
TP PHP 4

1. Finir le TP de la semaine dernière
2. Choix de langues
 - On souhaite pouvoir proposer notre site aussi bien en anglais qu'en français. On veut également pouvoir passer d'une langue à l'autre simplement. Enfin, on souhaite ne pas faire deux sites séparés pour chaque langue. PHP va nous permettre d'effectuer ceci assez simplement.
 - Dans un premier temps, créer les fichiers `profil_fr.php` et `profil_en.php`. Dans chacun de ces fichiers, créez les variables PHP `$titre`, `$login`, `$pwd`, contenant respectivement un titre de page en français et un titre en anglais, le texte « nom d'utilisateur » et le texte « login », le texte « mot de passe » et le texte « password ».
 - On suppose maintenant que la langue demandée est dans une variable `lang` envoyée par `$_GET`. En fonction de la valeur de cette variable, inclure le fichier php correspondant.
 - Remplacer dans `profil.php` le texte par les variables qui correspondent.
 - Faire les liens adéquat pour pouvoir changer la langue (on rappelle qu'on passe la langue choisie par la méthode GET).
 - Généraliser à l'ensemble du site.
 - Faire en sorte que les liens du menu conservent la langue choisie.
 - Ajouter un cookie (regarder la documentation de `setcookie()`) tel que la langue choisie est conservée même après avoir quitter la page pendant 1 mois. Un cookie est accessible dans le tableau `$_COOKIE`.
3. Mises en évidence
 - Faire en sorte que la page actuelle s'affiche en gras dans le menu.
 - Faire de même avec la langue actuelle.
4. Lecture/Écriture de fichiers
 - On va maintenant voir la lecture/écriture de fichiers en PHP. C'est assez semblable au C. Pour cela, on imagine que la description d'un profil d'utilisateur est stocké dans un fichier (disons « `files/login_descri.txt` »). On va afficher sa description sur son profil, puis on proposera de l'éditer.
 - Dans un premier temps, dans la fonction `affiche_profil`, ouvrir le fichier avec `fopen` (vérifier le code de retour !), le parcourir ligne à ligne avec `fgets()`, puis le fermer. Pendant le parcours, on concatène l'ensemble des lignes dans une variable `$profil`.
 - Écrire un formulaire HTML avec un champ caché de nom « `Edit1` » contenant en valeur `$profil`, et un bouton de validation avec pour valeur « Éditer ». La méthode du formulaire est par POST, et a pour cible la page `profil`.
 - Dans la page `profil`, s'il existe une variable passée par POST du nom de « `Edit1` », afficher un formulaire HTML de méthode POST et de cible la page `profil` contenant un champ `textarea` de nom « `Edit2` », avec le contenu de la variable « `Edit1` ».
 - Dans la page `profil`, s'il existe une variable passée par POST du nom de « `Edit2` », alors enlever toute balise HTML ou PHP pouvant créer des problèmes de sécurité avec `strip_tags()`, puis écrire dans le fichier correspondant au profil le contenu de la variable « `Edit2` » avec la fonction `fwrite()`. N'oubliez pas de fermer le fichier !
5. Accès privés
 - On souhaite qu'une partie du site ne soit accessible qu'à certains utilisateurs. Nous allons utiliser un fichier `.htaccess`, qui permet d'ajouter des règles d'authentification puisque nous avons pas accès à la configuration du serveur étudiant (voir aussi <http://httpd.apache.org/docs/trunk/fr/howto/htaccess.html>)
 - Créer un répertoire admin, contenant un fichier tout bête.
 - Créer dans ce répertoire un fichier `.htaccess` contenant :

AuthName "Authentication"

AuthType Basic

AuthUserFile "/chemin/vers/le/fichier/password"

Require valid-user

- Ce fichier demande une authentification pour l'accès à tout fichier se trouvant dans ce répertoire. Il a également besoin d'un chemin absolu vers un fichier d'authentification contenant des couples utilisateur:mot de passe (un fichier texte, avec un couple par ligne, avec séparation entre utilisateur et mot de passe par ':').
 - Créez un fichier pwd dans ce répertoire, contenant sur une ligne login:mdp
 - Ajouter le chemin absolu vers ce fichier dans votre fichier .htaccess (vous pouvez utiliser la commande unix pwd, où la fonction php realpath())
 - Mettre les mots de passes en clair est un peu risqué. On va chiffrer ces mots de passes à l'aide du programme htpasswd (htpasswd -h est assez clair sur les commandes à effectuer). Au pire, la fonction php crypt() donne le mot de passe crypté correspondant.
 - Même s'il est impossible de revenir au mot de passe initial à partir du mot de passe chiffré, on souhaite empêcher l'accès au fichier pwd. Ajouter dans votre fichier .htaccess les balises FilesMatch (<http://httpd.apache.org/docs/current/mod/core.html#filesmatch>), contenant « deny from all ».
6. Vous pouvez aller aider vos camarades n'ayant pas fini le TD.