



Rapport de Projet

Réalisé par : NMISSI Nadia

Encadré par : OUCHAD Houssni



1. Contexte et Objectifs du Projet

Le projet vise à développer une application de gestion d'activités, de membres et de réservations. Cette application permettra à l'administration de créer, afficher et supprimer des activités, des membres et des réservations. Les membres pourront consulter les activités et effectuer des réservations.

Objectifs Généraux

- Mettre en place une interface intuitive pour la gestion des activités.
- Faciliter la gestion des membres et de leurs informations.
- Offrir une gestion simple et efficace des réservations d'activités.

Objectifs Spécifiques

- Créer un système d'affichage et de suppression des activités.
- Gérer les membres : ajout, affichage et suppression.
- Gérer les réservations : affichage et suppression.
- Assurer la sécurité des données et prévenir les erreurs utilisateur.

2. Périmètre Fonctionnel

2.1. Gestion des Activités

- **Ajout d'activités** : Ajout par l'administrateur d'une activité via un formulaire avec les champs suivants :
 - Nom de l'activité
 - Description
 - Capacité
 - Date de début
 - Date de fin
- **Affichage des activités** : Les activités sont affichées sous forme de tableau.
- **Suppression des activités** : Possibilité de supprimer chaque activité avec confirmation de l'utilisateur.

2.2. Gestion des Membres

- **Ajout de membres** : Le membre peut (inscrire) ajouter lui-même en remplissant un formulaire avec les champs suivants :
 - Nom
 - Prénom
 - Email
 - Numéro de téléphone
- **Affichage des membres** : Affichage de la liste des membres enregistrés par l'administrateur.
- **Suppression des membres** : Suppression d'un membre via un bouton avec confirmation par l'administrateur.

2.3. Gestion des Réservations

- **Affichage des réservations** : Les réservations sont affichées sous forme de tableau avec les colonnes suivantes :
 - Identifiant de la réservation
 - Identifiant du membre
 - Identifiant de l'activité
 - Date de la réservation
 - Statut (Confirmée ou Annulée)
- **Suppression des réservations** : Possibilité de supprimer chaque réservation avec confirmation.

3. Architecture Technique

3.1. Technologies Utilisées

- **Frontend** : HTML5, CSS3, Tailwind CSS
- **Backend** : PHP (version 8.x)
- **Base de données** : MySQL

3.2. Structure de la Base de Données

Table : activite

```
CREATE TABLE activite (  
  id_Activite INT AUTO_INCREMENT PRIMARY KEY,
```

```
nom_Activité VARCHAR(100) NOT NULL,  
description TEXT,  
capacite INT,  
date_debut DATE,  
date_fin DATE  
);
```

Table : membres

```
CREATE TABLE membres (  
  id_membre INT AUTO_INCREMENT PRIMARY KEY,  
  nom VARCHAR(100) NOT NULL,  
  prenom VARCHAR(100) NOT NULL,  
  email VARCHAR(150) UNIQUE NOT NULL,  
  telephone VARCHAR(20) NOT NULL  
);
```

Table : reservations

```
CREATE TABLE reservations (  
  id_reservation INT AUTO_INCREMENT PRIMARY KEY,  
  idmembre INT NOT NULL,  
  idactivite INT NOT NULL,  
  date_reservation DATETIME NOT NULL,  
  statut ENUM('Confirmée', 'Annulée') DEFAULT 'Confirmée',  
  FOREIGN KEY (idmembre) REFERENCES membres(id_membre),  
  FOREIGN KEY (idactivite) REFERENCES activite(id_Activite)  
);
```

4. Interfaces Utilisateurs

4.1. Interface Principale

- **Page d'accueil** : Accès rapide aux fonctionnalités principales.

4.2. Interface Administrateur

- **Gestion des activités** : Ajout, affichage et suppression d'activités.

- **Gestion des membres** : Affichage et suppression des membres.
- **Gestion des réservations** : Affichage et suppression des réservations.

4.3. Interface Membre

- **Gestion des membres** : Inscription du membre.
- **Consultation des activités** : Liste des activités disponibles.
- **Effectuer une réservation** : Possibilité de réserver une activité disponible.

5. Sécurité

- **Validation des formulaires** : Contrôle des champs pour éviter les entrées vides.
- **Protection contre les injections SQL** : Utilisation de requêtes préparées avec PDO.
- **Gestion des erreurs** : Affichage des messages d'erreur clairs et pertinents.

6. Instructions pour configurer l'environnement et exécuter le projet

Prérequis

- Installez un serveur local (comme XAMPP, WAMP ou MAMP).
- Installez MySQL et PHP (version 8.x ou plus).
- Installez un éditeur de code (comme Visual Studio Code).

Clonage du projet

Clonez le dépôt GitHub du projet à l'aide de la commande suivante :

`git clone https://github.com/nmissi-nadia/Gestion-des-R-servations-dans-une-Salle-de-Sport.git`

Configuration de la base de données

Ouvrez l'interface phpMyAdmin.
Créez une base de données nommée **salle_sport**.
Importez le fichier **commande.sql** fourni dans le dépôt.

Configuration de l'application

Modifiez le fichier **db_connect.php** pour adapter les informations de connexion à votre base de données.

Lancement du serveur

Placez le dossier du projet dans le répertoire **htdocs** (pour XAMPP) ou **www** (pour WAMP).
Démarrez Apache et MySQL depuis le panneau de contrôle de XAMPP/WAMP.

Accès à l'application

Ouvrez un navigateur et accédez à l'URL suivante :

[http://localhost/ Gestion-des-R-servations-dans-une-Salle-de-Sport](http://localhost/Gestion-des-R-servations-dans-une-Salle-de-Sport)

7. Conclusion

Ce projet répond à un besoin de gestion simple et efficace des activités, membres et réservations. L'application garantit un système de gestion d'activité fluide et intuitif. Ce rapport résume l'ensemble des étapes de la création de l'application, du cahier des charges à la mise en production.