Cahier des charges

BLOG DE JEUX VIDEOS

Par LAIDOUNI ZAKARIA

Table des matières

1. Introduction

Je vous présente mon blog de jeux vidéo dédié à la publication d'articles sur les jeux préférés de son auteur et ainsi faire peut être découvrir à d'autres utilisateurs de nouveaux jeux.
Le blog propose un espace commentaires pour s'exprimer sur le jeu en question, des analyses, des astuces et des discussions approfondies sur divers jeux, offrant ainsi un espace de partage pour les passionnés de jeux vidéo qui pourrons même découvrir d'autre jeux auxquels on aurait jamais penser y jouer.

2. Résumé du projet

Le projet vise à créer un blog de jeux vidéos convivial et sécurisé. Ses objectifs principaux sont :

- Publication d'articles sur les jeux vidéo, incluant un espace commentaires.

- Interface d'administration sécurisée pour gérer le contenu.

- Fiabilité et sécurité avec des mesures de protection des données et d'authentification robustes.

Côté Front-End, les technologies utilisées incluent HTML5, CSS3 et Bootstrap pour assurer une interface utilisateur attrayante et responsive.

Côté Back-End, le projet repose sur PHP pour la logique serveur, MySQL pour la base de données et PDO pour l'accès aux données garantissant ainsi une gestion efficace des utilisateurs, des articles et des commentaires.

3/ Cahier des charges

3.1. Conceptualisation du Projet

Le projet consiste à concevoir et développer un blog de jeux vidéo offrant une plateforme pour la publication d'articles divers sur les jeux vidéo. Les principaux objectifs sont les suivants :

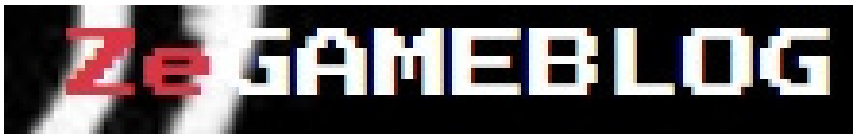
- **Objectifs du Projet :** Créer une plateforme interactive proposant des articles sur nos jeux vidéo préférés et ainsi pouvoir y laisser un commentaire si nous le souhaitons.

- **Cibles :** Le blog cible un large public, des joueurs occasionnels aux professionnels de l'industrie, en offrant un contenu informatif et divertissant.

- **Contenus** : Le blog abritera une diversité d'articles sur les jeux vidéo, accompagnés d'éléments visuels tels que des images du cover du jeux.

- **Conception Graphique :** L'objectif est de concevoir une interface attrayante avec un logo distinctif, une typographie claire et une palette de couleurs dans le thème du gaming pour que l'utilisateur se sente dans l'univers que le site veut proposer. Comme couleur pour le logo j'ai utilisé red rgb(255.0.0) et une Font qui s'appelle font-family: 'Press Start 2P', sans-serif; que j'ai récupéré en utilisant GoogleFont .

  Voici un screen du LOGO :

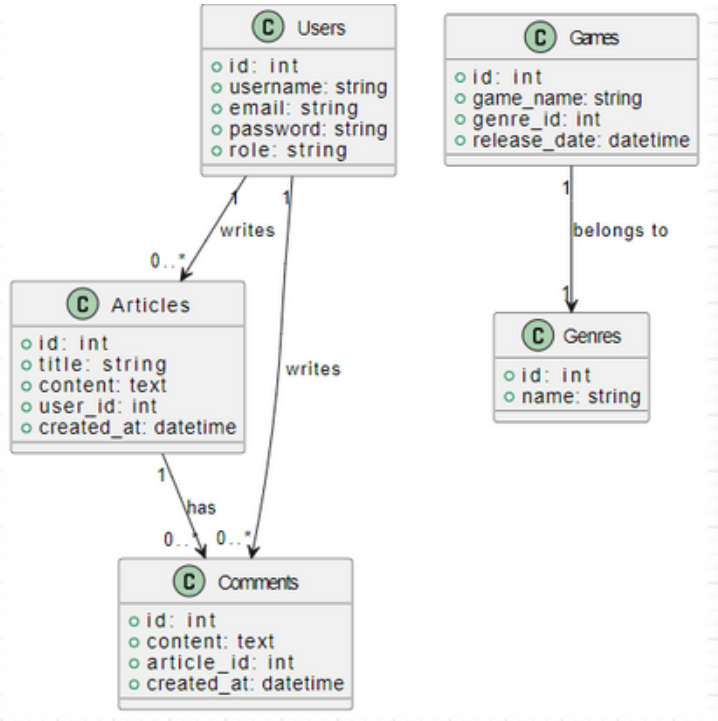


3.2. Utilisations et Rôles

- **Analyse et Organisation des Besoins :**

Une analyse des besoins des utilisateurs a été effectuée afin de concevoir une base de données adaptée avec une représentation visuelle à l'aide de schémas explicatifs.

Description des Classes et Relations :



Relations

- Un utilisateur (Users) peut avoir plusieurs articles (Articles) mais un article n'appartient qu'à un seul utilisateur (relation un-à-plusieurs ou One to many).

- Un utilisateur (Users) peut avoir plusieurs commentaires (Comments) mais un commentaire n'appartient qu'à un seul utilisateur (relation un-à-plusieurs ou One to many).

- Un article (Articles) peut avoir plusieurs commentaires (Comments) mais un commentaire n'appartient qu'à un seul article (relation un-à-plusieurs ou One to many).

- Un genre (Genres) peut avoir plusieurs jeux (Games) mais un jeu n'appartient qu'à un seul genre (relation un-à-plusieurs ou One to many).

Explication des Relations

1.Users to Articles :

Chaque utilisateur peut écrire plusieurs articles. Cette relation est représentée par une ligne reliant Users à Articles avec une multiplicité de 1 du côté de Users et 0..* du côté de Articles.

2.Users to Comments :

Chaque utilisateur peut écrire plusieurs commentaires. Cette relation est représentée par une ligne reliant Users à Comments avec une multiplicité de 1 du côté de Users et 0..* du côté de Comments.

3.Articles to Comments :

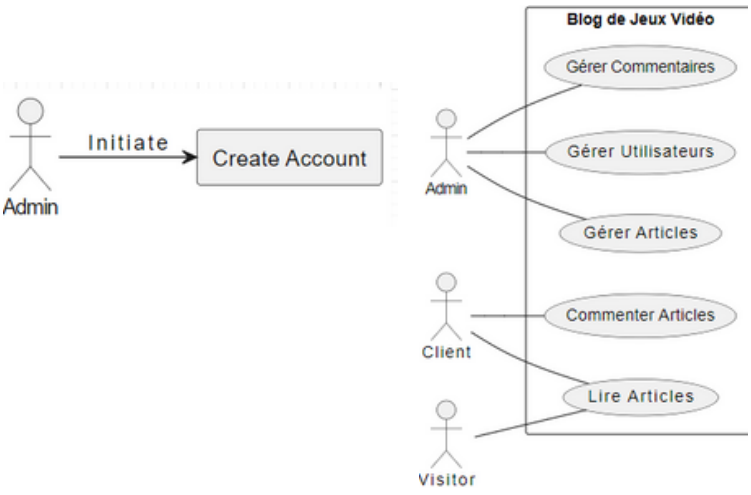
Chaque article peut avoir plusieurs commentaires. Cette relation est représentée par une ligne reliant Articles à Comments avec une multiplicité de 1 du côté de Articles et 0..* du côté de Comments.

4.Games to Genres :

Chaque jeu appartient à un genre spécifique, mais un genre peut avoir plusieurs jeux. Cette relation est représentée par une ligne reliant Games à Genres avec une multiplicité de 1 du côté de Genres et 0..* du côté de Games.

- **<u>Schémas du Processus Utilisateur :</u>**

Des schémas seront élaborés pour décrire le processus suivi par les différents utilisateurs (administrateurs, visiteurs, etc.) afin de clarifier leurs rôles et responsabilités.



3.3. Mise en place du projet

Listes des pages :

- **Page d'Accueil :**
  - Affiche une sélection d'articles récents et populaires.
  - Permet aux utilisateurs de naviguer vers différentes catégories de jeux.

- **Page d'Article Individuel :**
  - Affiche un article spécifique avec son contenu détaillé.
  - Permet aux utilisateurs de lire l'article et les commentaires associés.
  - Autorise les utilisateurs authentifiés à commenter l'article.

- **Page de Recherche :**
    - Permet aux utilisateurs de rechercher des articles en fonction de mots-clés ou de catégories de jeux.
    - Affiche les résultats de recherche pertinents.
- **Page de Connexion :**
    - Permet aux utilisateurs de se connecter à leur compte utilisateur.
    - Inclut des champs pour saisir le nom d'utilisateur et le mot de passe.
- **Page d'Inscription :**
    - Permet aux nouveaux utilisateurs de créer un compte sur le blog.
    - Inclut des champs pour saisir le nom d'utilisateur, l'adresse e-mail et le mot de passe.
- **Page de Profil Utilisateur :**
    - Affiche les informations et les paramètres du compte utilisateur.
    - Permet aux utilisateurs de modifier leur nom d'utilisateur leur adresse e-mail et leur mot de passe.

- **Page d'Administration :**
    - Réservée aux administrateurs du blog.
    - Permet de gérer les articles, les commentaires et les utilisateurs.
    - Autorise l'ajout, la modification et la suppression d'articles, de commentaires et d'utilisateurs.

- **Page de Gestion des Articles :**
    - Permet à l'administrateur de gérer les articles publiés.
    - Affiche une liste d'articles avec des options pour les éditer, les supprimer ou en créer de nouveaux.

- **Page de Gestion des Articles :**
    - Permet à l'administrateur de gérer les articles publiés.
    - Affiche une liste d'articles avec des options pour les éditer, les supprimer ou en créer de nouveaux.

- **Page de Gestion des Commentaires :**
    - Permet à l'administrateur de gérer les commentaires associés aux articles.
    - Affiche une liste de commentaires avec des options pour les approuver, les supprimer ou les modifier.

- **Page de Gestion des Utilisateurs :**
    - Permet à l'administrateur de gérer les comptes utilisateurs.
    - Affiche une liste d'utilisateurs avec des options pour modifier leurs rôles, réinitialiser leurs mots de passe ou les supprimer.

4/ Spécifications techniques

4.1/ Technologies du front-end

- **HTML5 et CSS3 :**
    - Intégrés dans toutes les pages de l'interface utilisateur et de l'interface d'administration pour la structure, le style et la mise en page. Le layout qui ce charge de mettre tout le contenu CSS et design de la page .

- **Bootstrap :**

    - Intégré dans les fichiers CSS de l'ensemble de l'application pour rendre l'interface utilisateur réactive et compatible avec différents appareils.
    - Utilisé spécifiquement dans les pages de l'interface utilisateur pour fournir une mise en page réactive et des composants pré-construits pour améliorer l'expérience utilisateur.

4.2/ Technologies du back-end

- **PHP structure MVC :**

    - Utilisé comme langage principal pour la logique métier et le traitement des requêtes côté serveur.

    - **Modèle-Vue-Contrôleur (MVC) :** Nous avons structuré notre application selon le modèle MVC pour séparer les préoccupations et améliorer la maintenabilité du code.

    - **Modèle (Model) :** Les modèles gèrent la logique des données de l'application. Par exemple, les modèles ArticleModel, CommentModel, GameModel, GenreModel, et UserModel sont responsables de l'interaction avec la base de données pour les différentes entités de notre application.

- **Vue (View) :** Les vues sont responsables de l'affichage des données. Elles récupèrent les données des modèles et les présentent à l'utilisateur sous forme de pages HTML.

- **Contrôleur (Controller) :** Les contrôleurs traitent les requêtes entrantes, exécutent la logique d'application nécessaire en utilisant les modèles et renvoient les résultats sous forme de vues. Par exemple, ArticleController et CommentController.

- **MySQL :**

  - Utilisé comme système de gestion de base de données pour stocker les articles, les commentaires, les utilisateurs, etc.

  - Intégré dans les requêtes SQL pour effectuer des opérations de lecture, d'écriture, de mise à jour et de suppression sur les données.

4.3/ Sécurité du site

Cryptage des mots de passe :

- Les mots de passe des utilisateurs sont stockés de manière sécurisée en utilisant l'algorithme de hachage password_hash de PHP avec l'option PASSWORD_DEFAULT. Cet algorithme utilise le bcrypt pour hacher les mots de passe de manière sécurisée.

- Avant d'être stockés dans la base de données, les mots de passe sont cryptés pour éviter toute compromission en cas d'accès non autorisé à la base de données.

```php
// Hasher le mot de passe pour des raisons de sécurité
$hashedPassword = password_hash($password, PASSWORD_DEFAULT);
```

Injections SQL :

Pour protéger contre les injections SQL, nous utilisons des requêtes paramétrées et validons toutes les entrées utilisateur. Nous échappons également toutes les données saisies par les utilisateurs avant de les inclure dans des requêtes SQL. Ces mesures assurent une séparation stricte entre les données et les instructions SQL, empêchant ainsi l'injection de code malveillant.

5/ Compétences du référentiel couvertes par le projet

5.1/ Partie front-end

5.1.1 CP1: Maquetter une application

Pour la conception des maquettes de notre blog de jeux vidéo, j'ai utilisé les outils suivants :

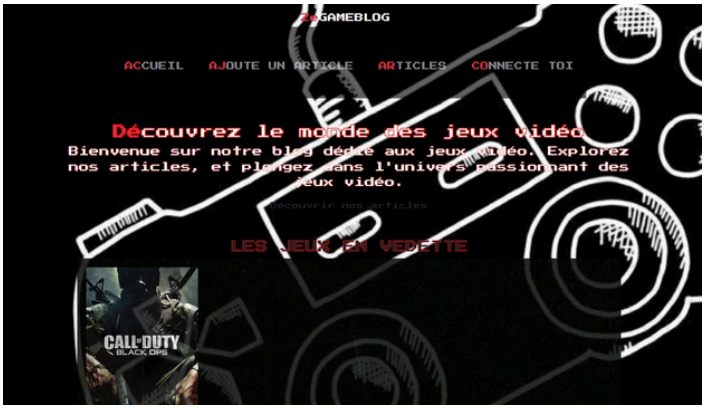
- **HTML & CSS :** Les bases de ma maquette ont été construites en utilisant HTML pour la structure et CSS pour la mise en page et le style. Ces technologies nous ont permis de créer une interface utilisateur propre et structurée.

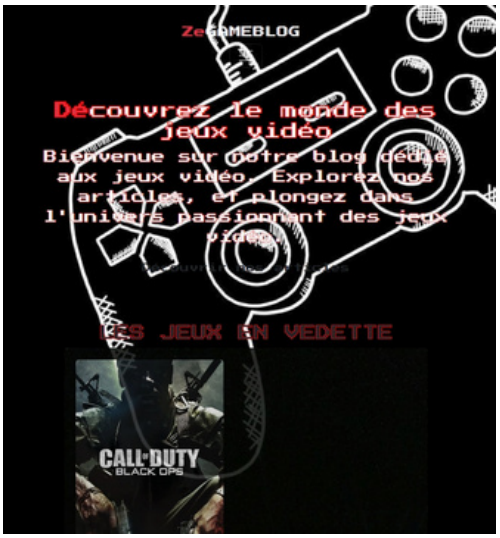
- **Bootstrap :** Pour assurer que notre blog soit réactif et s'adapte bien à différentes tailles d'écran, nous avons utilisé le framework CSS Bootstrap. Cela m'a permis d'intégrer facilement des composants réactifs et d'assurer une mise en page cohérente sur tous les dispositifs.

.

- **Visual Studio Code :** J'ai utilisé l'éditeur de code Visual Studio Code pour écrire et organiser mon code. Cet outil offre une excellente intégration avec les extensions et permet un développement efficace grâce à ses fonctionnalités de débogage et de versionnage de code.

Page d'accueil :





5.2/ Partie back-end

Connexion à la base de données

Pour accéder aux données de notre blog de jeux vidéo, nous avons mis en place un composant d'accès aux données en PHP en utilisant PDO (PHP Data Objects). PDO est une extension qui fournit une interface cohérente pour accéder aux bases de données en PHP, ce qui simplifie et sécurise la gestion des connexions et des requêtes SQL.

Voici comment j'ai procédé pour me connecter à la base de données :

1.Configuration des paramètres de connexion : J'ai créé un fichier de configuration (database.php) pour stocker les paramètres de connexion à la base de données, tels que le nom d'hôte, le nom de la base de données, l'utilisateur et le mot de passe.

```
{
    private $host = "localhost";
    private $db_name = "game_zakaria";
    private $username = "root";
    private $password = "";
    public $pdo;
```

2.Création de la classe Database : J'ai créé une classe Database pour gérer la connexion à la base de données et fournir une méthode pour obtenir une instance de connexion PDO.

```
class Database
{
    private $host = "localhost";
    private $db_name = "game_zakaria";
    private $username = "root";
    private $password = "";
    public $pdo;

    // Méthode pour obtenir la connexion à la base de données
    public function getConnection()
    {
        $this->pdo = null;

        try {
            $this->pdo = new PDO("mysql:host=" . $this->host . ";dbname=" . $this->db_name, $this->username, $this->password);
            $this->pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
            $this->pdo->exec("set names utf8");
        } catch (PDOException $exception) {
            // Afficher l'erreur
            echo "Erreur de connexion : " . $exception->getMessage();
            // Lancer à nouveau l'exception pour permettre au code appelant de gérer l'erreur
            throw $exception;
        }

        return $this->pdo;
    }
```

La conception de ma BDD :

- **Table "articles" :**
  - id (clé primaire)
  - title
  - content
  - user_id
  - created_at

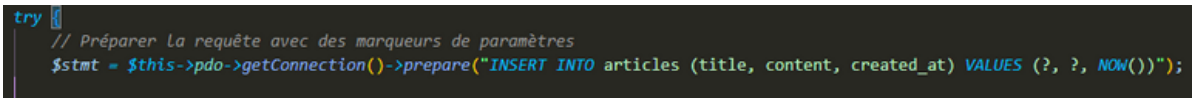
- **Table "comments" :**
  - id (clé primaire)
  - content
  - article_id
  - created_at

- **Table "game" :**
  - id (clé primaire)
  - game_name
  - genre_id
  - release_date

- **Table "genre" :**
  - id (clé primaire)
  - name

- **Table "users" :**
  - id (clé primaire)
  - username
  - email
  - password
  - role (pour définir les rôles des utilisateurs, par exemple, "admin" ou "user")

3.Utilisation de la connexion PDO : Dans les modèles (Models) de mon blog , j'utilise la méthode getConnection() de la classe Database pour obtenir une connexion PDO et exécuter les requêtes SQL.



6/ Réalisation personelles

7/ Jeu d'essai

Etape pour l'inscription :

Pour illustrer le fonctionnement de notre blog de jeux vidéo, nous allons passer en revue les étapes du processus d'inscription ainsi que de connexion . Ce jeu d'essai inclut des captures d'écran des interfaces utilisateur, ainsi que la gestion des erreurs sur les formulaires.

Étape 1 : Accéder à la page d'inscription

Description : L'utilisateur accède à la page d'inscription en cliquant sur la page “Connecte-toi” dans le menu de navigation puis sur le lien "Nouveau membre ? inscrivez-vous".

Capture d’ecran :

Étape 2 : Saisie des informations d'inscription

Description : L'utilisateur saisit les informations requises pour créer un compte, notamment un nom d'utilisateur, une adresse e-mail, un mot de passe.

Capture d’ecran :

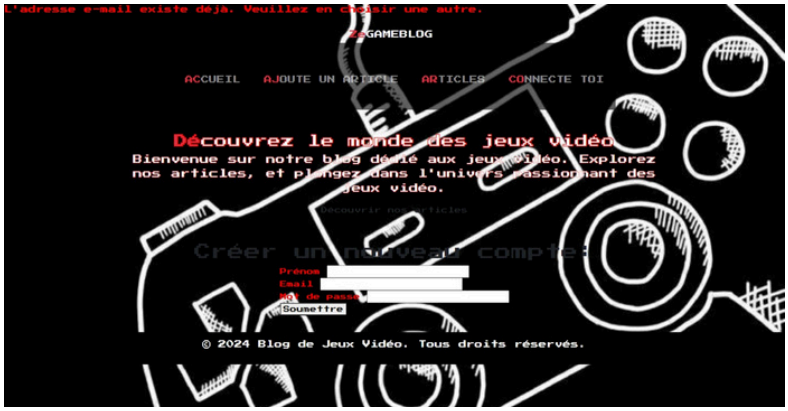
Étape 3 : Validation des informations et gestion des erreurs

Description : Lors de la soumission du formulaire, les informations saisies sont validées. Si des erreurs sont détectées (par exemple, le nom d'utilisateur est déjà pris, les mots de passe ne correspondent pas, ou l'adresse e-mail est invalide), des messages d'erreur appropriés s'affichent.

code erreur :

```php
// Vérifier si l'adresse e-mail existe déjà
if ($registerModel->emailExists($email)) {
    echo "<p style='color: red;'>L'adresse e-mail existe déjà. Veuillez en choisir une autre.</p>";
```

capture d'ecran :



Étape 4 : Soumission du formulaire et création de compte

Description : Si toutes les informations sont valides, l'utilisateur clique sur le bouton "Créer un compte". Les données sont envoyées au serveur pour vérification et création du compte. Une redirection va s'afficher avec un message "Inscription réussie. Redirection en cours..." qui redirigera l'utilisateur sur la page "Login" pour qu'il puisse se connecter avec le compte qu'il vient de créer.

code de redirection :

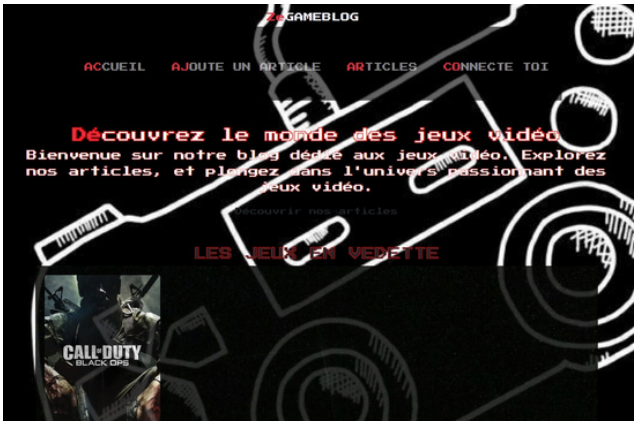
```php
if ($result) {
    echo "<p style='color: green;'>Inscription réussie. Redirection en cours...</p>";
    header("refresh:3;url=index.php?url=login"); // Rediriger après 3 secondes
```

Capture d'ecran :

Étape 1 : Accéder à la page de connexion

- **Description :** L'utilisateur accède à la page de connexion en cliquant sur le lien "Connecte-toi" dans le menu de navigation.

- **Capture d'écran:**



Étape 2 : Saisie des informations de connexion

- **Description :** L'utilisateur saisit son adresse e-mail et son mot de passe dans les champs de formulaire prévus à cet effet.

- **Capture d'écran :**

Étape 3 : Validation du formulaire

- **Description :** L'utilisateur clique sur le bouton "Se connecter". Le formulaire est soumis et le système vérifie les informations d'identification.

- **Capture d'écran:**

Gestion des erreurs

1. **Adresse e-mail ou mot de passe incorrect :**
    - **Description :** Si l'adresse e-mail ou le mot de passe est incorrect, un message d'erreur est affiché pour informer l'utilisateur.

- **Capture d'écran :**

- **Code de gestion des erreurs :**

```php
// Vérifier si la connexion a réussi
if ($result) {
    // echo "Connexion réussie";
    header("Location: index.php?url=profile&id=".$_SESSION['user_id']);
} else {
    echo "Adresse email ou mot de passe incorrect";
}
```

Champs vides :

- **Description :** Si l'utilisateur tente de soumettre le formulaire sans remplir tous les champs requis, un message d'erreur s'affiche. en haut a gauche qui sera quasiment pareil que celle de l'adresse mail ou mots de passe incorrect. il y'a en plus "cet adresse mail n'existe pas.

- **Capture d'écran :**

Code de gestion des erreurs :

```php
    //Authentification échouée
    if (empty($user)) {
        echo ("Cette adresse e-mail n'existe pas !");
    }
    return false;
}
```

Étape 4 : Redirection en cas de succès

- **Description :** Si les informations de connexion sont correctes, l'utilisateur est redirigé vers la page d'accueil ou son tableau de bord.

- **Capture d'écran :**

- **code de redirection :**

```php
if ($result) {
    // echo "Connexion réussie";
    header("Location: index.php?url=profile&id=" . $_SESSION['user_id']);
```

8/Veille technologique

J'ai utilisé **YouTube** https://www.youtube.com/
pour ses multiples tutoriels et vidéos pour acquérir un meilleur savoir sur plusieurs langages de développement tel que php et boostrap surtout pour le responsive. J'y ai aussi pu trouver des solutions aux problèmes rencontrés.

J'ai utilisé **GitHub** https://github.com/
car c'est une plateforme de développement collaboratif qui héberge pleins de projets open source de pleins d'autre développeurs beaucoup plus expérimentés.
Trouver des projets en explorant des dépôts GitHub qui peuvent être similaires à mon projet pour étudier leur structure et leur implémentation.

9/Problematique

Description du problème : Quand l'utilisateur se connecte et qu'il navigue sur d'autres pages il était déconnecté instantanément .

Comment maintenir la connexion de l'utilisateur lors de la navigation sur différentes pages du site sans qu'il ne soit déconnecté de manière inattendue ?

Solution :Pour résoudre ce problème, j'ai mis en place un mécanisme de gestion de session cohérent et persistant tout au long de la navigation sur le site. Voici les étapes que j'ai suivies pour y parvenir :

1. Initialisation des sessions
Pour garantir que les utilisateurs restent connectés lorsqu'ils naviguent sur le blog de jeux vidéo, j'ai veillé à ce que la session utilisateur soit initialisée correctement dès la connexion. Lorsqu'un utilisateur soumet ses informations de connexion, une session est démarrée avec session_start() et les informations de l'utilisateur, telles que son identifiant et son nom d'utilisateur, sont stockées dans S_SESSION. En sécurisant et en maintenant cette session active, l'utilisateur peut naviguer librement sur toutes les pages sans se déconnecter.

2. Vérification de l'état de connexion

Avant de charger chaque page, j'ai inclus un mécanisme de vérification de l'état de connexion de l'utilisateur pour garantir qu'il reste connecté. Ce mécanisme, souvent placé dans un contrôleur ou un middleware, vérifie si une session est active et si l'utilisateur est authentifié. Si l'utilisateur n'est pas connecté, il est redirigé vers la page de connexion.

Grâce à ces modifications, les utilisateurs peuvent maintenant naviguer sur toutes les pages du blog sans être déconnectés à chaque changement de page. Cela a considérablement amélioré l'expérience utilisateur en permettant une interaction plus fluide et continue avec le site. Les utilisateurs peuvent se connecter une seule fois et rester connectés pendant toute la durée de leur session, même s'ils visitent différentes sections du blog.

10/Conclusion Projet

En résumé, ce projet de blog de jeux vidéo a été conçu pour offrir une plateforme conviviale et interactive aux amateurs de jeux vidéo. Voici un récapitulatif des principaux aspects du projet :

Objectifs du Projet

Le principal objectif était de créer un espace où les utilisateurs peuvent publier des articles, lire des critiques de jeux et commenter les publications. Ce blog permet également de découvrir de nouveaux jeux vidéo grâce à une classification par genres et une présentation des dernières sorties.

Problématiques et Solutions

L'une des principales difficultés rencontrées était de maintenir la connexion de l'utilisateur active à travers la navigation sur différentes pages. Ce problème a été résolu en initialisant correctement les sessions et en stockant de manière sécurisée les informations utilisateur dans S_SESSION.

11/ Annexes

documentations supplémentaire , références,etc.