Connexion et Inscription

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** ».

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()**

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

3.1 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 de l'État de Connexion et du Nom de l'Utilisateur

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 »**.
- Les variables id et nom peuvent être utilisées sur d'autres pages pour personnaliser l'expérience utilisateur.
- Grâce aux variables de session, toutes les pages du site peuvent vérifier l'état de connexion de l'utilisateur et afficher les informations correspondantes.

5. Déconnexion

- **Procédure :**
  Pour se déconnecter, il suffit de cliquer sur le lien ou le bouton **« Se déconnecter »**.
- **Traitement :**
  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.

Page d'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 :

1. Le header

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

2. 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.

3. 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.

Page des activités

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.

- **Section 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.

- **Section 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.

Page de contact :

Frontend :

Rien de spécial, un formulaire centré au milieu de la page, avec deux boutons. L'un permet de tout supprimer, et l'autre d'envoyer les données au fichier qui les traite. La structure ressemble beaucoup à celle utilisée sur la page de devis.

Backend :

Vérification de la Méthode de Soumission

Le backend commence par vérifier si le formulaire a été soumis via la méthode POST. Cela signifie que le serveur ne traite les données que lorsqu'un utilisateur soumet effectivement le formulaire.

Récupération et Sécurisation des Données

Une fois le formulaire soumis, les données sont envoyées au serveur. Le backend récupère ces informations (nom, prénom, email, message) à partir de la requête POST. Ces données sont ensuite sécurisées en utilisant une méthode pour éviter des risques de sécurité comme les attaques XSS (Cross-Site Scripting). Cela permet de rendre l'application plus sûre en nettoyant les caractères spéciaux qui pourraient être utilisés de manière malveillante.

Préparation de l'Email

Le backend crée un email contenant toutes les informations envoyées par l'utilisateur (nom, prénom, email, message). Cet email sera envoyé à l'adresse spécifiée dans le script (celle de l'administrateur ou du responsable du site). Le sujet et le contenu de l'email sont également définis dynamiquement en fonction des données du formulaire.

Envoi de l'Email

Le script utilise la fonction PHP mail() pour envoyer l'email. Si l'email est envoyé correctement, l'utilisateur reçoit une confirmation (via une alerte JavaScript). Si l'envoi échoue, un message d'erreur est renvoyé à l'utilisateur pour lui indiquer qu'il doit réessayer.

Retour à la Page d'Accueil

Après l'envoi du message, que ce soit un succès ou un échec, l'utilisateur est redirigé vers une autre page du site (dans ce cas, la page d'accueil). Cela permet de nettoyer l'état du formulaire et de donner à l'utilisateur une nouvelle perspective après l'interaction.

Page de devis

Objectif de la Fonctionnalité

L'objectif principal de cette fonctionnalité est de valider et de calculer automatiquement un devis en fonction des données saisies par l'utilisateur sur un formulaire. Lorsque l'utilisateur remplit le formulaire avec ses coordonnées, son choix de service, le type de cours, le nombre de sessions, etc., un prix est calculé en fonction des informations fournies. Une fois le formulaire validé et soumis, l'utilisateur reçoit une confirmation par email et l'administrateur reçoit également une notification. Ce processus permet une communication fluide entre l'utilisateur et l'administrateur pour assurer un suivi rapide et efficace de la demande.

Fonctionnement :

1. **Collecte et Traitement des Données** :
    - L'utilisateur remplit un formulaire où il sélectionne son service (individuel, collectif, programme), le type de cours (yoga, fitness, musculation), le nombre de sessions et d'autres détails. Ces informations sont collectées via JavaScript.
2. **Calcul Dynamique du Prix** :
    - Le script JavaScript permet de calculer automatiquement le prix du devis en fonction des choix de l'utilisateur. Par exemple :
    - Si le service choisi est "individuel", le prix est de 50.
    - Si le service est "programme", un prix de base est ajouté et un supplément est ajouté en fonction du type de programme sélectionné (rééducation, entraînement, suivi).

- Le type de cours et le nombre de sessions influencent également le prix. Chaque type de cours a un coût par session qui est multiplié par le nombre de sessions.

3. **Affichage Dynamique des Champs** :
    - Selon le service choisi, certains champs du formulaire sont affichés ou masqués dynamiquement. Par exemple, si "programme" est sélectionné comme service, un champ supplémentaire apparaît pour permettre à l'utilisateur de choisir un type de programme.
4. **Validation du Formulaire** :
    - Avant de soumettre le formulaire, le script JavaScript valide les champs pour s'assurer que tous les champs obligatoires sont remplis et que les données sont correctes :
    - Les emails sont validés par une expression régulière pour s'assurer qu'ils respectent le format standard.
    - Les numéros de téléphone sont validés selon le format français.
    - Tous les champs nécessaires doivent être remplis (coordonnées, service, type de cours, etc.).

Si une erreur est trouvée, un message d'erreur est affiché, et le formulaire ne peut pas être soumis tant que l'utilisateur n'a pas corrigé ses erreurs.

5. **Soumission et Confirmation** :
    - Après validation, si aucun problème n'est détecté, l'utilisateur soumet le formulaire. Deux emails sont envoyés :
    - Un email de confirmation à l'utilisateur pour lui indiquer que sa demande a bien été reçue.
    - Un email à l'administrateur pour l'informer qu'une nouvelle demande de devis a été faite.

L'utilisateur est ensuite redirigé vers la page de devis ou reçoit un message d'erreur si quelque chose s'est mal passé durant l'envoi de l'email.

Configuration de l'envoi de mail sur mon PC

Afin de pouvoir envoyer des e-mails depuis un environnement local WAMP, une **configuration spécifique a été réalisée** sur mon ordinateur en utilisant **Sendmail** et le fichier php.ini de PHP.

Voici comment cela fonctionne :

- **Sendmail** est un petit outil qui permet à PHP, en local, d'envoyer des e-mails via un serveur SMTP (comme Gmail ou un serveur externe).
- Le fichier php.ini contient les paramètres qui permettent à PHP de communiquer avec Sendmail.

Pour que tout fonctionne correctement :

1. J'ai installé **Sendmail** pour **Windows**.
2. J'ai modifié le fichier sendmail.ini pour y inscrire les **données SMTP** :
    - Adresse du serveur SMTP : thiernooury433@gmail.com
    - Port : 587
    - Email et mot de passe d'application pour Gmail.
3. Ensuite, dans le fichier php.ini, j'ai activé l'envoi de mail en **liant PHP à Sendmail**
    - En indiquant le chemin vers l'exécutable sendmail.exe
    - En ajustant les paramètres de messagerie