



PROJET LIGUE 1

PHP application web



23 MAI 2023

Documentation projet Ligue 1

Table des matières

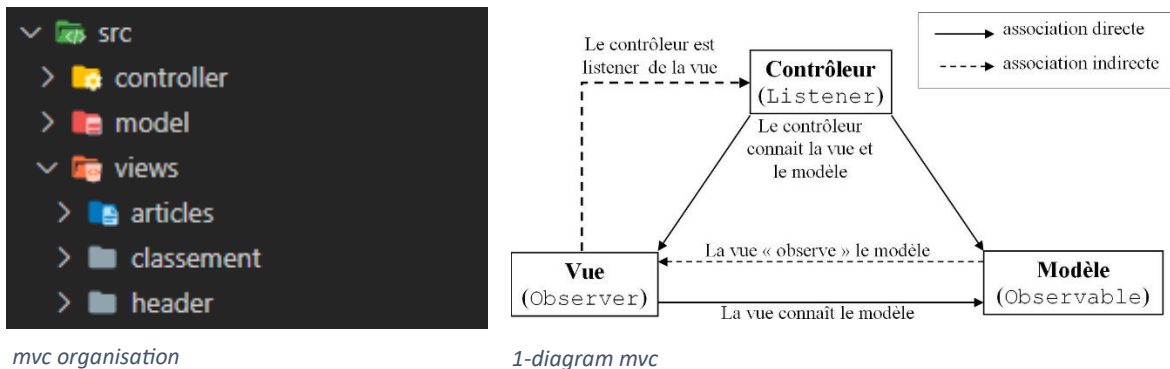
Documentation projet Ligue 1.....	0
Introduction.....	1
Architecture.....	2
Routeur.....	2
Base de données	4
Cas d'utilisations.....	6
Connexion.....	6
Inscription.....	6
Consulter classement :	6
Consulter liste articles :	7
Commenter :.....	7
Créer un article :	7
Consulter Profil :	8
Changement Email / Mot de passe / Suppression de compte :	8
Gestion utilisateurs :.....	9

Introduction

Le projet consiste en la création d'un site web dédié à la Ligue 1, le championnat de football français de première division. Le site permettra aux utilisateurs de consulter le classement des équipes, de lire et de poster des articles, de commenter les articles existants, ainsi que de se connecter et de créer un compte utilisateur.

Architecture

Le modèle architectural choisi est le Modèle-Vue-Contrôleur il est couramment utilisé pour développer des applications web. Il divise l'application en trois composants principaux : le modèle, la vue et le contrôleur. Chaque composant a un rôle spécifique dans la gestion des données, de la logique métier et de l'interface utilisateur.



- Le modèle représente la couche de données de l'application. Il est responsable de la gestion des données, de l'accès à la base de données et de la logique métier.
 - La vue représente la couche d'interface utilisateur de l'application. Elle est responsable de l'affichage des données et de l'interaction avec l'utilisateur.
 - Le contrôleur agit comme un intermédiaire entre le modèle et la vue. Il reçoit les requêtes de l'utilisateur, traite les données et détermine quelle vue afficher.
- Routeur

Routeur

Un routeur est un concept fondamental dans le développement web, notamment dans les architectures basées sur le modèle Modèle-Vue-Contrôleur (MVC) ou d'autres modèles similaires. Un routeur est responsable de la gestion des routes de l'application, c'est-à-dire de déterminer quelle action doit être exécutée en fonction de l'URL demandée par l'utilisateur.

ID	NOM	PRENOM	MAIL	ADMIN
1	admin	admin	admin	Administrateur
2	Rodriguez	Thomas	thomas.rodriguez38530@gmail.com	User
3	Rodriguez	Thomasz	thomas.rodriguez38530@gmail.com	User
4	RodriguezWWW	ThomasXX	thomas.rodriguez38530@gmail.com	User
5	Rodriguezaaaa	Thomasaaa	thomas.aarodriguez38530@gmail.com	User
6	Rodriguezqqq	Thomasqqq	thomasqq.rodriguez38530@gmail.com	User

2-route exemple

Lorsqu'un utilisateur accède à une URL spécifique dans un site web, le router analyse cette URL et détermine quelles actions ou quelles ressources doivent être associées à cette demande. Il agit comme une sorte de tableau de bord qui dirige la requête HTTP vers le bon contrôleur et la bonne méthode pour traiter cette requête.

```
$request_uri = $_SERVER['REQUEST_URI'];

if (isset($request_uri)) {
    $route = preg_split('/', $request_uri);
    handleroute($route);
}
```

3- handle request

```
function handleroute($route) {
    switch ($route[1]) {
        case '':
            require __DIR__ . '/views/menu.php';
            break;
        case 'login':
            if(isset($route[2])) to404();
            require __DIR__ . '/controller/c_login.php';
            break;
        case 'register':
            if(isset($route[2])) to404();
            require __DIR__ . '/controller/c_register.php';
            break;
    }
}
```

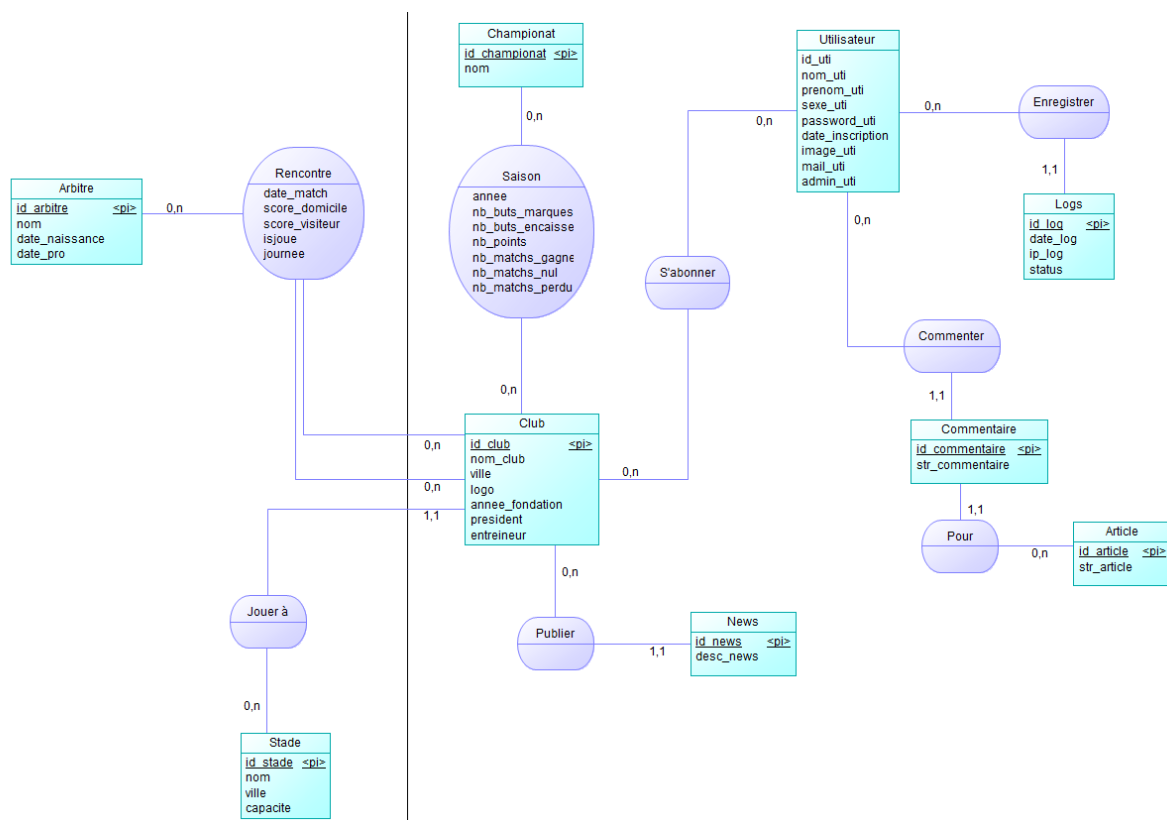
4-handle routes

Base de données

L'utilisation du moteur PostgreSQL pour la base de données de votre site web présente plusieurs avantages et offre des fonctionnalités bénéfiques pour un projet.

PostgreSQL est optimisé pour offrir de bonnes performances, même avec des charges de travail intensives. Il utilise des techniques d'indexation avancées, des optimisations de requêtes et des mécanismes de mise en cache pour améliorer les performances des requêtes et des opérations de lecture/écriture.

Voici le modèle conceptuel de données de notre base :



5-MCD

Une fonction stockée est un ensemble de commandes SQL et de logique métier qui est pré-compilé et stocké dans la base de données. Elle peut être appelée et exécutée ultérieurement en utilisant un nom et éventuellement des paramètres. Les fonctions stockées sont généralement écrites dans un langage procédural spécifique à la base de données, tel que PL/pgSQL pour PostgreSQL.

The screenshot shows the 'Properties' window for a PostgreSQL function named 'classement_saison'. The 'Procedure Name' is 'classement_saison' and the 'Type' is 'Function'. The 'Object ID' is '24578'. The 'Source' tab is selected, displaying the following SQL code:

```

CREATE OR REPLACE FUNCTION public.classement_saison(anneeselect integer, ligueselect integer)
RETURNS TABLE(id_championnat integer, id_club integer, nom_club character varying, nb_points integer, nb_buts_ma
LANGUAGE plpgsql
AS $function$
BEGIN
    RETURN QUERY
        SELECT saison.id_championnat, club.id_club, club.nom_club, saison.nb_points, saison.nb_buts_marques, sais
        (select * from get_nb_match_joue(club.id_club, anneeSelect)) as nb_match_joue,
        saison.nb_matches_gagne as nb_match_gagne,
        saison.nb_matches_nul as nb_match_nul,
        saison.nb_matches_perdu as nb_match_perdu,
        club.logo
        FROM club
        INNER JOIN saison ON club.id_club = saison.id_club
        WHERE saison.annee = anneeSelect AND saison.id_championnat = ligueSelect
        ORDER BY saison.nb_points DESC;
END;
$function$
;

```

6-fonction stockée code

Les fonctions stockées sont compilées et optimisées par le système de gestion de base de données, ce qui peut améliorer les performances lors de leur exécution. Elles réduisent également le trafic réseau, car seule la procédure stockée et les résultats sont transmis entre l'application et la base de données, plutôt que des requêtes SQL complexes.

The screenshot shows a list of functions in the database explorer. The functions listed are:

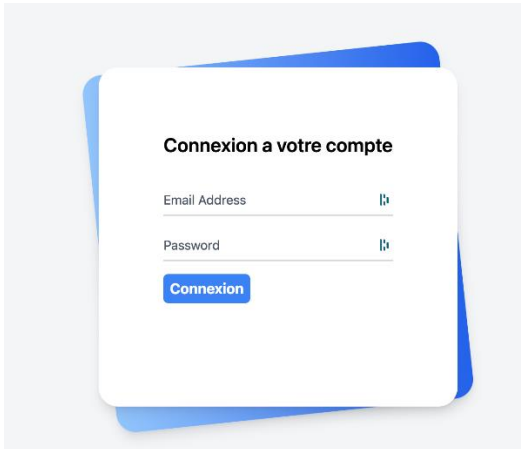
- change()
- classement_saison(int4, int4)
- getAnneeSaison(date)
- get_info_club(int4)
- get_matches_journee(int4, int4, int4)
- get_nb_buts_contre(int4)
- get_nb_buts_pour(int4)
- get_nb_match_gagne(int4)
- get_nb_match_joue(int4, int4)
- get_nb_match_nul(int4)
- get_nb_match_perdu(int4)

7-fonction stockée liste

Les fonctions stockées permettent de regrouper des blocs de code SQL complexes et répétitifs en une seule entité. Cela facilite la réutilisation du code, car la fonction peut être appelée à partir de différentes parties de votre application ou d'autres fonctions stockées.

Cas d'utilisations

Connexion :



Connexion a votre compte

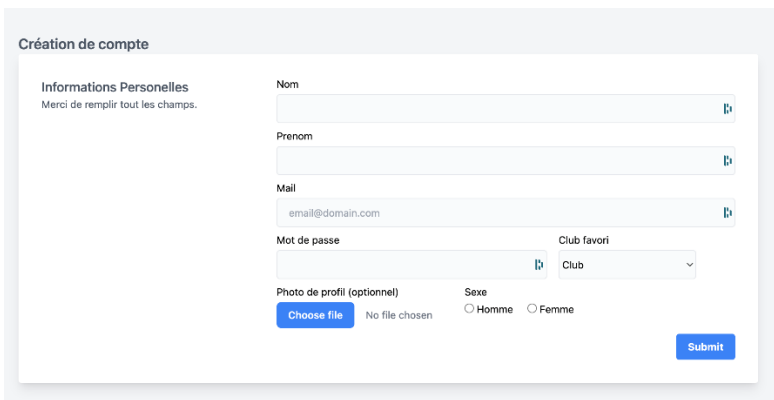
Email Address

Password

Connexion

8-login

Inscription :



Création de compte

Informations Personnelles
Merci de remplir tout les champs.

Nom

Prenom

Mail

Mot de passe Club favori


Photo de profil (optionnel) No file chosen

Sexe ☐ Homme ☐ Femme

Submit

9-register


Consulter classement :

Ligue 1

[Home](#)[Classement](#)[Articles](#)

[Login](#)[Register](#)

Classement saison




Red Star FC

Rank: 1

nb points: 88

But marqués: 60

But encaissés: 35




Paris Saint-Germain

Rank: 2

nb points: 67

But marqués: 75

But encaissés: 30



Olympique de Marseille

Rank: 3


nb points: 78

But marqués: 62

But encaissés: 36

10-classement

Consulter liste articles :

 **Ligue 1**

[Home](#) [Classement](#) [Articles](#) [Login](#) [Register](#)

Liste d'articles [Ecrire un article](#)

**super**
ceci est un article
Auteur: admin

**super**
ceci est un article
Auteur: admin

**oui**
non
Auteur: admin







11-liste articles

Commenter :

super
ceci est un article
[Supprimer](#)

Write a comment...
[Post comment](#)

Commentaires

Rodriguez
test

admin
tropbien

12-commentaire article

Créer un article :

Rédaction d'un article

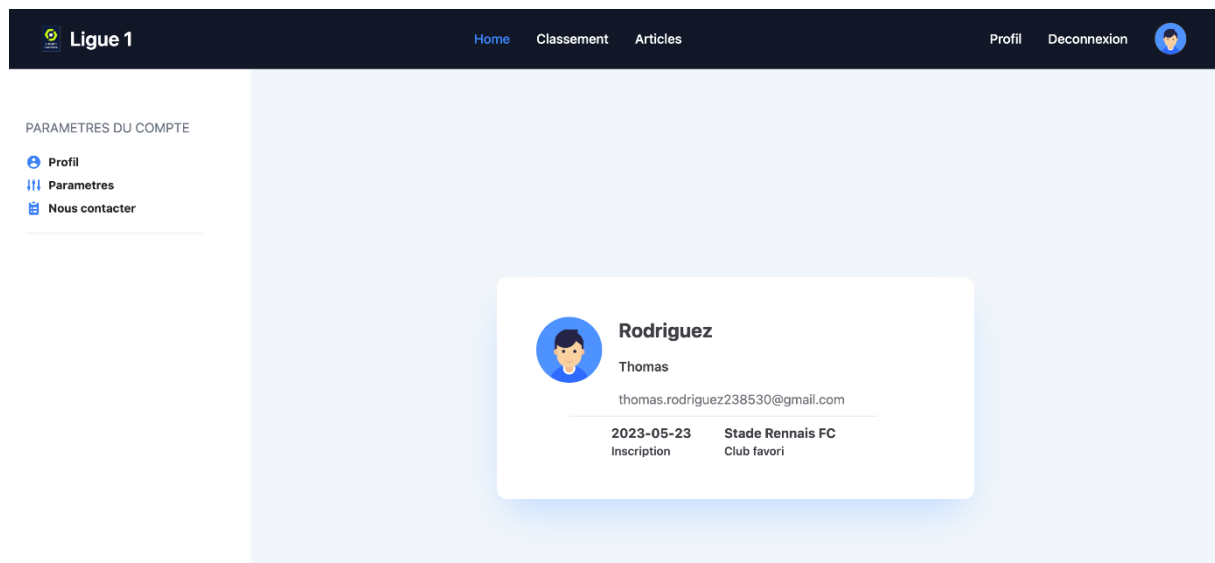
Titre

Contenu

Image (optionnel)
[Choose file](#) No file chosen
[Submit](#)

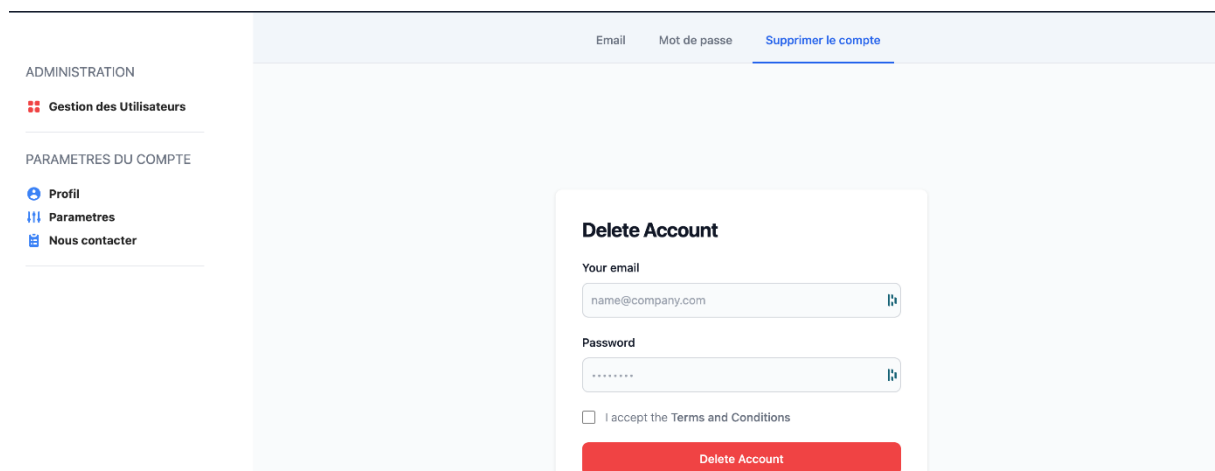
13-redaction article

Consulter Profil :



14-profil

Changement Email / Mot de passe / Suppression de compte :



15-changement info uti

Gestion utilisateurs :

Gestion des utilisateurs				
ID	NOM	PRENOM	MAIL	ADMIN
1	admin	admin	admin	Administrateur
2	Rodriguez	Thomas	thomas.rodriguez38530@gmail.com	User
3	Rodriguezz	Thomasz	thomas.rodriguezz38530@gmail.com	User
4	RodriguezWWW	ThomasXX	thomas.rodriguxxez38530@gmail.com	User
5	Rodriguezaaaa	Thomasaaa	thomas.aaarodriguez38530@gmail.com	User

16-gestion uti

1-diagram mvc	2
2-route exemple	3
3- handle request	3
4-handle routes	3
5-MCD.....	4
6-fonction stockée code.....	5
7-fonction stockée liste.....	5
8-login.....	6
9-register	6
10-classement	6
11-liste articles	7
12-commentaire article	7
13-redaction article	7
14-profil	8
15-changement info uti	8
16-gestion uti.....	9