# Documentation du projet "Site Ligue\_1"

## Introduction

La documentation suivante décrit le projet "Site Ligue\_1", un site web dédié à la ligue de football française. Ce site permet aux utilisateurs de consulter des articles, de voir le classement des clubs, de commenter les articles et de gérer leur profil utilisateur. Le projet est développé en utilisant PHP, HTML, CSS et JavaScript, avec une architecture MVC (Modèle-Vue-Contrôleur).

## Architecture du projet

Le projet "Site Ligue\_1" suit une architecture MVC (Modèle-Vue-Contrôleur) pour séparer la logique de présentation, la logique métier et l'accès aux données. Voici une description de chaque composant de l'architecture :

### Modèle

Le modèle représente la couche de données du projet. Il est responsable de la gestion des données, de l'accès à la base de données et des opérations CRUD (Create, Read, Update, Delete). Les modèles utilisés dans le projet sont les suivants :

#### GestionBDD

La classe "GestionBDD" gère la connexion à la base de données MySQL et fournit des méthodes pour exécuter des requêtes SQL. Elle offre les fonctionnalités suivantes :

* **connect()** : Établit une connexion à la base de données MySQL.
* **query($sql)** : Exécute une requête SQL et retourne les résultats.
* **execute($sql)** : Exécute une requête SQL sans retour de résultats (pour les opérations CREATE, UPDATE, DELETE).

#### GestionArticle

Le modèle "GestionArticle" gère les opérations liées aux articles, y compris la création, la lecture, la mise à jour et la suppression. Il offre les fonctionnalités suivantes :

* **getArticles()** : Récupère tous les articles disponibles.
* **getArticleById($id)** : Récupère un article existant en fonction de son identifiant.
* **createArticle($titre, $contenu, $auteur)** : Crée un nouvel article avec les détails spécifiés.
* **updateArticle($id, $titre, $contenu)** : Met à jour les détails d'un article existant.
* **deleteArticle($id)** : Supprime un article existant en fonction de son identifiant.

#### GestionUtilisateur

Le modèle "GestionUtilisateur" gère les opérations liées aux utilisateurs, y compris la création, la lecture, la mise à jour et la suppression. Il offre les fonctionnalités suivantes :

* **getUserByEmail($email)** : Récupère un utilisateur existant en fonction de son adresse e-mail.
* **createUser($nom, $prenom, $email, $password, $clubId, $sexe)** : Crée un nouvel utilisateur avec les détails spécifiés.
* **updateUser($id, $nom, $prenom, $email, $clubId, $sexe)** : Met à jour les détails d'un utilisateur existant.
* **deleteUser($id)** : Supprime un utilisateur existant en fonction de son identifiant.

#### GestionClub

Le modèle "GestionClub" gère les opérations liées aux clubs de football, y compris la création, la lecture, la mise à jour et la suppression. Il offre les fonctionnalités suivantes :

* **getClubs()** : Récupère tous les clubs disponibles.
* **getClubById($id)** : Récupère un club existant en fonction de son identifiant.
* **createClub($nomClub)** : Crée un nouveau club avec le nom spécifié.
* **updateClub($id, $nomClub)** : Met à jour le nom d'un club existant.
* **deleteClub($id)** : Supprime un club existant en fonction de son identifiant.

### Contrôleur

Le contrôleur représente la couche logique du projet. Il est responsable de la gestion des requêtes utilisateur, de l'appel des modèles appropriés et de la coordination des actions. Les contrôleurs utilisés dans le projet sont les suivants :

#### c\_articles.php

Le contrôleur "c\_articles" gère les requêtes liées aux articles, notamment l'affichage de la liste des articles, la création d'un nouvel article, la mise à jour d'un article existant et la suppression d'un article. Il offre les fonctionnalités suivantes :

* **displayArticles()** : Affiche la liste des articles disponibles.
* **displayArticle($id)** : Affiche un article spécifique en fonction de son identifiant.
* **createArticle()** : Gère la création d'un nouvel article.
* **updateArticle($id)** : Gère la mise à jour d'un article existant.
* **deleteArticle($id)** : Gère la suppression d'un article existant.

#### c\_classement.php

Le contrôleur "c\_classement" gère les requêtes liées au classement des clubs de football, notamment l'affichage du classement de la saison en cours. Il offre les fonctionnalités suivantes :

* **displayClassement()** : Affiche le classement de la saison en cours.

#### c\_login.php

Le contrôleur "c\_login" gère les requêtes liées à l'authentification des utilisateurs, notamment la connexion et la déconnexion. Il offre les fonctionnalités suivantes :

* **login()** : Gère le processus de connexion de l'utilisateur.
* **logout()** : Gère le processus de déconnexion de l'utilisateur.

#### c\_register.php

Le contrôleur "c\_register" gère les requêtes liées à l'inscription des utilisateurs, notamment la création d'un nouveau compte utilisateur. Il offre les fonctionnalités suivantes :

* **register()** : Gère le processus d'inscription d'un nouvel utilisateur.

#### c\_userinfo.php

Le contrôleur "c\_userinfo" gère les requêtes liées aux informations utilisateur, notamment l'affichage du profil utilisateur, la gestion des paramètres utilisateur et la gestion des contacts. Il offre les fonctionnalités suivantes :

* **displayProfile()** : Affiche le profil de l'utilisateur connecté.
* **displaySettings()** : Gère les paramètres de l'utilisateur connecté.
* **displayContact()** : Gère la gestion des contacts.

#### c\_disconnect.php

Le contrôleur "c\_disconnect" gère la requête de déconnexion de l'utilisateur connecté.

### Vue

La vue représente la couche de présentation du projet. Elle est responsable de l'affichage des données et de l'interaction avec l'utilisateur. Les vues utilisées dans le projet sont les suivantes :

#### menu.php

La vue "menu" affiche la page d'accueil du site avec des informations sur la Ligue 1, des liens vers des articles et d'autres fonctionnalités.

#### classement.php

La vue "classement" affiche le classement de la saison en cours en fonction des résultats des matchs. Elle affiche également des détails tels que les buts marqués, les buts encaissés et le nombre de points pour chaque club.

#### articles.php

La vue "articles" affiche la liste des articles disponibles, avec la possibilité de lire un article complet et de commenter les articles.

#### login.php

La vue "login" affiche le formulaire de connexion pour les utilisateurs enregistrés.

#### register.php

La vue "register" affiche le formulaire d'inscription pour les nouveaux utilisateurs.

#### userinfo.php

La vue "userinfo" affiche les informations utilisateur, y compris le profil utilisateur, les paramètres utilisateur et les contacts.

#### header.php

La vue "header" affiche l'en-tête du site, y compris la barre de navigation et les liens de connexion ou de déconnexion en fonction de l'état de l'utilisateur.

#### footer.php

La vue "footer" affiche le pied de page du site, avec des informations supplémentaires et des liens.

### Router

Le router est responsable de la gestion des routes et de la redirection des requêtes utilisateur vers les contrôleurs appropriés. Le fichier "router.php" définit les règles de routage pour chaque URL et appelle le contrôleur correspondant en fonction de la demande utilisateur.

### Index

Le fichier "index.php" est le point d'entrée du site. Il inclut le fichier "header.php" pour afficher l'en-tête, le fichier "router.php" pour gérer les routes et le fichier "footer.php" pour afficher le pied de page.

## Description et exemple du code de certaines classes

### Classe GestionBDD

La classe **GestionBDD** permet de gérer la connexion à la base de données. Elle utilise les informations de connexion stockées dans le fichier **config/database.php**.

<?php

class GestionBDD {

private $host = 'localhost';

private $dbname = 'nom\_de\_la\_base\_de\_donnees';

private $username = 'nom\_utilisateur';

private $password = 'mot\_de\_passe';

public function connect() {

try {

$dsn = "mysql:host={$this->host};dbname={$this->dbname}";

$options = [

PDO::ATTR\_ERRMODE => PDO::ERRMODE\_EXCEPTION,

PDO::ATTR\_DEFAULT\_FETCH\_MODE => PDO::FETCH\_OBJ

];

return new PDO($dsn, $this->username, $this->password, $options);

} catch (PDOException $e) {

die("Erreur de connexion à la base de données : " . $e->getMessage());

}

}

}

?>

### Classe GestionArticle

La classe **GestionArticle** gère la manipulation des articles dans la base de données. Elle utilise la connexion à la base de données fournie par la classe **GestionBDD**.

<?php

class GestionArticle {

private $db;

public function \_\_construct($db) {

$this->db = $db;

}

public function getArticles() {

// Code pour récupérer tous les articles de la base de données

}

public function getArticleById($id) {

// Code pour récupérer un article par son identifiant

}

public function createArticle($title, $content) {

// Code pour créer un nouvel article dans la base de données

}

public function updateArticle($id, $title, $content) {

// Code pour mettre à jour un article dans la base de données

}

public function deleteArticle($id) {

// Code pour supprimer un article de la base de données

}

}

?>

### Classe Article

La classe **Article** représente un article avec ses propriétés et méthodes associées

<?php

class Article {

private $id;

private $title;

private $content;

public function \_\_construct($id, $title, $content) {

$this->id = $id;

$this->title = $title;

$this->content = $content;

}

public function getId() {

return $this->id;

}

public function getTitle() {

return $this->title;

}

public function getContent() {

return $this->content;

}

}

?>

### Classe User

La classe **User** représente un utilisateur avec ses propriétés et méthodes associées. Voici un exemple de code commenté :

<?php

class User {

private $id;

private $username;

private $email;

public function \_\_construct($id, $username, $email) {

$this->id = $id;

$this->username = $username;

$this->email = $email;

}

public function getId() {

return $this->id;

}

public function getUsername() {

return $this->username;

}

public function getEmail() {

return $this->email;

}

}

?>

### Classe Club

La classe **Club** représente un club de football avec ses propriétés et méthodes associées

<?php

class Club {

private $id;

private $name;

private $country;

public function \_\_construct($id, $name, $country) {

$this->id = $id;

$this->name = $name;

$this->country = $country;

}

public function getId() {

return $this->id;

}

public function getName() {

return $this->name;

}

public function getCountry() {

return $this->country;

}

}

?>

## Conclusion

La documentation ci-dessus décrit l'architecture et les composants du projet "Site Ligue\_1". Elle couvre les modèles, les contrôleurs, les vues, le router et les fichiers d'entrée. Cette documentation fournit une compréhension globale du projet et peut servir de référence pour le développement ultérieur et les modifications du site.