**Connexion et l'Inscription des Utilisateurs**

**FRONTEND :**

**Vérification du remplissage des formulaires :**

Dans cette partie il est question de vérifier les informations remplies dans les formulaires de **connexion** et d’**inscription** par l’utilisateur sont conformes en utilisant javascript avant de les traiter côté backend.

Pour ce faire nous avons écris deux fonctions une pour vérifier les informations du formulaire de connexion (**checkform**) et une autre pour verifier les informations du formulaire d’inscription (**checkformSignup**) lorsque l’utilisateur clique sur « **se connecter** » ou « **créer le compte** ».

1. **Vérification du formulaire d’inscription** (**checkform**)

Cette fonction réalise les instructions suivantes :

* Récupérer les éléments l’email et le mot de passe du formulaire de connexion via leur id.
* Retirer les messages d’erreur précédents pour éviter l’accumulation lors de plusieurs soumissions
* Vérifier si l’email ou le mot de passe est vide, si oui un message d’erreur est généré spécifiant de remplir tous les champs.
* Vérifier à l'aide d'une expression régulière (**regexmail**) si le format de l'adresse mail est correct sur le plan de la forme. Si l'adresse mail n'est pas valide (ex : exemple@domaine.com), il sera signalé à l'utilisateur qu'il doit renseigner un email valide.
* Empêcher la soumission du formulaire en cas d’erreur avec **event.preventDefault()**

1. **Validation du formulaire d’inscription (checkformSignup)**

Cette fonction réalise les instructions suivantes :

* Récupérer les différents champs du formulaire d'inscription : nom, prénom, date de naissance, adresse, etc.
* Supprimer d'éventuels messages d'erreurs déjà affichés.
* Vérifier que tous les champs sont bien renseignés. Si ce n'est pas le cas, un message apparaît.
* S'assurer que les deux mots de passe donnés sont bien similaires. Si ce n'est pas le cas, un message d'erreur apparaîtra.
* Vérifie que l'utilisateur a bien coché la case "J'accepte les conditions d'utilisation". Si ce n'est pas le cas, il bloque l'envoi et envoie un message d’erreur.
* Vérifier à l'aide d'une expression régulière (**regexmail**) si le format de l'adresse mail est correct sur le plan de la forme. Si l'adresse mail n'est pas valide (ex : exemple@domain.com), il sera signalé à l'utilisateur qu'il doit renseigner un email valide.
* En cas d’erreur, ajoute un message dans le bloc #block2 qui signalera visuellement à l’utilisateur qu’il y a une ou plusieurs erreurs.

**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.

**Activités :**

**Frontend :**

**Objectif :** Créer une page qui présente nos activités ainsi que les cours et qui permet aux utilisateurs de s’inscrire aux cours.

**Structure :**

Elle est constituée d’un **header**, d’un **main**, d’un **footer.**

Concernant le header et le footer nous en avons parlé en amont au niveau de la page Accueil.

* **Main de la page des activités**

Il est constitué de 3 partie, une partie heros, une parties d’inscription aux cours , une bannière de promotion et une un bouton qui quand l’utilisateur n’est pas connecté l’invite à s’inscrire.

* **Partie 1 : héros**

Cette section constituée d’un texte percutant et d’une image frappante, permet d’attirer l’attention de l’utilisateur et éventuellement de le persuader à vouloir en apprendre plus sur le site.

* **Partie 2 : Activités**

Cette partie est dédié à la présentation de nos cours avec toutes les informations importantes comme l’intitulé du cours, le nombre d’heure, etc. Chaque cours est constitué en block avec son titre et sa description ainsi qu’un bouton s’inscrire qui apparait lorsque l’utilisateur est connecté.

* **Partie 3 : Promotion**

Cette partie propose à l’utilisateur une promotion de réduction des frais lors de la première réservation l’incitant ainsi à s’inscrire à notre site.

Lorsque l’utilisateur s’inscrit à un cours il est redirigé vers une page qui se nomme **récapitulatif d’inscription** qui contient un résumé de l’inscription au cours et un bouton de retours à la page de cours.

**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.

**Accueil :**

**Objectif :** Créer une page d’accueil qui présente les fonctionnalités du site, les services qu’il offre, une présentation du site (à propos, les coachs).

**Structure :**

Elle est structurée de façon suivante :

**1-header**

**2-Main**

**3-Footer**

* + - **Le header**

Il est constitué du logo du site Sportify, et des liens vers les fonctionnalités ou les autres pages du site.

* + - **Le main**

Il est constitué de 5 sections permettant à l’utilisateur d’avoir une impression sur le site.

* **Section 1 : héros**

Cette section constituée d’un texte percutant et d’une image frappante, permet d’attirer l’attention de l’utilisateur et éventuellement de le persuader à vouloir en apprendre plus sur le site.

* **Section 2 : à propos**

Cette section est constituée de deux textes un à gauche et l’autre à droite.

Le texte de gauche présente de façon brève l’entreprise et son objectif, et le texte de droite qui incite l’utilisateur à nous rejoindre.

* **Section 3 : services**

Cette section présente les services qu’offre le site Sportify ainsi qu’une brève description de ces services.

* **Section 4 : coachs**

Dans cette section, 3 de nos coachs disent un mot pour motiver les utilisateurs.

* **Section 5 : Témoignages**

Cette section contient les témoignages de nos clients satisfaits.

* + - **Le footer**

Cette partie contient l’adresse où se déroule les activités, le mail de l’université, le numéro du site, et le copyright de Sportify.