**Connexion et l'Inscription des Utilisateurs**

**FRONTEND :**

**BACKEND :**

Cette section décrit la méthode de gestion de la connexion, de l'inscription, de l'affichage du statut de connexion et de la déconnexion pour les utilisateurs. Le site n'étant pas hébergé sur un serveur externe en temps réel, la connexion se fait vers une base de données locale.

**1. Création de la Table des Comptes**

**Création de la table Account :**  
Créer la table nécessaire pour stocker les informations des utilisateurs.  
Puisque le site est hébergé en local, la base de données est installée sur notre ordinateur. Lors du chargement de la page de connexion, le serveur crée et connecte automatiquement à la base locale et crée la table (si elle n'existe pas déjà) via le fichier **data.php**, a table, nommée **Account**, comporte un identifiant (id) et d’autres informations relatives à l’utilisateur.

**2. Connexion à son Compte**

**2.1 Remplissage du Formulaire**

* L'utilisateur charge la page de connexion, remplit le formulaire et appuie sur le bouton **« Se connecter »**.

**2.2 Traitement dans le Fichier login.php**

* **Envoi des Données :**  
  Les informations saisies sont transmises au fichier **login.php**.
* **Vérification :**  
  Le script parcourt (via une boucle WHILE) la table **Account** pour vérifier que
  + L'email existe bien.
  + Le mot de passe correspond à l'email renseigné.
* **Mise à Jour des Variables de Session :**  
  Si la vérification est réussie, des variables sont initialisées, en particulier la variable de session  
  $\_SESSION['login\_success'] qui est mise à **true** afin d'indiquer que la connexion est effective.
* **Gestion de l'Échec :**  
  En cas d'échec, la variable de session reste à **false** et un message d'erreur « L'email ou le mot de passe n'est pas correct » est affiché sur la page de connexion.  
  Le fichier **login.php** redirige ensuite l'utilisateur :
  + Vers la page **home.php** en cas de succès.
  + Vers la page de connexion en cas d'erreur, suivi d'une suppression (unset) de la variable de session pour éviter que le message d'erreur ne persiste.

**3. Inscription**

* 1. **Remplissage du Formulaire d'Inscription**
* L'utilisateur complète le formulaire d'inscription et soumet ses informations, ces informations sont envoyées au fichier **signup.php,** Le système se connecte à la base de données préalablement créée par **data.php** et récupère les données via la variable globale $\_POST.

**3.2 Vérification et Insertion dans la Base de Données**

* **Vérification de l'Email :**  
  Avant d'insérer les données, le script vérifie si l'email est déjà utilisé.
  + Si l'email existe, une variable (exists) est définie sur **true**, l'insertion est annulée, et un message d'erreur est affiché sur la page de connexion grâce à la variable de session $\_SESSION['email\_used'] «l'email est déjà utilisé ».
* **Insertion des Données :**  
  Si l'email n'existe pas encore dans la table, les données sont insérées directement dans la table **accounts**.  
  Le compte est alors créé avec succès, et l'utilisateur est redirigé vers la page de connexion avec un message de confirmation indiquant la bonne création du compte.

**4. Affichage du Nom de l'Utilisateur sur le Header**

**4.1 Initialisation des Variables de Session**

* Lors de la connexion réussie, la variable $\_SESSION['login\_success'] est définie sur **true**.
  + D’autres variables importantes sont également initialisées, telles que :
    - $\_SESSION['id']
    - $\_SESSION['prenom']
    - $\_SESSION['nom']

**4.2 Affichage dans le Header**

* Le header du site utilise la variable $\_SESSION['login\_success'] pour déterminer si un utilisateur est connecté.
* Si c'est le cas, le header affiche :
  + Le prénom de l'utilisateur.
  + Un bouton ou un lien **« Se déconnecter »**.
* La variables de session relié a ‘id’ sera utilisé ultérieurement avec la page des activités.

Grâce aux variables de session, on peut savoir l’état de connexion de l’utilisateur (soit connecté ou non connectée).

**5. Déconnexion**  
Pour se déconnecter, il suffit de cliquer sur le lien ou le bouton **« Se déconnecter »** situé dans le header.  
Le serveur exécute alors le code PHP contenu dans le fichier **disconnect.php**, lequel appelle la fonction session\_unset().  
Cette fonction supprime toutes les variables de session créées, et l'utilisateur est ensuite redirigé vers la page d'accueil.

**Inscriptions aux activités :**

**Frontend :**

**Backend :**

* + - 1. **Les tables de réservation**

Dès que l'utilisateur charge la page des activités, le contenu du fichier data.php sera exécuté également. Ce fichier se connecte à la base de données déjà créée et crée les tables xxx\_res. Chaque sport dispose de sa propre table, laquelle contient un identifiant de réservation, l'identifiant de l'utilisateur ainsi que d'autres attributs. Ce fichier permet également de créer une table Max\_members qui contient les identifiants des sports pouvant exister sur notre plateforme ainsi que le nombre maximal de places pouvant être réservées. Cette table sera utilisée ultérieurement pour déterminer le nombre de places restantes, comme nous en reparlerons par la suite. De plus, les données sont insérées directement dans cette table, et la variable count sert à éviter de remplir cette table plus d'une fois, puisque le contenu du fichier peut être exécuté plusieurs fois si l'utilisateur rafraîchit la page des activités.

* + - 1. **Réservation des séance :**
  + **2.1 Le botton de reservation :**

J'assume que vous avez lu la partie concernant la connexion, car à ce stade nous allons utiliser la variable $\_SESSION['login\_success'], qui détermine si le bouton de réservation doit être affiché à l'utilisateur. Cette condition, placée avant chaque ligne HTML dans le fichier *activities*, vérifie si l'utilisateur est connecté. S'il l'est, le bouton de réservation est visible. Sinon, il ne voit que le bouton de connexion, présent en bas de la page ainsi que dans l'en-tête.

* + **2.2 Le botton de connexion en bas de la page :**

Rien de spécial, c'est le même algorithme que celui utilisé pour le header : il affiche le bouton de connexion si l'utilisateur n'est pas connecté et ne l'affiche pas dans le cas contraire.

* + **2.3 Indication de max et places restantes pour chaque cours**

Grâce à deux fichiers, « set max members.php » et « remaining places.php», nous pouvons indiquer le nombre maximum de places ainsi que le nombre de places restantes. En effet, le premier fichier saisit les informations nécessaires dans les variables de session pour définir le nombre maximum de places. Chaque cours possède sa propre variable de session (c'est-à-dire une case dans une table de session qui agit comme un dictionnaire).

Le deuxième fichier dépend du premier, car il effectue un calcul avant d'initialiser d'autres valeurs de session correspondant au nombre de places restantes. Pour ce faire, il calcule le nombre de lignes de chaque table de cours, ce qui représente le nombre de places occupées, puis soustrait ce chiffre du nombre maximum de places pour obtenir le nombre de places restantes. Toutes ces variables seront utilisées dans le fichier « activities.php », où ces statistiques sont affichées.

* + **2.2 Réservation d’une séance**

Il suffit de cliquer sur le bouton de réservation, ce qui exécutera le code contenu dans le fichier nommé « append\_xxx.php », où xxx représente l'intitulé du cours.

Parlons maintenant du fichier « append\_xxx.php ». Tous ces fichiers utilisent le même algorithme ; la seule différence réside dans le cours pour lequel l'utilisateur a effectué la réservation. Voici comment cela fonctionne :

* On initialise une variable $\_SESSION["cour\_reservé"], qui sera utilisée dans le fichier appelé ensuite, intitulé «already\_subscribed.php».
  + - * + Dans ce fichier, on vérifie si l'utilisateur est déjà inscrit à ce cours. S'il l'est, un message indiquant qu'il est déjà inscrit est affiché (voir la condition dans le fichier activities.php, qui dépend de la valeur de $\_SESSION["is\_subscribed"]).
        + Dans le cas contraire, on passe à l'étape suivante dans le fichier « append\_xxx.php », qui consiste à tester s'il reste des places disponibles en se basant sur la variable liée au cours à réserver.
        + Si tout se passe bien, le cours sera réservé en ajoutant une ligne dans la table correspondante. Si aucune place n'est disponible, une erreur est affichée dans la page activities.php, conditionnée par la valeur de $\_SESSION["registration\_success\_max\_member"].

Ce processus s'applique à tous les autres fichiers présents dans le dossier « append ».

* + **Page de Recap**

Si l'opération de réservation réussit, alors on affiche quelques informations relatives à l'utilisateur ainsi qu'au cours, en utilisant les données stockées dans la variable de session et en appliquant une condition de niveau spécifique lorsque le cours concerne le yoga ou le Pilates.