

TP PHP 3

Dans ce TP, nous allons créer un site web contenant plusieurs pages et partageant des informations à l'aide de formulaires et de sessions.

Mise en place

Dans votre répertoire WWW/test, créez un répertoire dédié à ce nouveau site :

WWW/test/my_facebook/

Dans ce répertoire, vous aurez besoin de deux sous répertoires : « includes » et « contents ».

1- Créez les fichiers « header.php » et « footer.php » à l'intérieur du répertoire includes. Ils doivent contenir respectivement les entêtes (jusqu'à <body>) et pieds de page de votre page html.

2- Dans le répertoire « contents » créez le fichier accueil.php qui ne contiendra que :

```
<p>Page d'accueil</p>
```

3- Enfin, à la racine du site, dans le répertoire « my_facebook », créez la page index.php qui contiendra trois instructions include : pour l'entête, le contenu puis le pied de page.

Vous avez maintenant une structure de site correct, nous allons pouvoir créer les différentes pages.

Pagination et menu

Dans le répertoire « includes », créez un fichier « settings.php ». Ce fichier ne devra contenir que du PHP et ne générer du HTML qu'à travers des fonctions. Ainsi, si vous essayez de voir ce fichier avec votre navigateur, une page blanche s'affichera. Tous les prétraitements et définitions de variables ou fonctions se feront dans ce fichier.

Attention : pensez à bien ouvrir votre balise php dès la première ligne du fichier et à ne laisser aucune ligne vide après la fermeture de la balise. En effet, cela générerait des lignes vides « déchets » dans votre fichier HTML.

Nous allons déclarer un tableau \$pages qui contiendra des informations sur les pages du site. Les clés seront des chaînes de caractères (sans accents ni espaces) qui représenteront le nom des pages. Il leur sera associé un tableau dont la première valeur sera le titre de la page et la seconde, le sous titre :

```
$pages = array(
    « accueil » => array(« My facebook », « Bienvenue sur My facebook »),
    « profil » => array(« Profil My Facebook », « Voici un profil My Facebook »),
    « conditions » => array(« Conditions », « Nos conditions d'utilisation »)
);
```

Après la déclaration du tableau, testez si une variable « page » a été envoyée en GET. Si oui, vérifiez qu'elle est bien une clé du tableau. Si la page envoyée est incorrecte ou si aucune page n'a été envoyée, initialisez la variable \$page à « accueil », sinon, utilisez la valeur envoyée par GET.

Maintenant, créez les fichiers profil.php et conditions.php dans le répertoire « contents », ils pourront contenir quelques lignes de texte dans des paragraphes. A chaque clé du tableau \$pages, doit correspondre un fichier de même nom dans le répertoire contents,

Reprenez votre fichier index.php. Utilisez la fonction **require_once** pour inclure le fichier « settings.php ». Puis modifiez le fichier index pour qu'il inclue la page correcte et non plus uniquement la page d'accueil. Testez vos différentes pages en envoyant l'argument par l'url.

Enfin, avant d'inclure le contenu de la page, créez un menu dans le fichier index.php. Votre menu devra être une liste html :

```
<ul>
    <li><a href=«index.php?page=... »>Titre de la page</a></li>
    ...
</ul>
```

Pour cela, utilisez un foreach sur le tableau \$pages

Titre de Pages, H1 et H2

Vérifiez bien que votre fichier settings.php a été inclus avant l'entête html. Cependant, faites bien attention à ne pas laisser de lignes vides avant l'ouverture de votre balise php dans index.php pour qu'aucune ligne ne soit générée avant l'entête.

A présent, modifiez votre fichier header.php pour que le titre de la page (balise <title> à l'intérieur de <head>) soit le titre de la page courante, défini dans le tableau \$pages. Dans le fichier index.php affichez le titre et le sous titre de la page à l'aide des balises <h1> et <h2> avant l'inclusion du contenu.

Gestion des 404

Pour l'instant, quand on demande une page inconnue, le site affiche la page d'accueil. Nous voulons à présent générer une page d'erreur « 404 ». Lors de l'ouverture d'une page un « client » fait une demande sur un « serveur », si la page est présente, le serveur renvoie un code de retour égal à 200, si la page n'a pas été trouvée, il renvoie 404. Ici, c'est en théorie la page « index.php » qui est demandée au serveur, mais nous voulons modifier sa réponse pour qu'il renvoie un code 404 et signifie ainsi au client que la page n'existe pas. Pour cela, on utilise la fonction **header** de cette façon :

```
header("HTTP/1.0 404 Not Found");
```

Dans votre fichier settings.php, créez une fonction **affiche_404** qui appelle la fonction **header**, puis affiche un message d'erreur et enfin sort du script php avec **exit**. Vous appellerez cette fonction si la page demandée en GET n'existe pas.

Profils

Nous allons maintenant créer des utilisateurs avec profils.

Dans votre fichier settings.php, déclarez un tableau \$users dont les clés seront des login utilisateurs et les valeurs des tableaux contenant un mot de passe, un nom, un prénom, une description et un nom d'image en format jpg.

Si la page sélectionnée est la page « profil », testez si vous avez reçu une variable « user » en GET et si elle correspond à un utilisateur du tableau. Si oui, initialisez la variable \$selected_user.

A présent, dans la page « profil.php », si la variable \$selected_user a été initialisée, affichez le profil correspondant. Pour cela, vous pouvez définir une fonction **affiche_profil** dans settings.php et l'appeler dans le fichier profil.php.

Les images des profils devront se trouver dans : my_facebook/images/

Si aucun profil n'a été envoyé, affichez un menu avec des liens vers les différents profils du type

```
<a href="profil.php?user=login" > Prénom Nom </a>.
```

Utilisez pour cela une boucle foreach sur le tableau \$users.

Formulaires et sessions

Pour éviter les problèmes d'atteinte à la vie privée, nous voulons maintenant n'afficher les profils que si le visiteur est identifié. Pour cela, nous allons utiliser des **sessions**. Une session est une série d'informations enregistrées sur un visiteur donné. Chaque nouveau visiteur est identifié et on peut enregistrer sur le serveur des informations temporaires le concernant. Pour cela nous allons avoir besoin des fonctions **session_start()**, **session_destroy()**, **isset(\$var)**, **unset(\$var)** ainsi que du tableau **\$_SESSION**

Première étape : le formulaire d'identification

Dans votre fichier settings.php, testez si vous avez reçu en POST un login et un mot de passe. Si c'est le cas, vérifiez qu'ils correspondent à un utilisateur valide : il faut que le login soit dans le tableau \$users et que le mot de passe soit correct. Si tout est correct, vous pouvez initialiser la variable \$identified_user. Si un mauvais login ou mot de passe a été envoyé, enregistrez un message d'erreur dans une variable \$error_login.

A présent, dans votre page profil.php, vérifiez avant d'afficher le contenu que la variable \$identified_user a été initialisée. Sinon, affichez un formulaire demandant un login et un mot de passe. N'oubliez pas d'afficher le message d'erreur s'il existe.

Nous avons maintenant un formulaire, mais si l'utilisateur change de page, l'identification est perdue, c'est pourquoi nous avons besoin des sessions.

Deuxième étape : sessions

Après la déclaration du tableau d'utilisateurs mais avant la vérification du login et du mot de passe, utilisez **session_start()** pour démarrer une session. Puis vérifiez s'il existe ou non une valeur \$_SESSION[« user »] avec **isset**. Si la variable existe et a une valeur correcte, initialisez \$identified_user.

Changez votre test sur les variables login et mot de passe pour qu'il ne soit effectué que si \$identified_user n'est pas déjà initialisé.

Enfin, lorsqu'un login et un mot de passe corrects sont envoyés, initialisez la valeur de \$_SESSION[« user »].

Troisième étape : log out

En bas de page (dans le fichier footer.php, avant </body>). Créez un lien vers index.php?deconnection=1. Ajoutez le code nécessaire dans settings.php pour supprimer la session avec **session_destroy** dans le cas où la déconnexion a été demandée.

Votre site est maintenant fonctionnel

vous pouvez améliorer son aspect visuel à l'aide d'un fichier CSS que vous appellerez dans le header html.