

## TP Authentification à un site et gestion de sessions en PHP

### Exercice 1 : création

En utilisant l'interface PHPMyAdmin, créez dans votre base de données, une table "utilisateurs". Cette dernière doit contenir les champs nom, prenom, login (qui pourra correspondre à une adresse email), un mot de passe, ainsi qu'un numéro d'identifiant auto-incrémenté (id\_personne) servant de clé primaire.

*utilisateurs(id\_personne, nom, prenom, login, mot\_de\_passe)*

Remplissez, toujours à l'aide de l'interface PHPMyAdmin, la table avec au moins 3 utilisateurs.

### Exercice 2 : consultation

Ecrivez un script PHP (affiche\_utilisateurs.php) qui affiche le contenu de la table "utilisateurs".

### Exercice 3 : identification des utilisateurs

1. Ecrivez un formulaire ("login.php") qui demande à l'utilisateur son login et son mot de passe.

2. Ecrivez ensuite le script PHP qui traite le formulaire de login ("traitelogin.php") : vérification de l'existence du login et du mot de passe. Si l'utilisateur est reconnu, affichez un récapitulatif de ses données. Sinon, renvoyez-le à la page de login en lui indiquant qu'il s'est mal identifié : pour cela vous pouvez soit :

- *simple* : renvoyer un indicateur d'erreur en GET (login.php?erreur=login) et modifier le fichier login.php en conséquence.

- *plus compliqué* : utiliser Ajax

3. Effectuez les modifications nécessaires pour qu'une session soit créée. Modifiez le script **traitelogin.php** pour sauvegarder l'id et le nom de la personne identifiée comme variable de session si la personne entre une bonne combinaison login/mot de passe. Pensez à modifier toutes les pages de ce TP pour qu'une session soit démarrée.

4. modifiez le script « affiche\_utilisateurs.php » pour que la page ne s'affiche que si l'utilisateur est effectivement connecté : on vérifie que \$\_SESSION["login"] existe pour afficher le contenu, sinon on affiche un message du type « vous devez être logué pour accéder à cette ressource »

## SEANCE 2

### Exercice 4 : mots de passe

Pour introduire un peu de sécurité, le mot de passe ne sera pas stocké en clair dans la base de données. Vous allez utiliser la fonction password\_hash() pour "hacher" le mot de passe et ainsi stocker le condensat. Pensez éventuellement à modifier la taille du champ mot de passe dans la base.

Modifiez alors la procédure d'identification (attention, il faut comparer le condensat avec `password_verify()`!).

### **Exercice 5**

Modifier la page consultation de manière à afficher le nom de l'utilisateur actuellement logué en rouge, à l'aide de l'information récupérée dans les variables de session

### **Exercice 6 : formulaire d'inscription**

1. Créez un formulaire permettant à l'utilisateur de saisir toutes les informations relatives à la table "utilisateurs". (Vous pourrez vérifier en JS que les champs sont bien remplis avant d'envoyer le formulaire). Pensez à demander de saisir 2 fois le mot de passe pour parer à une éventuelle faute de frappe.

2. Ecrivez un script "traite\_inscription.php" qui sera appelé par le formulaire, et qui effectuera le traitement des données soumises.

### **Exercice 7 Ajout d'une photo de profil**

L'objectif est que chaque utilisateur puisse, lorsqu'il s'est inscrit, ajouter une photo de profil.

- Ajouter un champ photo (varchar 80) dans la table utilisateur.

Chaque photo devra être stockée dans un répertoire unique « photo ». Tous les fichiers de photo auront pour nom profil\_XX où XX représente l'id de l'utilisateur.

Créez une page **form\_photo.php** qui propose à un utilisateur connecté d'uploader une photo pour son profil. Pour cela créez un formulaire récapitulant les informations d'un utilisateur logué et lui proposant d'ajouter une photo de profil. Pour cela vous récupérerez les informations de l'utilisateur dans la base de données à partir de son id qui a été stocké comme variable de session.

Le formulaire envoie la photo au script **ajout\_photo.php** qui vérifie que la photo est bien un jpeg de moins de 1Mo, sauvegarde la photo dans le bon répertoire avec le bon nom (ex : photo/profil\_12.jpg).

Question : Dans le cas présent, le stockage du chemin dans la base de donnée était-il nécessaire ?

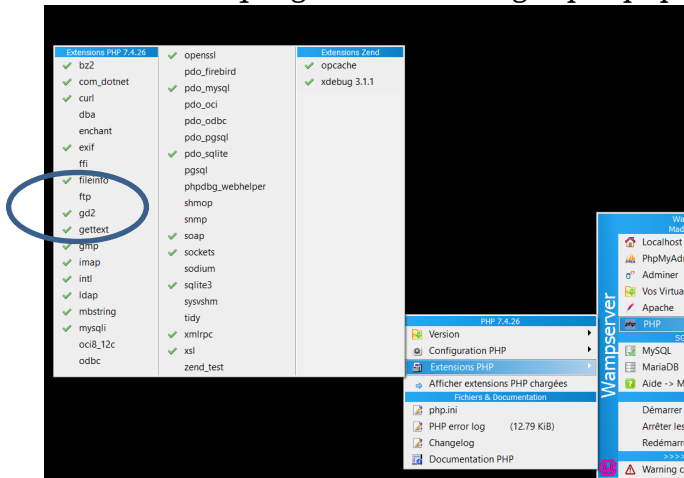
### **Exercice 8 (pour aller plus loin)**

Modifiez le script afin que la photo soit enregistrée au format photo d'identité (828 x 1143 pixels)

Renaud Eppstein  
MMI2 2025-2026

Pour cela on pourra utiliser la librairie « gd » de php et plus particulièrement la fonction `imagecopyresampled`.

Pour vérifier que gd est bien chargée par php :



Liens utiles sur le redimensionnement :

<https://stacklima.com/comment-redimensionner-une-image-jpeg-en-php/>

on pourrait aussi crop une image à l'aide d'un script JS :

<https://phpspot.com/php/upload-and-crop-image-using-php-and-jquery/>