

---

# Rapport technique Web

## MyManager

12 mai 2020

---

Akarioh Samir, Bougrine Rayan, Karunanayakage Shamal, Lachat Gabin  
Groupe 20

### Sommaire

- I. Présentation du site
- II. Les fichiers
- III. La base de donnée
- IV. Les parties du site
- V. Organisation du travail
- VI. Les problèmes rencontrés et les solutions
- VII. Le futur du site

## I. Présentation du site

Notre site My Manager a pour but de permettre aux utilisateurs de partager leur expérience du jeu Football Manager.

Étant nous 4 fans du jeu, nous voulions que d'autres fans est accès à un site de discussion et de partage. En effet, les utilisateurs pourront discuter entre eux à travers un forum mais aussi partager les différentes tactiques qu'ils utilisent au sein du jeu. Ainsi les autres pourront laisser des avis dans un espace commentaires dédié à chaque tactique mais également lui donner une note sur 5, qui permettra de créer des grades d'utilisateurs basés sur la moyenne des notes reçues.

Les utilisateurs auront également accès à une page pour rechercher les différents joueurs du jeu selon certains critères (nom, niveau ...). La également un espace commentaire et un système de note sera mise en place.

Ainsi le but rechercher par notre équipe est de réunir et fédérer toute la communauté de Football Manager.

---

## II. Les fichiers

Pour visualiser le code php nous avons décidé d'utiliser des serveurs locaux WAMP et MAMP, au lieu d'utiliser un hébergeur web (gratuit ou payant).

Chemin vers les fichiers : /projet/views/

On a ensuite les fichiers comme le montre le schéma suivant:

```
> projet
  |>views
  |  |>Base de donnée
  |  |  | MyM.sql
  |  |>JS
  |  |  | formulaire_validation.js
  |  |>images
  |  |  | toutes les images utilisées sur le site
  |  |>polices
  |  |  | toutes les polices utilisées sur le site
  |  accueil.php
  |  api.php
  |  connexion.php
  |  db_connect.php
  |  deconnexion.php
  |  edit.php
  |  forum_accueil.php
  |  forum_edit.php
  |  forum_newtopic.php
  |  forum_topic.php
  |  functions.php
  |  header.php
  |  inscription.php
  |  joueurs_recherche.php
  |  joueurs.php
  |  pasdeRS.php
  |  profil.php
  |  recherche.php
  |  style_forum.css
  |  style_MyM.css
  |  tact_edit.php
  |  tact.php
  |  tactique_joueurs.php
  |  tactique_newtact.php
  |  tactique.php
  |  topic.php
```

---

### III. La base de donnée

Présentation de la base de donnée :

Nom de la base de donnée : MyM

-composition : regroupe les différentes composition disponible dans le joueurs.

id (integer NOT NULL),  
compo (text)

-espace\_com\_joueur : ici est stocker les commentaires des utilisateur de la page joueurs.

id integer NOT NULL,  
id\_joueur integer,  
id\_posteur integer,  
dateheure\_post timestamp with time zone,  
dateheure\_edition timestamp with time zone,  
contenu text

-espace\_com\_tactique : ici est stocker les commentaires des utilisateur de la page tactiques.

id integer NOT NULL,  
id\_tactique integer,  
id\_posteur integer,  
dateheure\_post timestamp with time zone,  
dateheure\_edition timestamp with time zone,  
contenu text

-f\_categories : ici sont stocker le nom des catégorie du forum.

id integer NOT NULL,  
nom text

-f\_messages: ici sont stocker les commentaires des utilisateur sur le forum.

id integer NOT NULL,  
id\_topic integer,  
id\_posteur integer,  
dateheure\_post timestamp with time zone,  
dateheure\_edition timestamp with time zone,  
meilleur\_reponse integer,  
contenu text

-f\_souscategories : ici sont stocker le nom des sous-catégories du forum.

id integer NOT NULL,  
id\_cate integer,  
nom text

- f\_topics : ici sont stocker les sujets du forum créé par les utilisateur.

id integer NOT NULL,  
id\_createur integer,  
sujet text,  
contenu text,  
dateheure\_creat timestamp with time zone,  
résolu integer

-f\_topicscate : ici on relie les topics aux catégories et aux sous-catégories.

id\_topicscate integer NOT NULL,  
id\_topic integer,  
id\_cate integer,  
id\_souscate integer

-joueurs : ici est stocker les joueurs disponible pour la recherche de joueurs.

---

```
id integer NOT NULL,  
nom text,  
prenom text,  
age integer,  
nation_id integer,  
poste text,  
attaque integer,  
defense integer,  
technique integer,  
note double precision,  
valeur double precision,  
note_utilisateur double precision
```

-membre : ici sont stocker les membres qui s'inscrive sur le forum

```
id integer NOT NULL,  
nom text,  
prenom text,  
pseudo text,  
mail text,  
club text,  
m_version text,  
motdepasse text,  
m_admin integer
```

-nation : ici on a accès aux pays des joueurs.

```
id integer NOT NULL,  
nom text
```

-tactique : ici est stocker toutes les tactique que l'utilisateur crée via la page tactique.

```
id integer NOT NULL,  
nom integer NOT NULL,  
id_membre integer,  
equipe text,  
composition integer,  
commentaire text,  
dateheure_creat timestamp with time zone,  
joueur1 text,  
joueur2 text,  
joueur3 text,  
joueur4 text,  
joueur5 text,  
joueur6 text,  
joueur7 text,  
joueur8 text,  
joueur9 text,  
joueur10 text,  
joueur11 text
```

---

## IV. Les parties du site

- **connexion à la base de donnée** : nous avons décidé de mettre la connexion à la base de donnée dans un fichier à part, `db_connect.php`, afin de ne pas répéter la ligne de code de connexion sur tous les fichiers. On a juste à inclure le fichier `db_connect.php` à chaque fichier, donc on évite ainsi aux clients qui souhaite acquérir de ne pas avoir à modifier tous les fichiers pour qu'ils soient tous connectés à la base de donnée, on a donc un gain de temps et d'argent pour les investisseurs.

On a décidé d'utiliser PDO afin de faciliter le switch entre les types de bases de données, qui nous a servi lorsqu'on a switch entre `phpmyadmin` et `phpgadmin`.

On va d'abord parler des deux fichiers que nous utilisons dans tous les fichiers du site, on les a séparé en fichier à part pour raccourcir la longueur des fichiers.

- **header** : représente la tête de la page c'est à dire la partie qui regroupe le logo, le nom du site, le menu de navigation (Accueil, Tactique, Joueur, Forum, Connexion) et un bouton de déconnexion si on est connecté au site.

On utilise dans ce fichier du php pour les liens de navigations afin de garder l'id d'une personne connecté. On crée une session via `session_start()` afin de mémoriser l'id d'un membre connecté, et propre à chacun. On utilise un `isset($_SESSION['id'])` dans une condition if pour savoir si l'utilisateur qui navigue sur le site est connecté ou pas, ce qui permettra par la suite d'accéder à certaines fonctionnalités réservé au membre.

- **footer** : représente le pied de la page c'est à dire la partie qui contient un lien pour envoyer un mail, et des liens (dans des images) qui vont vers les réseaux sociaux (qui n'existent pas encore, les liens renvoies donc vers une page qui s'excuse de l'absence de réseaux sociaux pour l'instant).

Sur toute les parties du site on retrouve le `session_start()` pour permettre à un utilisateur qui se connecte de rester connecté sur toute les pages du site, tous les fichiers commencent donc par `<?php session_start() ?>` à la toute première ligne.

Les différentes parties du site:

- **accueil** : La page d'accueil n'apporte que des informations et est codée en html.

- **joueur** : cette partie est décomposée en 3 fichiers différents qui sont `joueurs.php`, `joueurs_recherche.php`, `recherche.php`. Cette partie permet de rechercher des joueurs (présents dans la base de données). L'accueil de cette partie se trouve dans le `joueurs.php` on peut trouvé une petite liste de 5 joueurs qui sont les premiers id de la base de données joueurs, une simple requête permet de récupérer cette petite liste puis de les afficher avec un lien redirectionnel vers la page du joueur associé. On a également un système de filtre qui permettrait de rechercher n'importe quel joueur de la bdd en fonction de son nom, de son prénom ou de ses statistiques.

L'utilisation de filtre va fonctionner à l'aide d'une API, tous les filtres sont initiés de façons à ce qu'on affiche tous les joueurs si les filtres ne sont pas remplis. Puis en fonction de la valeur rentrée (dans une zone de texte) ou choisie (dans une liste déroulante) on complétera une requête pour afficher l'ensemble (un ou plusieurs) des joueurs qui remplissent les critères choisis par l'utilisateur. Les critères remplis dans les filtres par l'utilisateur sont récupérés par la méthode POST. La recherche de joueur peut s'effectuer sans connexion au préalable au site.

---

Les résultats des filtres vont s'afficher sur la page codée dans recherche.php, vu que notre base de donnée joueurs est très petite, on affiche les joueurs sur la même page, lorsqu'on aura une base de donnée bien plus étoffée on pourra les répartir sur plusieurs pages (on le fait déjà sur l'affichage des commentaires).

La page joueur\_recherche affiche les données d'un joueur, choisi au préalable. Des requêtes SQL permettent d'afficher les informations qui sont stockées dans la base de donnée à l'aide de l'id du joueur que l'on peut récupérer dans l'URL à l'aide de la méthode GET. Cette page présente également un espace commentaire où un utilisateur connecté peut poster un commentaire que l'on insère dans la table espace\_com\_joueur avec la requête SQL « INSERT INTO ... VALUES ... ».

On affiche ensuite les commentaires déjà présents (plus celui qu'on vient de poster si on vient d'en poster un) par dix, les dix suivants sont affichés sur la page suivante on utilise « LIMIT ... OFFSET ».

- **tactique** : Cette partie est composée de 4 fichiers tactique.php, tactique\_newtact.php, tactique\_joueur.php, tact.php. Elle permet de voir, poster ou commenter une tactique postée par soi ou un autre membre inscrit du site.

L'accueil de cette partie est codée par le fichier tactique.php et liste les différents utilisateurs ayant postés au moins une tactique dans un tableau chaque ligne du tableau présente le pseudo de l'utilisateur et les différentes tactiques postées par cet utilisateur. Si on appuie sur une de ces tactiques on obtient plus d'information sur la tactique avec notamment une photo de la tactique et la liste de joueur à côté qui compose l'équipe. Là aussi chaque tactique possède un id qui lui est propre et qui permet de récupérer les informations depuis la table tactique assez facilement. Cette page aussi admet un espace commentaire mais cette fois ci les commentaires seront stockés dans la table espace\_com\_tactique de la bdd. et fonctionne de façon identique à l'espace commentaire de la partie joueur.

Un membre inscrit peut également poster une nouvelle tactique avec le nom, l'équipe, la composition, les joueurs et le commentaire qu'il souhaite dans la mesure du possible (par exemple il ne peut choisir que des joueurs présents dans la table joueurs, il peut choisir les joueurs grâce à des menus déroulants en fonction du poste dans la composition. La création de la tactique se fait dans les fichiers tactique\_newtact.php et tactique\_joueurs.php, on utilise la méthode GET pour récupérer l'id de la tactique que l'on met dans l'URL.

- **forum** : Cette partie est composée de 6 fichiers forum\_accueil.php, forum\_edit.php, forum\_newtopic.php, forum\_topic.php, topic.php et edit.php. Elle permet de réunir et fédérer la communauté de Football Manager. Tous les utilisateurs inscrits ou non inscrits peuvent lire les topics et les réponses associées, mais seul un membre connecté peut répondre à un topic et en créer un nouveau.

L'accueil se trouve dans forum\_accueil.php et liste les catégories avec ses sous-catégories. Si on appuie sur une catégorie alors on listera l'ensemble des topics associés à cette catégorie, et si on clique sur une sous-catégorie on listera l'ensemble des topics liés à la catégorie et au sous-catégorie sélectionnée, par exemple si on clique sur Discussion générale de Football Manager 2020 alors on listera tous les topics qui ont pour id\_cate égale à l'id de la catégorie, et id\_sscate a id de la sous-catégorie, on utilise pour cela la table f\_topicscate, qui réunit à chaque topic, l'id de la catégorie et de la sous-catégorie associés. En cliquant sur une catégorie ou une sous-catégorie on nous envoie sur la page forum\_topic.php. En cliquant sur un des sujets de topics on accèdera à la page topic.php où on aura toutes les informations sur le topic et un espace où les utilisateurs connectés peuvent répondre au topic et qui fonctionne de la même manière que les espaces commentaires de tactique et joueur mais qui insère dans la table f\_messages on peut également éditer le sujet ou le contenu du topic si le \$\_SESSION['id'] est égal à l'id du créateur du topic, on utilise pour cela un « UPDATE ... SET ... WHERE ».

---

Pour le moment il est impossible de créer une catégorie ou une sous-catégorie même pour un administrateur, mais on pourra l'implémenter assez facilement si jamais cela sera demandé par les investisseurs.

Un utilisateur connecté peut également créer un nouveau topic. En choisissant la catégorie la sous catégorie le sujet et le message. On insert donc f\_topics mais aussi f\_topicscate lors de la création. Il faut également savoir que seul un Administrateur peut poster un topic dans la catégorie Annonces.

- **Inscription, Connexion et Profil** : Cette partie est composée de 2 fichiers connexion.php, inscription.php. L'inscription sert à créer un membre dans la base de donnée, lors de l'inscription on ne complète que le pseudo, le mail, un deuxième mail (mail de confirmation), un mot de passe et un deuxième mot de passe (mot de passe de confirmation). Ainsi dans un premier, on vérifie si tous les champs sont remplis, si le pseudo ne dépasse 50 caractères, si les mails sont identiques et si les mots de passe sont identiques. Le formulaire pour l'inscription est validé par du JavaScript. On insert, lorsque tout est conforme aux attentes, les informations du formulaires dans la table membre, les autres colonnes de la table sont NULL, sauf m\_admin qui prends la valeur 0 (1 signifie qu'on est administrateur, 0 non). Pour la connexion on a besoin que du pseudo et du mot de passe, on vérifie ainsi si le pseudo existe dans la base de donnée membre avec un « WHERE pseudo=? » Avec le pseudo récupérer du formulaire et avec un « ->rowCount() » on compte le nombre de lignes que renvoie notre requête si il est différent de 0 alors le pseudo existe et on récupère son mot de passe de la base de donnée et on le compare avec celui rentré dans le formulaire si les deux correspondent alors la connexion est réussi sinon on redemande pseudo et mot de passe.

Une fois la connexion réussi on arrive sur la page profil, ou on peut voir les informations de son compte que l'on peut modifier en remplissant un formulaire. Sur cette page un utilisateur peut voir son activité sur le site (avec une série de requête) nombre de commentaires postés, de topics ou de tactiques postés.

## V. Organisation du travail

Nous avons utiliser un Drive pour se partager les fichiers, afin que tous puissent avoir accès aux dernières version du site. Nous avons également utiliser serveur discord pour faire des réunions, environ tous les 3 jours pour discuter de l'avancement du projet, et de la suite du projet sachant que chacun avait une page attribuée à avancer lors de chaque réunion.

## VI. Les problèmes rencontrés et les solutions

Le plus grand problème qu'on a eu fut sur les requêtes sql : on a vu qu'il fallait utiliser le pgsql , il a donc fallu que nous fassions des manip afin de passer en pgsql. D'abord on a du recréer une base de donnée sur phpadmin, on a pas réussi à importer la base de donnée exporté depuis phpmyadmin.

Mais le fait qu'on ai utilisé PDO a grandement facilité les modifications car il a seulement fallut se connecter à la base de donnée du phpadmin dans le fichier db\_connect.php :

```
$bdd = new PDO("pgsql:host=localhost; dbname=MyM; port=5432; user=postgres; password=root");
```

Il a quand même fallut changer plusieurs requêtes, et notamment gérer la différence entre le simple et le double guillemets.

On a également le problème du JavaScript et son fonctionnement pour la validation du formulaire où il fallait jongler entre le php et le JavaScript. L'utilisation du json et du XMLHttpRequest (vu sur le web).

---

Pour le forum, nous avons dans un premier temps souhaité télécharger et utiliser une template déjà existante sur mybb mais nous avons décidé en équipe de refaire un forum.

Un autre problème fut la récupération de donnée des formulaires, surtout sur la compréhension de comment il fallait faire.

On a également sur la longueur des fichiers on a donc décidé de mettre le header et le footer dans des fichiers header.php et footer.php .

On aurait également du mettre les codes php dans un fichier séparé pour diminuer la longueur des fichiers.

## VI. Le futur du site

Comme vous l'avez vu sur la page tactique les joueurs affichés en dessous de l'image de la tactique, le futur de cette sera d'afficher les photos des joueurs sur la photo. Et modifier les listes déroulante par un système de drag and drop. Une image dynamique est le but recherché.

Nous allons modifier notre code afin de faire fonctionner le système de notation communautaire via des étoiles allant de 0 à 5 pour noter les tactiques des utilisateurs mais aussi les joueurs disponibles dans la base de données.

Nous allons aussi créer un système de grade sur le forum afin de récompenser nos plus fervents supporters par exemple un grade si on a une moyenne de note de plus 4 étoiles sur les tactiques postées. Cela permettra une meilleure entraide sur le forum.