



# Gestion de la Salle de Sport

# Introduction

Ce document fournit des explications sur la gestion dynamique des données pour un projet utilisant un formulaire HTML et un backend en PHP avec une base de données MySQL. L'objectif est de permettre l'ajout de nouvelles données via un formulaire et leur affichage dynamique.



# 1. Structure Générale

## 1.1 technologie utilisé

- **Frontend** : HTML, TailwindCSS pour le style.
- **Backend** : PHP pour le traitement des données.
- **Base de Données** : MySQL pour le stockage et la gestion des données.



## 1.2. Navigation

- **Formulaire** : Permet l'ajout de nouvelles données.
- **Tableau Dynamique** : Affiche les données stockées.

## 2. Gestion des Données

Ajout de Données avec un Formulaire

### Structure du Formulaire HTML

Ce formulaire permet de collecter les informations à ajouter :

```
<form class="max-w-xl mx-auto" action="inscription.php" method="post">
  <div class="mb-4">
    <input type="text" name="nom" placeholder="Nom" class="w-full p-3" required>
  </div>
  <div class="mb-4">
    <input type="text" name="prenom" placeholder="Prénom" class="w-full focus:ring focus:ring-[#ff129cbf] p-3" required>
  </div>
  <div class="mb-4">
    <input type="email" name="email" placeholder="Email" class="w-full focus:ring focus:ring-[#ff129cbf] p-3" required>
  </div>
  <div class="mb-4">
    <input type="tel" name="telephone" placeholder="N° de téléphone" pattern="[0-9]*" class="w-full focus:ring focus:ring-[#ff129cbf] p-3" required>
  </div>
  <div class="mb-4">
    <select name="activite_id" class="w-full focus:ring focus:ring-[#ff129cbf] p-3 transition" required>
      <option value="">Veuillez choisir une activité</option>
      <?php foreach ($activites as $activite): ?>
        <option value="<?= htmlspecialchars($activite['id_activitée']) ?>"><?= htmlspecialchars($activite['Nom_activitée']) ?></option>
      <?php endforeach; ?>
    </select>
  </div>
  <button type="submit" class="bg-[#ff129cbf] hover:opacity-[0.8] text-white font-bold py-3 px-[16rem] transition ">Réserver</button>
</form>
```



# Traitement des Données en PHP

Le fichier ajouter.php gère l'insertion des données dans la base de donnée

```
<?php
include 'db_connection.php';

try {
    $stmt = $pdo->query("SELECT * FROM activités");
    $activites = $stmt->fetchAll(PDO::FETCH_ASSOC);
} catch (PDOException $e) {
    echo "Erreur lors de la récupération des activités : " . $e->getMessage();
    $activites = [];
}

?>
```

# Affichage dans un Tableau

Les données sont affichées dans un tableau HTML stylisé avec TailwindCSS :

```
<div class="container mx-auto p-6">
  <h1 class="text-2xl font-bold mb-4 text-center">Liste des Activités</h1>
  <div class="overflow-x-auto">
    <table class="min-w-full border border-gray-200 bg-white rounded-lg shadow-md">
      <thead>
        <tr class="bg-purple-900 text-white text-left">
          <th class="px-4 py-2">Id Activité</th>
          <th class="px-4 py-2">Nom Activité</th>
          <th class="px-4 py-2">Description</th>
          <th class="px-4 py-2">Capacité</th>
          <th class="px-4 py-2">Date Début</th>
          <th class="px-4 py-2">Date Fin</th>
          <th class="px-4 py-2">Disponibilité</th>
        </tr>
      </thead>
      <tbody class="divide-y divide-gray-200">
        <?php foreach ($activites as $activite): ?>
          <tr>
            <td class="px-4 py-2 text-gray-700"><?= htmlspecialchars($activite['id_activitée']) ?></td>
            <td class="px-4 py-2 text-gray-700"><?= htmlspecialchars($activite['Nom_activitée']) ?></td>
            <td class="px-4 py-2 text-gray-500"><?= htmlspecialchars($activite['Description']) ?></td>
            <td class="px-4 py-2 text-gray-500"><?= htmlspecialchars($activite['capacité']) ?></td>
            <td class="px-4 py-2 text-gray-500"><?= htmlspecialchars($activite['date_debut']) ?></td>
            <td class="px-4 py-2 text-gray-500"><?= htmlspecialchars($activite['date_fine']) ?></td>
            <td class="px-4 py-2 text-gray-500"><?= htmlspecialchars($activite['disponobilité']) ?></td>
          </tr>
        <?php endforeach; ?>
      </tbody>
    </table>
  </div>
</div>
```



Cette page PHP permet d'ajouter un nouveau membre à la base de données. Les informations sont transmises via un formulaire HTML utilisant la méthode POST. Le script traite les données et les insère dans une table SQL nommée membre.

```
<?php
include 'db_connection.php';

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    try {
        $stmt = $pdo->prepare("INSERT INTO membre (Nom, Prenom, Email, telephone) VALUES (?, ?, ?, ?)");

        $stmt->execute([
            $_POST['nom'],
            $_POST['prenom'],
            $_POST['email'],
            $_POST['telephone'],
            // $_POST['activite_id']
        ]);

        header("Location: index.php?success=1");
        exit();
    } catch (PDOException $e) {
        header("Location: index.php?error=1");
        exit();
    }
}
```

Cette page PHP récupère et affiche dynamiquement les membres enregistrés dans une base de données, offrant une vue conviviale sous forme de tableau. En haut, un menu de navigation stylisé guide l'utilisateur, tandis que le contenu principal présente une table responsive des membres. Les membres sont récupérés via une requête SQL sur la table membre, et chaque résultat est affiché dans une ligne de tableau, grâce à une boucle PHP

```
<div class="overflow-x-auto p-4 bg-white rounded-lg shadow-md mt-[4rem]">
  <table class="table-auto w-full border-collapse border border-gray-200">
    <thead class="bg-purple-700 text-white">
      <tr> ...
    </tr>
    </thead>
    <tbody>
      <?php foreach ($membres as $membre): ?>
        <tr class="odd:bg-purple-50 even:bg-purple-100 hover:bg-purple-200">
          <td class="px-4 py-2 border border-gray-300"><?= htmlspecialchars($membre['id_membre']) ?></td>
          <td class="px-4 py-2 border border-gray-300"><?= htmlspecialchars($membre['Nom']) ?></td>
          <td class="px-4 py-2 border border-gray-300"><?= htmlspecialchars($membre['Prenom']) ?></td>
          <td class="px-4 py-2 border border-gray-300"><?= htmlspecialchars($membre['Email']) ?></td>
          <td class="px-4 py-2 border border-gray-300"><?= htmlspecialchars($membre['telephone']) ?></td>
        </tr>
      <?php endforeach; ?>
    </tbody>
  </table>
</div>
```






```
<?php
$host = 'localhost';
$dbname = 'Salle_sport';
$username = 'root';
$password = '12345';

try {
    $pdo = new PDO("mysql:host=$host;dbname=$dbname", $username, $password);
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
} catch (PDOException $e) {
    echo "Erreur de connexion : " . $e->getMessage();
}
?>
```



code PHP permet de configurer une connexion à une base de données MySQL pour une application de gestion d'une salle de sport. Voici une explication concise :

- 
- **Variables de configuration** : Les détails de la base de données, comme le nom d'hôte (localhost), le nom de la base de données (Salle\_sport), l'utilisateur (root), et le mot de passe (12345), sont définis pour établir la connexion.
  - **Connexion PDO** : Une instance de l'objet PDO est créée en utilisant les paramètres précédents. Cette méthode fournit une interface sécurisée et robuste pour interagir avec la base de données.
  - **Gestion des erreurs** : L'attribut PDO::ATTR\_ERRMODE est réglé sur PDO::ERRMODE\_EXCEPTION pour que les erreurs soient signalées sous forme d'exceptions, ce qui facilite le débogage.
  - **Bloc try-catch** : Si une erreur survient lors de la tentative de connexion (par exemple, un mot de passe incorrect), un message explicite est affiché via catch, informant l'utilisateur de l'échec.



# Conclusion

Cette application offre une gestion efficace des données dynamiques. Elle peut être améliorée avec des fonctionnalités supplémentaires pour une meilleure expérience utilisateur.