



# Wolf Fitness

Crée par Lecomte Paul

le 03.05.2024

Dernière modification le 08.07.2024



## Introduction

### **Objectif du document**

Ce document vise à fournir une documentation technique et graphique complète du projet. Il va détailler la structure, les fonctionnalités, l'architecture ainsi que les difficultés rencontrées lors du développement. Cette documentation servira de guide pour les développeurs, les utilisateurs et toute autre personne intéressée par le projet.

### **Source du projet**

Pour ce projet, j'ai décidé de prendre une idée que j'avais eue durant le premier projet de l'année : Wolf Fitness. Étant un sportif moi-même, ce projet me tenait beaucoup à cœur. Ce module PHP m'a permis de créer un site web fonctionnel et interactif.

### **Description est besoins du projet**

Wolf Fitness est une application web de média social pour les passionnés de fitness, inspirée de Twitter. Le besoin principal est de créer une plateforme où les utilisateurs peuvent partager leurs progrès, planifier des entraînements et interagir avec d'autres passionnés de fitness.

### **Fonctionnalités du projet**

- Inscription et connexion des utilisateurs
- Gestion et modification du profil utilisateur
- Publication et modification de posts
- Création et utilisation d'entraînements
- Création de commentaires sur les posts
- Mettre des likes sur les posts
- Affichage des posts et des likes sur le profil



## Outils utilisé

### Frontend

- HTML
- CSS
- JavaScript
- Bulma framework

### Backend

- PHP

### Animation

- GSAP

### Base de données

- MySQL

## Bulma

Bulma est un framework CSS moderne et open-source basé sur Flexbox, conçu pour créer facilement des interfaces utilisateur réactives et élégantes. Il propose une grille flexible, des typographies intuitives, et une gamme de composants pré-stylisés tels que des navbars et des cartes. Bulma est entièrement modulaire, permettant d'inclure uniquement les parties nécessaires, et est facilement personnalisable via des variables Sass. Sa syntaxe simple et son adoption rapide en font un choix idéal pour les développeurs souhaitant accélérer le développement front-end. Pour plus d'informations, consultez [bulma.io](https://bulma.io).



## GSAP

GSAP (GreenSock Animation Platform) est un framework JavaScript puissant et flexible pour créer des animations web performantes. Il permet de manipuler le DOM de manière fluide et efficace, avec une syntaxe simple et intuitive pour des animations complexes. Compatible avec la plupart des navigateurs modernes et intégrable avec des frameworks comme React et Vue, GSAP offre des outils comme les timelines et les courbes de Bézier, ainsi que des extensions comme ScrollTrigger pour des animations déclenchées par le défilement, pour en s'avoir plus vous pouvez aller sur [GSAP](https://greensock.com/gsap/)

## Fonctionnement général de l'application

Pour avoir toutes les interactions de l'application, l'utilisateur devra se connecter ou créer un compte. S'il choisit de créer un compte, il devra entrer un nom d'utilisateur, un email, un mot de passe et une image de profil. Une fois toutes ces informations saisies et vérifiées, il sera redirigé vers feed.php, la page principale, et aura accès à :

- Regarder des posts
- Liker des posts
- Commenter des posts
- Faire une recherche de posts
- Créer des posts
- Aller sur le profil pour voir tous les posts créés ou likés, ainsi que modifier son profil
- Aller sur l'onglet "Training" pour pouvoir créer des entraînements, les utiliser ou les modifier

Voici en général les fonctionnalités disponibles sur le site. Je prévois d'ajouter plus de fonctionnalités dans les mois à venir pour une mise en ligne très bientôt.



## Organigramme

Cet organigramme représente la structure des pages et des fonctionnalités du site, permettant de visualiser le parcours de l'utilisateur depuis ses premiers pas sur le site.



## Architecture

### Architecture procédural

L'application a été construite sur le modèle procédural, ce qui signifie que tout le code PHP est directement dans la page, comme vous pouvez le voir dans l'exemple ci-dessous.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Cours PHP / MySQL</title>
5     <meta charset="utf-8">
6     <link rel="stylesheet" href="cours.css">
7   </head>
8   <body>
9     <h1>Bases de données MySQL</h1>
10    <?php
11      $servername = 'localhost';
12      $username = 'root';
13      $password = 'root';
14
15      //On établit la connexion
16      $conn = new mysqli($servername, $username, $password);
17
18      //On vérifie la connexion
19      if($conn->connect_error){
20        die('Erreur : ' . $conn->connect_error);
21      }
22      echo 'Connexion réussie';
23    ?>
24  </body>
25 </html>
```

Ce modèle a été choisi pour sa simplicité, au prix d'un code un peu plus long et parfois un peu plus brouillon par rapport au modèle MVC, qui demande une structure beaucoup plus minutieuse mais est plus propre.

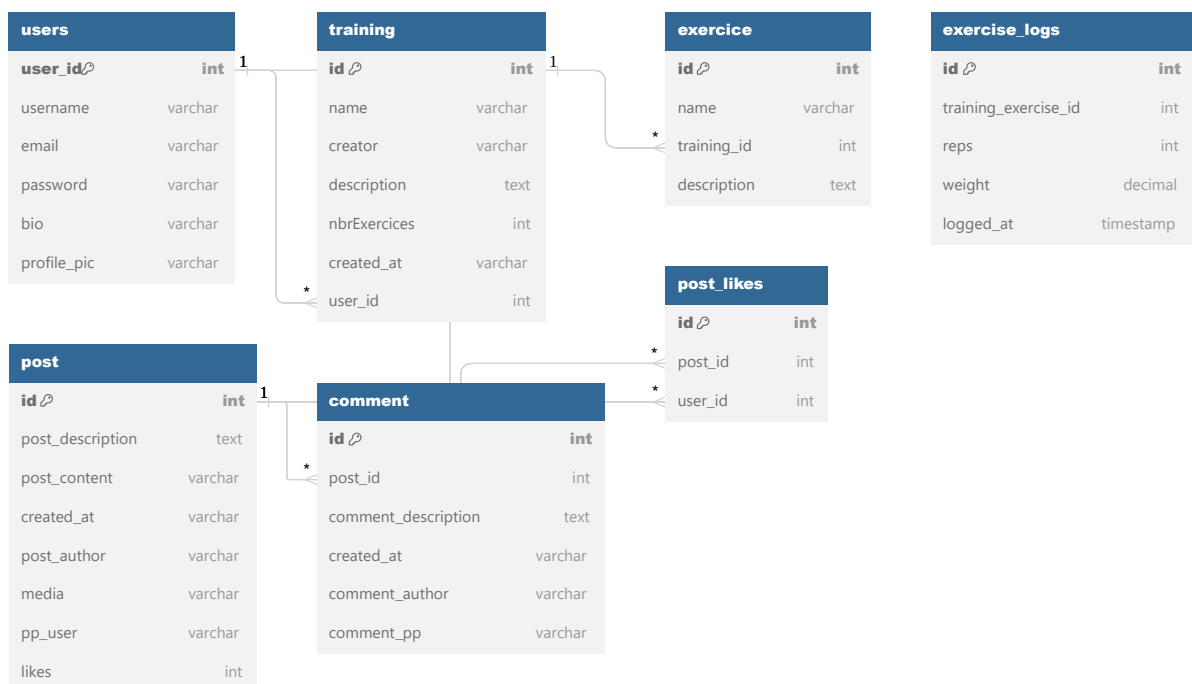


## Dossier est fichier

- Frontend
  - assets/ : Contient les images est le logo
  - components/ : Dossier pour les fichiers réutilisable
  - views/ : Toutes les pages avec un visuel
  - stylesheet/ : Fichier CSS
  - js/ : Fichier JavaScript
- backend
  - database/ : Contient les backup de la base de donnés
  - uploads/ : Dossier pour les images uploadées

## Base de donn  

Voici le diagramme de la base de donn  es avec les relations de chaque table. Cette base de donn  es a   t   cr    e avec MySQL.



## Difficultés rencontrées est solutions

Durant le développement de l'application, je suis bien évidemment tombé sur des difficultés et des problèmes que j'ai dû résoudre pour avancer et proposer un produit fonctionnel. Ci-dessous, j'ai choisi quelques problèmes que j'ai rencontrés avec leurs solutions.

### **1. Problème :**

Comment enregistrer des likes, faire en sorte qu'un utilisateur connecté puisse liker une fois un post et que s'il reluke, cela enlève son like, et qu'un utilisateur non connecté puisse voir ces likes.

### **Solution :**

J'ai dû créer une table dans ma base de données qui récupère l'id du post et l'id de l'utilisateur. Par la suite, quand on clique sur like, cela fait une requête à la base post\_likes pour voir si l'utilisateur a déjà liké. Si oui, le like est supprimé, sinon il est enregistré et ensuite incrémenté dans le post pour afficher le nombre de likes.

### **2. Problème :**

Comment faire en sorte que l'utilisateur puisse ajouter un entraînement dans un post.

### **Solution :**

La solution fut de récupérer l'id de l'entraînement quand on clique sur ajouter, puis de pousser tout cela avec les informations de l'utilisateur pour ensuite pouvoir l'afficher dans les entraînements.





## Futur du projet et améliorations

La version présentée à ce jour n'est qu'un prototype et donc pas une version finale. L'idée principale du projet s'y trouve mais il y a encore beaucoup d'améliorations et de fonctionnalités à ajouter, ce qui se fera au cours des années suivantes. Ci-dessous une liste des fonctionnalités que je rajouterai prochainement :

1. Chronomètre pour les temps de repos entre chaque série
2. Refaire tout l'aspect graphique pour que ce soit plus plaisant à regarder
3. Faire une version mobile
4. Un système de messages privés entre utilisateurs
5. Ajouter un système pour suivre d'autres utilisateurs



## Remerciements

Ce projet a été vraiment super à développer. En plus d'être un défi, il m'a permis de me dépasser et d'apprendre plein de nouvelles techniques et technologies qui me serviront dans mes prochains projets.

Je remercie mes professeurs à la SAE de Genève pour leurs enseignements et leur disponibilité.

Je remercie aussi l'équipe derrière le framework Bulma qui m'a permis de gagner du temps sur le côté frontend du projet.

Merci pour le temps passé à lire ce document et à tester/juger mon travail.

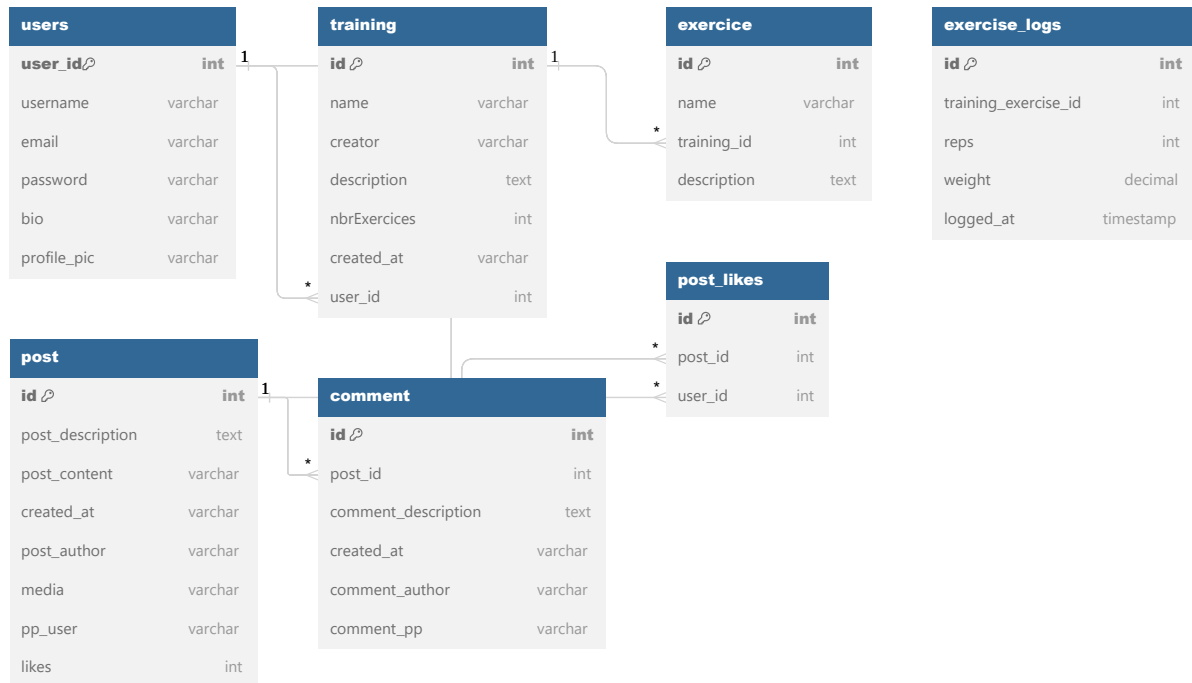


## Annexes

### 1. Organigramme



## 2. Schéma de la base de données



3. Site map

