

TP3 : Les Sessions en PHP

Exercice 1 : On démarre une session en initialisant trois variables dans la session : prenom, age, race. Chacune a sa valeur.

1. d'afficher les variables de session. Je vais donc lire séquentiellement le tableau \$_SESSION avec une boucle foreach de la manière suivante :
2. Pour chaque couple clé/valeur de \$_SESSION, affichez dans un (élément de liste) en gras la clé et en italique sa valeur.

Utilisez la fonction ucfirst : cette fonction (UpperCase First) affiche la valeur qui lui est soumise avec la première lettre en majuscule.

On a ainsi de quoi afficher toutes les variables si besoin, à n'importe quel moment je peux faire appel à \$_SESSION['race'] par exemple. Par print_r(\$_SESSION); on affiche ainsi dans le code source :

Array

```
( [prenom] => Mounir  
  [age] => inconnu  
  [race] => humain)
```

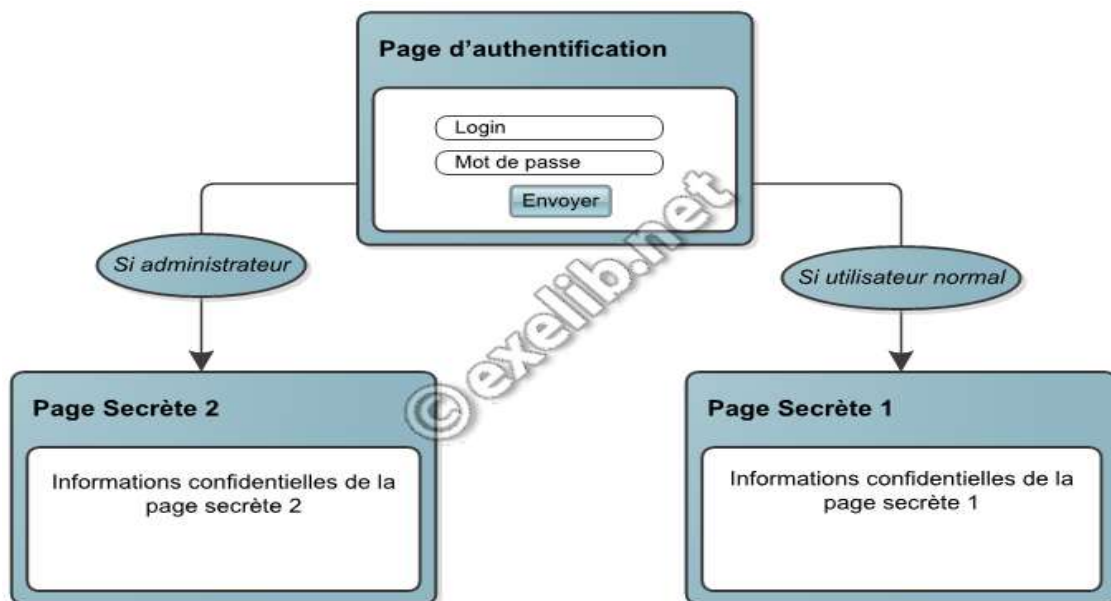
Exercice 2- Authentification protégé

Objectif:

- Protéger l'accès à des pages privées.
- Utiliser et détruire les sessions.
- Redirection.

Énoncé :

Le but de l'exercice est de réaliser la structure suivante :



Schémas authentification avec compte Admin

1. Créer la page "login.php" qui demande à l'utilisateur la saisie de son login et son mot de passe dans un formulaire .

On suppose avoir deux utilisateurs: User1 (Login="User",Passw="U123") et Admin (Login="Admin",Passw="Admin123") :

- a. L'admin peut accéder à toutes les pages alors que User1 ne peut accéder qu'à la page secrète_1.php seulement.
 - b. Si l'utilisateur demande les pages secrètes directement, il devra être redirigé vers la page d'authentification.
 - c. Si l'utilisateur est déjà authentifié et redemande la page login.php , ça serait mieux de lui afficher un message de bienvenue et un lien pour la déconnexion au lieu d'afficher le formulaire d'authentification.
2. Afficher dans les pages secrètes un message de bienvenue avec le login de l'utilisateur (Par exemple :Bonjour User1) et un lien pour la déconnexion.

Exercice 3- Authentification via une base de données en PHP

On aura donc besoin des pages suivantes:

- **inscription.php**: qui permet à l'utilisateur de créer un compte. Cette page contient un formulaire avec les champs: nom, prenom, login, mot de passe, confirmation de mot de passe et un bouton d'envoi.
- **login.php**: qui authentifie l'utilisateur. Elle contient un formulaire renfermant une zone de texte, une zone de mot de passe et un bouton d'envoi.
- **session.php**: qui représente la page à accès limité. Aucun visiteur n'a le droit de voir son contenu s'il n'a pas été authentifié par la page **login.php**.
- **deconnexion.php**: est la page qui permet de déconnecter le client (détruire la session) et rediriger le navigateur vers la page **login.php**.
- **connexion.php**: qui contient la chaîne de connexion.

Lors de l'inscription, aucun champ ne doit être laissé vide et les champs "mot de passe" et "confirmation de mot de passe" doivent être identiques. Si tous les champs sont valides, on vérifie d'abord si le login n'existe pas déjà dans la base de donnée. Si non alors les informations de l'utilisateur seront placées dans la table "utilisateurs" vue dans la page précédente dont voici la structure:

```
CREATE TABLE `utilisateurs` (  
  `id` int(10) unsigned NOT NULL auto_increment,  
  `date` timestamp NOT NULL default CURRENT_TIMESTAMP on update  
CURRENT_TIMESTAMP,  
  `nom` varchar(40) NOT NULL,  
  `prenom` varchar(40) NOT NULL,  
  `login` varchar(40) NOT NULL,  
  `pass` varchar(40) NOT NULL,  
  PRIMARY KEY (`id`)  
);
```