**Solutions pour la 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 : jean@bonjour), 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 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.