Documentation Site Web Ligue 1

Sommaire

1. Introduction

Lors de ma 2e année de BTS SIO SLAM, nous avons étudié les langages Web (HTML, CSS, PHP, JavaScript), mais nous nous sommes particulièrement intéressés au côté backend de la programmation au travers d'un projet Web. Ce projet consiste à créer un site internet de foot sur la ligue 1, comportant différentes pages et interrogeant une Base de données PostgreSQL.

Nous avons travaillé sous le modèle MVC qui est une bonne façon de travailler, car elle permet de répartir les logiques de code dans différentes classes.

1.1. Différence entre Frontend et backend

Frontend : Imaginez que vous êtes dans un restaurant. Le frontend serait la salle de restaurant où vous vous asseyez, le menu que vous parcourez, et le serveur qui prend votre commande. En termes de développement web, le frontend est tout ce que l'utilisateur voit et avec quoi il interagit sur le site web. Cela comprend la conception, la mise en page, les couleurs, les boutons, les formulaires et tout autre élément visuel. Les technologies couramment utilisées dans le développement frontend comprennent HTML, CSS et JavaScript.

Backend : En revenant à notre analogie du restaurant, le backend serait la cuisine, où le chef prépare votre repas, et tous les processus qui se déroulent en coulisses pour s'assurer que votre repas arrive à votre table comme vous l'avez commandé. Dans le développement web, le backend fait référence à tout ce qui se passe en coulisses d'un site web. Cela comprend la gestion des bases de données, la logique métier (comme les calculs), la sécurité, l'authentification, et tout ce qui est nécessaire pour traiter les demandes de l'utilisateur et renvoyer les bonnes données. Les technologies couramment utilisées dans le développement backend comprennent Node.js, Ruby, Python, PHP, Java, et d'autres.

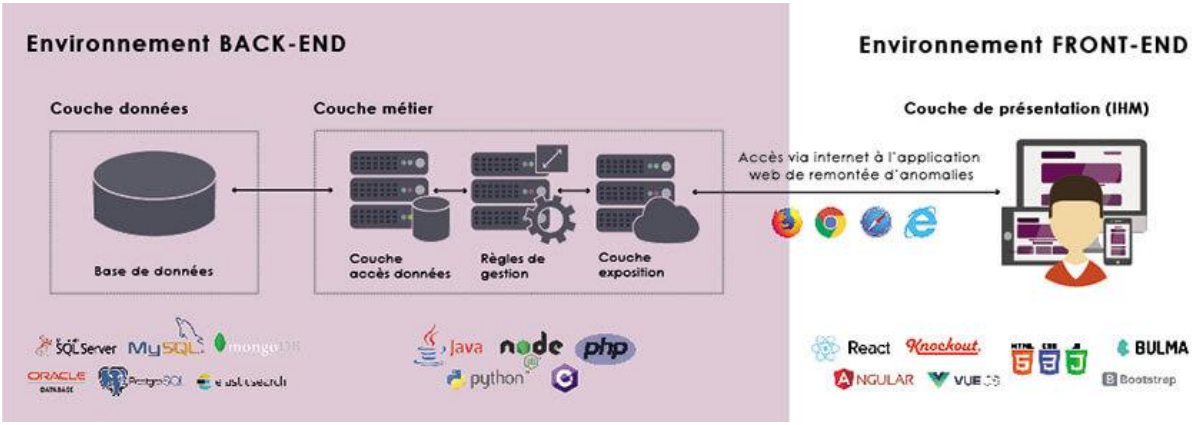


Figure 1 : schéma représentatif de la différence entre backend et frontend

La différence principale entre le frontend et le backend réside donc dans leur rôle et leur interaction avec l'utilisateur. Le frontend est tout ce que l'utilisateur voit et interagit, tandis que le backend est tout ce qui se passe en coulisses pour rendre cela possible.

2. Prérequis

Le site internet a été développé en utilisant les langages de programmation principaux du web, à savoir PHP, HTML et JavaScript. Actuellement, le site fonctionne en local car il n'est pas encore hébergé sur un serveur web. Pour la gestion des données, nous utilisons une base de données PostgreSQL, également en local.

Le code source du site est disponible sur GitHub. Pour y accéder, veuillez cliquer ici.

3. Installation et configuration

Pour pouvoir accéder au site web et profiter de celui-ci, je vais vous présenter comment pouvoir faire tourner un site web multi page sur une machine sur Microsoft Visual Studio, via une commande.

Vous devez avoir télécharger PHP préalablement puis avoir mis dans les variables d'environnement le path vers le dossier de PHP que vous venez de télécharger.

Ensuite il vous suffit de lancer Visual studio et d'ouvrir le projet, pour ensuite dans le terminal entrer cette commande : « PHP -S localhost :3000 -t src »

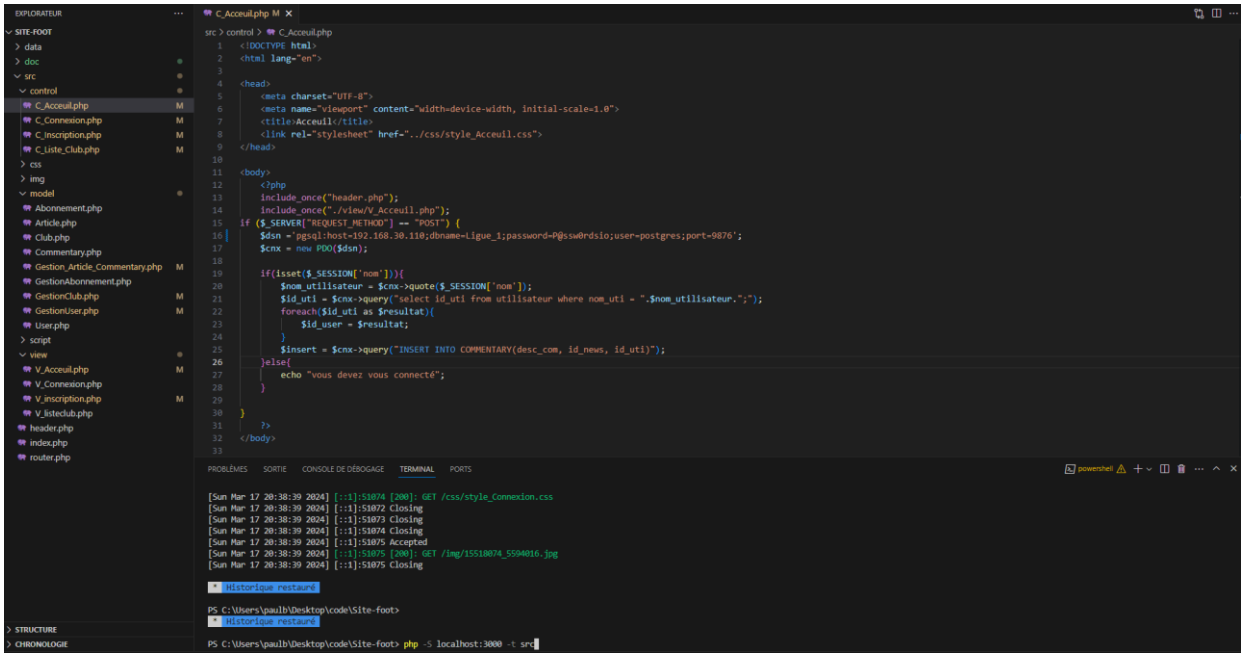


Figure 2 : Screenshot de Visual Studio

Ensuite vous pouvez accéder au site internet via le lien « localhost:3000 ».

4. Structure du projet

Maintenant nous verrons toute la structure du code derrière le site internet.

4.1. Présentation rapide du modèle MVC

Le modèle MVC est composé de 3 grands packages le modèle, la vue et le contrôleur, chacun a son rôle dans une application, voici leurs définitions respectives :

Modèle : Dans le modèle MVC, le "Modèle" représente les données et les règles métier de votre application. Il interagit avec la base de données pour créer, lire, mettre à jour et supprimer des informations.

Vue : La "Vue" est responsable de la présentation des données à l'utilisateur. Elle prend les données du modèle et les affiche d'une manière que l'utilisateur peut comprendre.

Contrôleur : Le "Contrôleur" agit comme un intermédiaire entre le modèle et la vue. Il traite les entrées de l'utilisateur, interagit avec le modèle pour obtenir ou modifier des données, et met à jour la vue en conséquence.

4.2. Détails du dossier src

Le dossier src comporte tout le code du projet et permet d'accéder à toute la logique et le fonctionnement du site internet.

4.2.1. Détails du dossier model

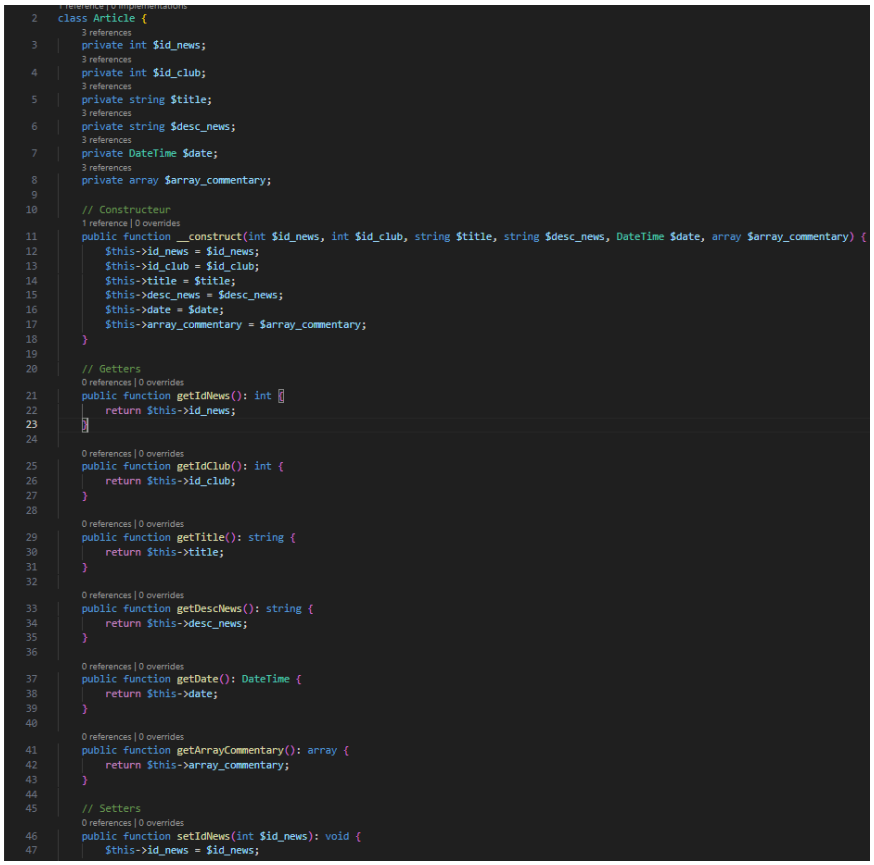
Le dossier model comme expliquer précédemment comporte toutes les classes Modèle, c'est-à-dire que les classes modèle vont interagir avec la base de données, il y'aura des classes qui ont des paramètres représentant des tables sur la base de données.

4.2.1.1. La classe Article

Cette classe représente un article associé à un club.

- Propriétés : La classe a trois propriétés privées :
    - $id_news (privé): Un entier qui représente l'identifiant unique de l'article.
    - $id_club (privé): Un entier qui représente l'identifiant du club associé à l'article.
    - $title (privé): Une chaîne de caractères qui représente le titre de l'article.
    - $desc_news (privé): Une chaîne de caractères qui représente la description de l'article.
    - $date (privé): Un objet de type DateTime qui représente la date de publication de l'article.
    - $array_commentary (privé): Un tableau qui contient les commentaires associés à l'article.

- Constructeur : Le constructeur de la classe prend trois paramètres ($id_news, $desc_news, $array_commentary) qui sont utilisés pour initialiser les propriétés de l'objet lors de sa création.
- Getters:
    - getIdNews(): Renvoie l'identifiant unique de l'article.
    - getIdClub(): Renvoie l'identifiant du club associé à l'article.
    - getTitle(): Renvoie le titre de l'article.
    - getDescNews(): Renvoie la description de l'article.
    - getDate(): Renvoie la date de publication de l'article sous forme d'objet DateTime.
    - getArrayCommentary(): Renvoie le tableau des commentaires associés à l'article.
- Setters:
    - setIdNews($id_news): Définie l'identifiant unique de l'article.
    - setIdClub($id_club): Définie l'identifiant du club associé à l'article.
    - setTitle($title): Définie le titre de l'article.
    - setDescNews($desc_news): Définie la description de l'article.
    - setDate(DateTime $date): Définie la date de publication de l'article sous forme d'objet DateTime.
    - setArrayCommentary($array_commentary): Définie le tableau des commentaires associés à l'article.



```php
class Article {
    private int $id_news;
    private int $id_club;
    private string $title;
    private string $desc_news;
    private DateTime $date;
    private array $array_commentary;

    // Constructeur
    public function __construct(int $id_news, int $id_club, string $title, string $desc_news, DateTime $date, array $array_commentary) {
        $this->id_news = $id_news;
        $this->id_club = $id_club;
        $this->title = $title;
        $this->desc_news = $desc_news;
        $this->date = $date;
        $this->array_commentary = $array_commentary;
    }

    // Getters
    public function getIdNews(): int {
        return $this->id_news;
    }

    public function getIdClub(): int {
        return $this->id_club;
    }

    public function getTitle(): string {
        return $this->title;
    }

    public function getDescNews(): string {
        return $this->desc_news;
    }

    public function getDate(): DateTime {
        return $this->date;
    }

    public function getArrayCommentary(): array {
        return $this->array_commentary;
    }

    // Setters
    public function setIdNews(int $id_news): void {
        $this->id_news = $id_news;
```

4.2.1.2. La classe Club

Cette classe représente un club sportif.

- Propriétés:
  - $id (privé): Un entier qui représente l'identifiant unique du club.
  - $nom_club (privé): Une chaîne de caractères qui représente le nom du club.
  - $ligue_club (privé): Une chaîne de caractères qui représente la ligue dans laquelle évolue le club.
- Constructeur : Le constructeur de la classe prend trois paramètres ($id, $nom_club, $ligue_club) qui sont utilisés pour initialiser les propriétés de l'objet lors de sa création.
- Getters:
  - getId(): Renvoie l'identifiant unique du club.
  - getNomClub(): Renvoie le nom du club.
  - getLigueClub(): Renvoie la ligue dans laquelle évolue le club.
- Setters:
  - setId($id): Définie l'identifiant unique du club.
  - setNomClub($nom_club): Définie le nom du club.

- setLigueClub($ligue_club): Définie la ligue dans laquelle évolue le club.

src > model > Club.php

```php
<?php
class Club {
    private int $id;
    private string $nom_club;
    private string $ligue_club;

    public function __construct(int $id, string $nom_club, string $ligue_club) {
        $this->id = $id;
        $this->nom_club = $nom_club;
        $this->ligue_club = $ligue_club;
    }

    public function getId(): int {
        return $this->id;
    }

    public function setId(int $id): void {
        $this->id = $id;
    }

    public function getNomClub(): string {
        return $this->nom_club;
    }

    public function setNomClub(string $nom_club): void {
        $this->nom_club = $nom_club;
    }

    public function getLigueClub(): string {
        return $this->ligue_club;
    }

    public function setLigueClub(string $ligue_club): void {
        $this->ligue_club = $ligue_club;
    }
}
?>
```

4.2.1.3. La classe Commentary

Cette classe représente un commentaire associé à un article.

- Propriétés:
  - $id_news (privé): Un entier qui représente l'identifiant de l'article associé au commentaire.
  - $id_uti (privé): Une chaîne de caractères qui représente l'identifiant de l'utilisateur ayant écrit le commentaire.
  - $desc_commentary (privé): Une chaîne de caractères qui représente le contenu du commentaire.
  - $date (privé): Un objet de type DateTime qui représente la date de publication du commentaire.

- Constructeur : Le constructeur de la classe prend trois paramètres ($id_news, $desc_com, $id_uti) qui sont utilisés pour initialiser les propriétés de l'objet lors de sa création.
- Getters:
    - getIdNews(): Renvoie l'identifiant de l'article associé au commentaire.
    - getIdUti(): Renvoie l'identifiant de l'utilisateur ayant écrit le commentaire.
    - getDescCommentary(): Renvoie le contenu du commentaire.
    - getDate(): Renvoie la date de publication du commentaire sous forme d'objet DateTime.
- Setters:
    - setIdNews(int $id_news): Définie l'identifiant de l'article associé au commentaire.
    - setIdUti(String $id_uti): Définie l'identifiant de l'utilisateur ayant écrit le commentaire.
    - setDescCommentary(string $desc_commentary): Définie le contenu du commentaire.
    - setDate(DateTime $date): Définie la date de publication du commentaire sous forme d'objet DateTime.

```php
1 reference | 0 implementations
class Commentary {
    3 references
    private int $id_news;
    3 references
    private String $id_uti;
    3 references
    private string $desc_commentary;
    3 references
    private DateTime $date;

    // Constructeur
    1 reference | 0 overrides
    public function __construct(int $id_news, String $id_uti, string $desc_commentary, DateTime $date) {
        $this->id_news = $id_news;
        $this->id_uti = $id_uti;
        $this->desc_commentary = $desc_commentary;
        $this->date = $date;
    }

    // Getters
    0 references | 0 overrides
    public function getIdNews(): int {
        return $this->id_news;
    }

    0 references | 0 overrides
    public function getIdUti(): String {
        return $this->id_uti;
    }

    0 references | 0 overrides
    public function getDescCommentary(): string {
        return $this->desc_commentary;
    }

    0 references | 0 overrides
    public function getDate(): DateTime {
        return $this->date;
    }

    // Setters
    0 references | 0 overrides
    public function setIdNews(int $id_news): void {
        $this->id_news = $id_news;
    }

    0 references | 0 overrides
    public function setIdUti(String $id_uti): void {
        $this->id_uti = $id_uti;
    }

    0 references | 0 overrides
    public function setDescCommentary(string $desc_commentary): void {
        $this->desc_commentary = $desc_commentary;
    }
```

4.2.1.4. La classe User

Cette classe représente un utilisateur du site web.

- Propriétés:
    - $id_club (privé): Un entier qui représente l'identifiant du club auquel l'utilisateur est rattaché (peut être null si l'utilisateur n'est pas membre d'un club).
    - $nom (privé): Une chaîne de caractères qui représente le nom de l'utilisateur.

  - $prenom (privé): Une chaîne de caractères qui représente le prénom de l'utilisateur.
  - $mail (privé): Une chaîne de caractères qui représente l'adresse email de l'utilisateur.
  - $mdp (privé): Une chaîne de caractères qui représente le mot de passe de l'utilisateur (il est recommandé de stocker le mot de passe de façon sécurisée en utilisant un algorithme de hash).
  - $sexe (privé): Une chaîne de caractères qui représente le sexe de l'utilisateur.
  - $image (privé): Une chaîne de caractères qui représente le chemin d'accès à l'image de profil de l'utilisateur (peut être null si l'utilisateur n'a pas d'image de profil).

- Constructeur : Le constructeur de la classe prend sept paramètres ($id_club, $nom, $prenom, $mail, $mdp, $sexe, $image) qui sont utilisés pour initialiser les propriétés de l'objet lors de sa création.
- Getters:
  - getNom(): Renvoie le nom de l'utilisateur.
  - getPrenom(): Renvoie le prénom de l'utilisateur.
  - getMail(): Renvoie l'adresse email de l'utilisateur.
  - getMdp(): Renvoie le mot de passe de l'utilisateur (attention à la sécurité).
  - getSexe(): Renvoie le sexe de l'utilisateur.
  - getIdClub(): Renvoie l'identifiant du club auquel l'utilisateur est rattaché.
  - getImage(): Renvoie le chemin d'accès à l'image de profil de l'utilisateur.
- Setters:
  - setNom(string $nom): Définie le nom de l'utilisateur.
  - setPrenom(string $prenom): Définie le prénom de l'utilisateur.
  - setMail(string $mail): Définie l'adresse email de l'utilisateur.
  - setMdp($mdp): Attention à la sécurité - Définie le mot de passe de l'utilisateur (il est fortement recommandé de stocker le mot de passe de façon sécurisée en utilisant un algorithme de hash).
  - setSexe(string $sexe): Définie le sexe de l'utilisateur.
  - setIdClub(int $id_club): Définie l'identifiant du club auquel l'utilisateur est rattaché.
  - setImage(string $image): Définie le chemin d'accès à l'image de profil de l'utilisateur.

src > model > User.php

```php
<?php
class User {
    private int $id_club;
    private string $nom;
    private string $prenom;
    private string $mail;
    private string $mdp;
    private string $sexe;
    private string $image;

    public function __construct(int $id_club, string $nom, string $prenom, string $mail, string $mdp, string $sexe, string $image) {
        $this->nom = $nom;
        $this->prenom = $prenom;
        $this->mail = $mail;
        $this->mdp = $mdp;
        $this->sexe = $sexe;
        $this->id_club = $id_club;
        $this->image = $image;
    }

    public function getNom(): string {
        return $this->nom;
    }

    public function setNom(string $nom) {
        $this->nom = $nom;
    }

    public function getPrenom(): string {
        return $this->prenom;
    }

    public function setPrenom(string $prenom) {
        $this->prenom = $prenom;
    }

    public function getMail(): string {
        return $this->mail;
    }

    public function setMail(string $mail) {
        $this->mail = $mail;
    }
```

4.2.1.5. La classe Championnat

Cette classe représente le classement d'un championnat sportif.

- Propriétés:
    - $nom_club (privé): Une chaîne de caractères qui représente le nom du club.
    - $matchs_gagnes (privé): Un entier qui représente le nombre de matchs gagnés par le club.
    - $matchs_perdus (privé): Un entier qui représente le nombre de matchs perdus par le club.
    - $matchs_nuls (privé): Un entier qui représente le nombre de matchs nuls du club.
    - $buts_marques (privé): Un entier qui représente le nombre de buts marqués par le club.
    - $buts_encaissees (privé): Un entier qui représente le nombre de buts encaissés par le club.

  - $difference_buts (privé): Un entier qui représente la différence de buts du club (buts marqués - buts encaissés).
  - $nb_points (privé): Un entier qui représente le nombre de points du club.
  - $logo_club (privé): Une chaîne de caractères qui représente le chemin d'accès au logo du club (peut être null si le logo n'est pas disponible).

- Getters:
  - getNomClub(): Renvoie le nom du club.
  - getMatchsGagnes(): Renvoie le nombre de matchs gagnés par le club.
  - getMatchsPerdus(): Renvoie le nombre de matchs perdus par le club.
  - getMatchsNuls(): Renvoie le nombre de matchs nuls du club.
  - getButsMarques(): Renvoie le nombre de buts marqués par le club.
  - getButsEncaissees(): Renvoie le nombre de buts encaissés par le club.
  - getDifferenceButs(): Renvoie la différence de buts du club.
  - getNbPoints(): Renvoie le nombre de points du club.
  - getLogoClub(): Renvoie le chemin d'accès au logo du club.
- Setters:
  - setNomClub(string $nom_club): Définie le nom du club.
  - setMatchsGagnes(int $matchs_gagnes): Définie le nombre de matchs gagnés par le club.
  - setMatchsPerdus(int $matchs_perdus): Définie le nombre de matchs perdus par le club.
  - setMatchsNuls(int $matchs_nuls): Définie le nombre de matchs nuls du club.
  - setButsMarques(int $buts_marques): Définie le nombre de buts marqués par le club.
  - setButsEncaissees(int $buts_encaissees): Définie le nombre de buts encaissés par le club.
  - setDifferenceButs(int $difference_buts): Définie la différence de buts du club.
  - setNbPoints(int $nb_points): Définie le nombre de points du club.

```php
class Championnat
{
    3 references
    private string $nom_club;
    3 references
    private int $matchs_gagnes;
    3 references
    private int $matchs_perdus;
    3 references
    private int $matchs_nuls;
    3 references
    private int $buts_marques;
    3 references
    private int $buts_encaissees;
    3 references
    private int $difference_buts;
    3 references
    private int $nb_points;
    3 references
    private string $logo_club;

    1 reference | 0 overrides
    public function __construct(string $nom_club,int $matchs_gagnes,int $matchs_perdus,int $matchs_nuls,int $b
        $this->nom_club = $nom_club;
        $this->matchs_gagnes = $matchs_gagnes;
        $this->matchs_perdus = $matchs_perdus;
        $this->matchs_nuls = $matchs_nuls;
        $this->buts_marques = $buts_marques;
        $this->buts_encaissees = $buts_encaissees;
        $this->difference_buts = $difference_buts;
        $this->nb_points = $nb_points;
        $this->logo_club = $logo_club;
    }

    0 references | 0 overrides
    public function getNomClub(): string
    {
        return $this->nom_club;
    }

    0 references | 0 overrides
    public function setNomClub(string $nom_club): void
    {
        $this->nom_club = $nom_club;
    }

    0 references | 0 overrides
    public function getMatchsGagnes(): int
    {
        return $this->matchs_gagnes;
    }

    0 references | 0 overrides
    public function setMatchsGagnes(int $matchs_gagnes): void
    {
        $this->matchs_gagnes = $matchs_gagnes;
    }

    0 references | 0 overrides
    public function getMatchsPerdus(): int
```

4.2.2. Détails du dossier view

Le dossier View comporte les Classe PHP qui sont le frontend des pages du site. Elles sont là pour contenir le code de ce que l'utilisateur voit.

4.2.2.1. La classe V_Acceuil

Voici le contenu de la classe V_Acceuil :

- En-tête:
    - Un titre "Mon Site de Football"
    - Un lien vers la page d'inscription
    - Un lien vers la page de classement
- Section principale:
    - Un message de bienvenue avec une brève description du site
    - Une section "Les Derniers articles de la ligue 1 !"
    - Une liste d'articles récents avec leurs titres, descriptions et commentaires
- Commentaires:
    - Chaque article affiche une section "Les commentaires :"
    - Chaque commentaire affiche le nom d'utilisateur, le contenu du commentaire et une rupture de ligne.
    - Un formulaire permet d'ajouter un nouveau commentaire à un article.

4.2.2.2. La classe V_Connexion

Cette classe V_connexion est très simpliste, car elle contient simplement des zones de texte avec login et mot de passe pour que l'utilisateur se connecte avec un bouton se connecter.

4.2.2.3. La classe V_inscription

Cette classe V_inscription permet à un utilisateur de se créer un compte, en entrant ses informations personnelles, tel que :

- Son nom
- Son prénom
- Son adresse électronique
- Son mot de passe et la confirmation de celui-ci
- Son sexe
- Une image de profil
- Le championnat qui l'intéresse
- Son club favori

Pour ensuite la validé via le bouton « Validez »

4.2.2.4. La classe V_Classement_saison

Cette classe V_classement_saison permet à l'utilisateur de consulter le classement des équipes de Ligue 1, sur les différentes saisons ainsi que sur la saison actuelle.

L'utilisateur peut choisir l'année de la saison via une liste déroulante.

4.2.2.5. La classe V_Profil

Cette classe V_Profil permet a l'utilisateur ,lorsqu'il est connecté, de pouvoir accéder a son profil avec ses informations et la possibilité de se déconnecter.

4.2.3. Details du dossier control

Le dossier control comporte toutes les classes qui ont de la logique de traitement de donnée et qui utilisent les classes de vue.

4.2.3.1. La classe C_Acceuil

La classe C_Acceuil vas contenir la vue de la classe accueil et tout le traitement qu'il y a derrière :

- Inclut le fichier "header.php" qui contient le code pour l'en-tête du site.
- Inclut le fichier "V_Acceuil.php" qui contient le code pour la vue d'accueil du site.
- Vérifie si la méthode de requête est POST, ce qui signifie qu'un formulaire a été soumis.
- Établit une connexion à une base de données PostgreSQL.
- Si une session est en cours et qu'un nom d'utilisateur est défini, le code fait plusieurs choses :
    - Il récupère l'identifiant de l'utilisateur à partir de la base de données.
    - Il insère un nouveau commentaire dans la base de données.
- Si aucune session n'est en cours, le code affiche un message indiquant à l'utilisateur qu'il doit se connecter.

4.2.3.2. La classe C_Connexion

La classe C_Connexion vas contenir la vue de la classe Connexion mais elle va avoir toute la logique derrière les interactions que fait l'utilisateur :

- Inclut le fichier de vue ainsi que le header
- récupère l'adresse électronique et le mot de passe du formulaire. Le mot de passe est haché avec MD5 pour des raisons de sécurité.
- exécute une requête SQL pour sélectionner l'utilisateur correspondant à l'adresse électronique et au mot de passe fournis.

- Si un utilisateur correspondant est trouvé (c'est-à-dire si le nombre de lignes retournées par la requête est 1), le code fait plusieurs choses :
    - Il récupère les informations de l'utilisateur de la base de données et les stocke dans des variables de session.
    - Il redirige l'utilisateur vers la page "Connexion.php".
- Si aucun utilisateur correspondant n'est trouvé, le code affiche un message d'erreur.

4.2.3.3. La classe C_Inscription

La classe C_Inscription vas contenir la vue de la classe Inscription mais elle va avoir toute la logique derrière les interactions que fait l'utilisateur :

- Inclut la vue correspondante
- Vérifie si la méthode de requête est POST, ce qui signifie qu'un formulaire a été soumis.
- Si c'est le cas, le code fait plusieurs choses :
    - Établit une connexion à une base de données PostgreSQL.
    - Récupère les informations du formulaire d'inscription.
    - Si une image a été téléchargée, le code la déplace dans un répertoire spécifique et enregistre le chemin de l'image.
    - Crée un nouvel objet User avec les informations du formulaire et l'image téléchargée.
    - Envoie l'objet User à la base de données.
    - Affiche un message indiquant que la création du compte a réussi.
- Si la méthode de requête n'est pas POST, le code inclut le fichier "V_inscription.php" qui contient probablement le code pour la vue de la page d'inscription.

4.2.3.4. La classe C_Liste_Club

La classe C_Liste_Club vas contenir la vue de la classe ListeClub mais elle va avoir toute la logique derrière les interactions que fait l'utilisateur :

- Inclus la vue correspondante ainsi que le header pour permettre a l'utilisateur de naviguer entre les pages
- Établit une connexion à une base de données PostgreSQL.
- Crée un nouvel objet GestionClub est utilisé pour gérer les opérations liées aux clubs.
    - Appelle la méthode getLIsteClub() de l'objet GestionClub pour obtenir une liste de clubs.

Cette page ne contient pas énormément de code mais permet simplement de montrer a l'utilisateur les clubs existant avec leurs championnats.

4.2.4. Details du dossier script

Le dossier script contient un fichier script.js qui lui vas contenir du code javascript que le code du site va interroger.

JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives et à ce titre est une partie essentielle des applications web. Ce langage permet de rendre vivant les pages et de les rendre dynamique et agréable pour l'utilisateur.

Le fichier comporte 5 classes.

4.2.4.1. La classe checked_form

```
function checked_form(event) {
    let message_formulaire_false = document.getElementById("message_non_valide");
    if (same_password() && validate() && before_submit()) {
        var f = document.getElementById("formulaire");
        f.submit();

    }else{
        message_formulaire_false.innerHTML = "Vous avez mal informé les informations dans le formulaire !";
        event.preventDefault();
    }

}
```

Cette fonction est appelée lors de la soumission du formulaire. Elle vérifie si les mots de passe correspondent, si le mot de passe est fort et si tous les champs nécessaires sont remplis en utilisant les fonctions same_password(), validate() et before_submit(). Si toutes ces conditions sont remplies, le formulaire est soumis. Sinon, un message d'erreur est affiché et la soumission du formulaire est empêchée.

4.2.4.2. La classe validate

```
function validate() {
    let msg;
    let str = document.getElementById("mdp").value;

    if (str.match( /[0-9]/g) &&
            str.match( /[A-Z]/g) &&
            str.match(/[a-z]/g) &&
            str.match( /[^a-zA-Z\d]/g) &&
            str.length >= 12) {
        msg = "<p style='color:green'>Mot de passe fort.</p>";
    }else{
        msg = "<p style='color:red'>Mot de passe faible.</p>";
        }
    document.getElementById("message_conforme").innerHTML= msg;
    if (msg === "<p style='color:green'>Mot de passe fort.</p>"){
        return true;
    }else {
        return false;
    }
}
```

Cette fonction vérifie la force du mot de passe. Elle vérifie si le mot de passe contient au moins un chiffre, une lettre majuscule, une lettre minuscule, un caractère spécial et s'il a une longueur d'au moins 12 caractères. Si le mot de passe est fort, un message vert est affiché et la fonction retourne true. Sinon, un message rouge est affiché et la fonction retourne false.

4.2.4.3. La classe same_password

```
35  function same_password(){
36      let mdp = document.getElementById("mdp").value;
37      let confirmmdp = document.getElementById("confirmmdp").value;
38      let message = document.getElementById("passwordmessage");
39      let invalide_message = document.getElementById("message_invalide");
40
41      if (mdp.trim() !== "" && confirmmdp.trim() !== "") {
42          if (mdp === confirmmdp) {
43              message.innerHTML = "correct";
44              invalide_message.innerHTML = "";
45              return true;
46          } else {
47              invalide_message.innerHTML = "Mot de passe différent";
48              message.innerHTML = "";
49              return false;
50          }
51      } else {
52          invalide_message.innerHTML = "";
53          message.innerHTML = "";
54          return false;
55      }
56  }
57
```

Cette fonction vérifie si le mot de passe et la confirmation du mot de passe correspondent. Si c'est le cas, un message "correct" est affiché et la fonction retourne true. Sinon, un message d'erreur est affiché et la fonction retourne false.

4.2.4.4. La classe before_submit

```
58  function before_submit(){
59      let nom = document.getElementById("nom").value;
60      let prenom = document.getElementById("prenom").value;
61      let mail = document.getElementById("mail").value;
62      let sexe = document.getElementById("sexe").value;
63      let club_pref = document.getElementById("club_pref").value;
64      let ligue = document.getElementById("championnat").value;
65
66      if(nom.trim() !== "" && prenom.trim() !== "" && sexe !== "" && ligue !== "" && mail.includes("@")){
67          return true;
68      }else{
69          return false;
70      }
71
72
73  }
```

Cette fonction vérifie si tous les champs nécessaires sont remplis et si l'adresse électronique contient un "@". Si c'est le cas, la fonction retourne true. Sinon, elle retourne false.

4.2.4.5. La classe selectALLClubs

```
75  // Fonction pour cocher ou décocher toutes les cases individuelles
76  function selectAllClubs() {
77      var checkboxes = document.querySelectorAll('.select-all');
78      var clubCheckboxes = document.querySelectorAll('input[name="club_news[]"]');
79
80      for (var i = 0; i < clubCheckboxes.length; i++) {
81          clubCheckboxes[i].checked = checkboxes[0].checked;
82      }
83  }
```

Cette fonction est utilisée pour cocher ou décocher toutes les cases de sélection des clubs. Si la case “Sélectionner tout” est cochée, toutes les cases individuelles sont cochées. Si la case “Sélectionner tout” est décochée, toutes les cases individuelles sont décochées.

4.2.5. Détails du dossier img

Le dossier img contient toutes les photos de profil des utilisateurs avec pour nom à chaque fois :

Nomutilisateur_prenomutilisateur_nomdelimageimporter.jpg

4.2.6. Détails du dossier Css

Le Css est utilisé en Web pour uniquement faire du visuel, Frontend, ce langage nous permet de mettre des couleurs sur les pages, gérer les positions des paragraphes, gérer le responsives.

Il est la pour rendre agréable la visite du site web pour l’utilisateur.

5. Fonctionnement du site web

Pour généraliser le Site web voici un résumé de toutes les fonctionnalités que nous offre le site web :

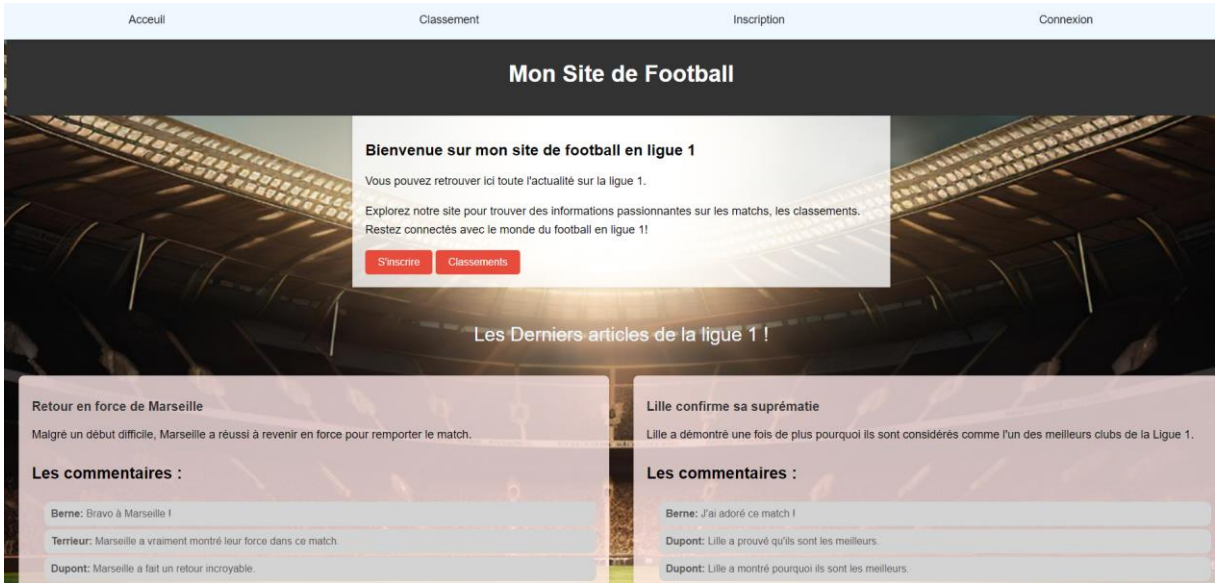
Lorsqu’un utilisateur visite le site, il peut s’inscrire en fournissant des informations telles que son nom, son prénom, son adresse électronique, son mot de passe, son sexe, son club préféré et une image de profil. Ces informations sont stockées dans un objet User et envoyées à la base de données.

Une fois inscrit, l’utilisateur peut se connecter. Le site vérifie les informations de connexion en les comparant aux données stockées dans la base de données. Si les informations sont correctes, l’utilisateur est connecté et une session est créée.

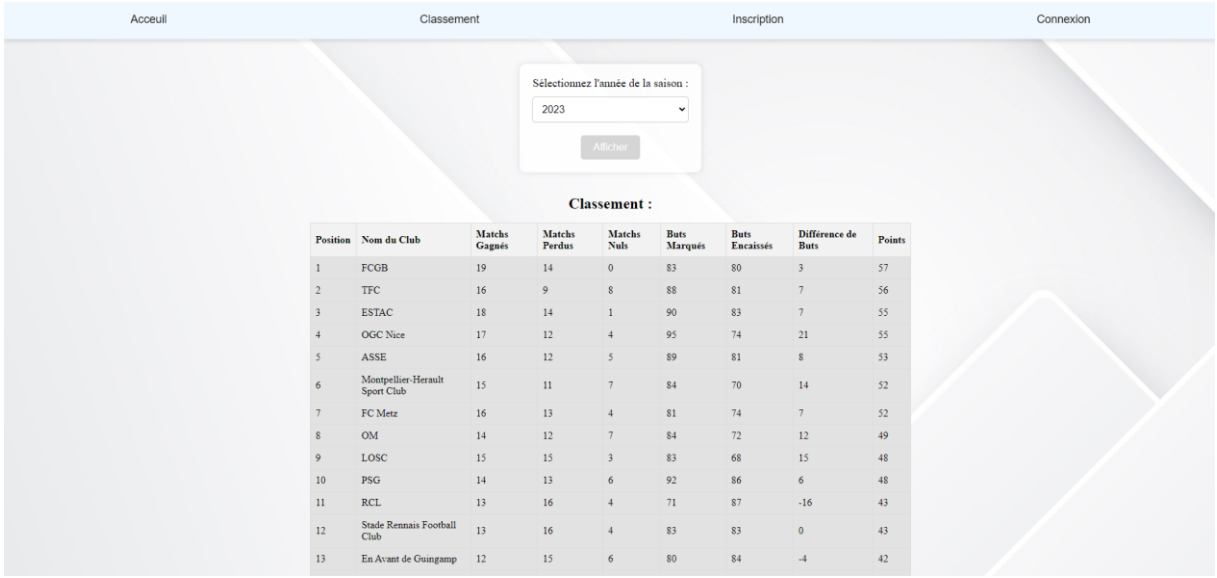
Le site propose également une liste de clubs que l’utilisateur peut consulter. Chaque club est représenté par un objet Club, qui contient des informations telles que l’identifiant du club, le nom du club et la ligue du club.

6. Rendu du Site Web

6.1. Page d’accueil

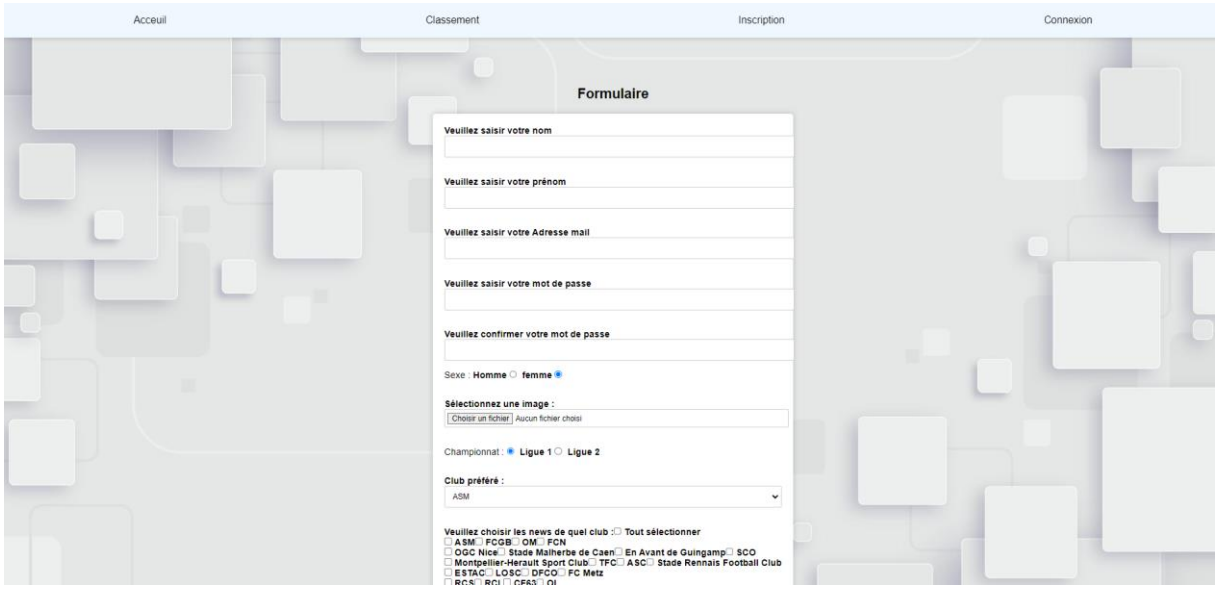


6.2. Page de classement

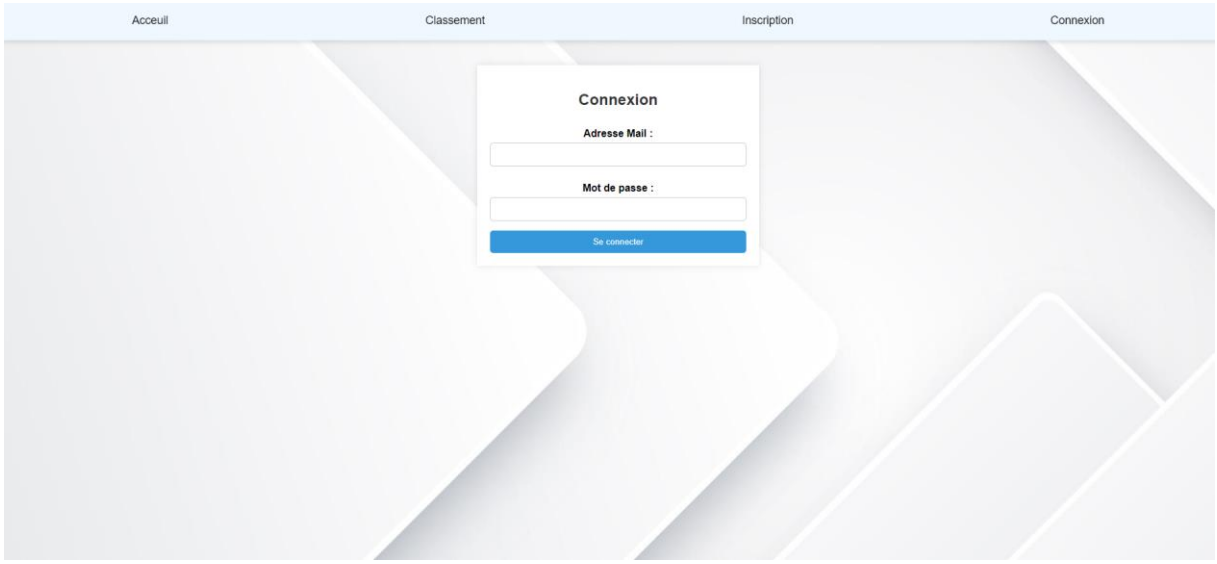


Position	Nom du Club	Matchs Gagnés	Matchs Perdus	Matchs Nuls	Buts Marqués	Buts Encaissés	Différence de Buts	Points
1	FCGB	19	14	0	83	80	3	57
2	TFC	16	9	8	88	81	7	56
3	ESTAC	18	14	1	90	83	7	55
4	OGC Nice	17	12	4	95	74	21	55
5	ASSE	16	12	5	89	81	8	53
6	Montpellier-Herault Sport Club	15	11	7	84	70	14	52
7	FC Metz	16	13	4	81	74	7	52
8	OM	14	12	7	84	72	12	49
9	LOSC	15	15	3	83	68	15	48
10	PSG	14	13	6	92	86	6	48
11	RCL	13	16	4	71	87	-16	43
12	Stade Rennais Football Club	13	16	4	83	83	0	43
13	En Avant de Guingamp	12	15	6	80	84	-4	42

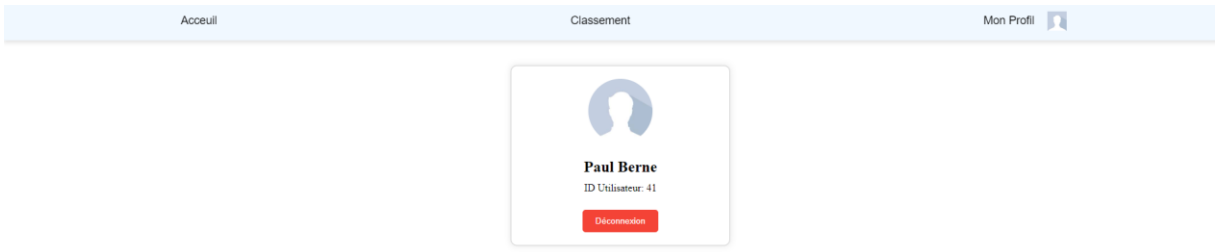
6.3. Page d'inscription



6.4. Page de Connexion



6.5. Page de profil



7. Contact

Si vous souhaitez me contacter :

Adresse électronique : Paulberne@gmail.com

Numéro de téléphone : 07 83 51 97 15

Ou via LinkedIn : https://www.linkedin.com/in/paul-berne/

8. Conclusion

Ce projet m'a permis d'approfondir mes connaissances en termes de Web car j'ai pu exploiter un grande partie des fonctionnalités qui s'offre a moi dans ce domaine.

J'ai pu intéragir avec une base de donnée, mettre en place un modèle MVC et prendre conscience des enjeux de la cybersécurité dans le Web avec l'importance de la place des vérifications dans celui-ci.

La base de données nous été fournis préalablement et en utilisant les préférences de l'utilisateur dans ses choix de clubs préférés.