer(true);
```

```
try {
```

```
    $mail->isSMTP();
```

```
    $mail->Host = 'smtp.example.com';
```

```
    $mail->SMTPAuth = true;
```

```
    $mail->Username = 'votre_email@example.com';
```

```
$mail->Password = 'mot_de_passe';

$mail->SMTPSecure = 'tls';

$mail->Port = 587;

$mail->setFrom('no-reply@universjeux.com', 'L'Univers des Jeux');

$mail->addAddress($userEmail);

$mail->Subject = 'Confirmation d'inscription';

$mail->Body = 'Merci de vous être inscrit sur notre site !';

$mail->send();

} catch (Exception $e) {

    echo "Erreur : {$mail->ErrorInfo}";

}
```

6. Sécurité et Bonnes Pratiques

- **Sécurisation des entrées utilisateur** (utilisation de `PDO::prepare` pour éviter les injections SQL).
 - **Hachage des mots de passe** avec `password_hash()` lors de l'inscription.
 - **Validation côté client** avec JavaScript pour améliorer l'expérience utilisateur.
 - **Utilisation des sessions** pour gérer l'authentification.
-

7. API de Paiement (Stripe, PayPal, etc.)

Fichier : `paiement.php`

- Expliquer l'intégration avec Stripe/PayPal
- Fournir un exemple de script de paiement sécurisé

8. Gestion des Erreurs et Débogage

- **Logs des erreurs** : Utilisation de `error_log` ou activation du mode `PDO::ERRMODE_EXCEPTION`.
 - **Outils de débogage** : Utilisation de `var_dump()`, `print_r()`, et Xdebug.
-

9. FAQ et Ressources

- Liens vers les documentations officielles (PHP, MySQL, Stripe, etc.).
- Solutions aux erreurs courantes (problèmes de connexion, erreurs SQL, etc.).