

Compte Rendu TP MySQL

0 – Introduction

A travers les quelques séances que nous avons passé à travailler sur le PHP et sur les bases de données, nous avons eu pour objectif la réalisation d'un formulaire sous forme de page internet, où des personnes pourraient renseigner des informations les concernant. L'objectif étant par la suite de récupérer ces données dans une base de données MySQL afin de pouvoir les traiter et les utiliser. Pour ce faire, nous avons utilisé l'éditeur HTML 'TextWrangler', le navigateur internet 'Firefox', le logiciel de transfert FTP 'Fetch', ainsi qu'un compte Free permettant de stocker jusqu'à 10Go de données, et gérant les bases de données MySQL.

L'adresse de ce site perso est : <http://romaricche04.free.fr/cherbonr/>.

1 - Index.html

Dans un premier temps, nous allons créer un répertoire appelé 'cherbonr' où nous allons placer tout les fichiers et dossiers relatifs au site internet. Puis, afin de pouvoir afficher une page lorsque nous nous connectons au site, nous créons un fichier 'index.html' qui sera affiché par défaut à l'adresse url suivante : <http://romaricche04.free.fr/cherbonr/>.

Dans ce fichier html, nous allons retrouver la structure classique composée du <HEAD> puis du <BODY>. Nous insérons un titre dans le <HEAD> à l'aide de la balise <TITLE>, ainsi qu'une balise <META> qui indiquera le type du fichier. Le tout étant placé à l'intérieur d'une balise <HTML>. Enfin, afin de vérifier le bon fonctionnement du fichier, on insert un petit texte dans le <BODY>.

2 – Autoindex, .htaccess, et .htpasswd

Une fois cela réalisé, nous avons créé un dossier 'autoindex' placé à la racine du site, où nous avons placé deux fichiers TXT : HEADER.txt et README.txt. Le but est ici de pouvoir accéder aux fichiers placés dans ce répertoire, sans passer par une page HTML. Ainsi le HEADER.txt permet d'afficher un petit texte en entête, tandis que le README.txt s'affiche comme un commentaire sous la liste des dossiers et fichiers.

De plus afin de protéger l'accès à cette partie du site, nous avons créé un fichier .htaccess que nous avons placé dans ce répertoire. Ce fichier comporte les instructions suivantes :

```
ErrorDocument 404 cherbonr/http/erreur.html
PerlSetVar AuthFile cherbonr/http/interdit/.htpasswd
AuthName "Acces restreint dans cette zone : montrez votre patte blanche..."
AuthType Basic
require valid-user
```

La première ligne renvoi à la page erreur.html dans le cas où l'adresse url rentrée ne serait pas correcte. La seconde ligne quant à elle indique où se trouve la liste des utilisateurs autorisés, ainsi que les mots de passe associés à chacun d'entre eux. Enfin les autres lignes indiquent le type de sécurité mis en place pour cette page.

Les utilisateurs recensés et autorisés à accéder à cette partie du site sont indiqués dans le .htpasswd qui se trouve lui dans le dossier cherbonr\http\interdit\ :

```
romaric:bernard
bfricottin:clochett
```

La mise en forme dans ce fichier étant la suivante : nom_utilisateur:mot_de_passe
Enfin, afin de protéger ce fichier, et que personne ne puisse le voir ni le modifier, on place dans ce même répertoire le .htaccess suivant :

```
deny from all
```

Enfin, dans un souci de compatibilité des encodages UTF-8, on place un .htaccess à la racine du site, dans le dossier 'cherbonr' où l'on insert les lignes suivantes :

```
adddefaultcharset utf-8
errordocument 404 cherbonr/http/erreur.php
```

2 – Erreur.php

Comme nous l'avons vu précédemment, nous avons utiliser une page d'erreur personnalisée en cas de problème rencontré par le serveur (page non trouvée, erreur d'authentification...). Nous allons maintenant voir comment cette page située dans le répertoire cherbonr\http\ a été construite.

Dans un premier temps, on remarque que c'est une page html des plus basique avec un <HEAD> et un <BODY>. Mais intéressons nous plus particulièrement au <BODY> :

Tout d'abord, on a inséré un « FATAL ERROR » pour faire peur à l'utilisateur, qui est la seule partie statique de la page, le reste étant en PHP.

La première chose à remarquer est la façon dont on signale le passage au PHP, en effet pour cela il faut insérer la commande suivante : '<?php' le 'php' étant facultatif, et pour terminer, insérer celle-ci : '?>'.

Concentrons nous à présent sur le contenu de cette partie en PHP. Nous avons d'abord appelé la fonction 'echo' qui permet l'affichage d'un texte (ici « Coucou c'est moi »), puis on la termine par un ';' . On notera la présence du '\n' qui applique un retour à la ligne. Ensuite, on dresse une liste non exhaustive d'erreurs possibles en leur attribuant le numéro correspondant grâce à la fonction 'array' :

```
$erreur=array(
    '404' => "Page non trouvée",
    '401' => "Page non autorisée",
    '500' => "Erreur serveur",
);
```

et on en appelle une par la fonction 'echo' (ici la 401). Ensuite on va afficher l'erreur retournée par le serveur grâce à la ligne suivante :

```
echo "<p>Le message d'erreur est :
".$erreur[$_SERVER['REDIRECT_STATUS']]."</p>\n";
```

Par la suite, on va afficher toutes les erreurs listées précédemment, grâce à la fonction 'echo' couplée à une boucle 'foreach'. Enfin, afin d'avoir une vision globale des informations retournées par le serveur, nous allons créer une table dans laquelle on va afficher ces informations :

```
<table>

<?php
    foreach($_SERVER as $k => $v){
        echo "<tr><td><tt>$k</tt></td><td><i>$v</i></td></tr>\n";
    }
?>
</table>
```

La page d'erreur ainsi terminée (mais non réaliste), nous allons à présent nous intéresser à la création d'un formulaire en ligne.

3 – Form.php

Pour cette page HTML, nous allons créer un <HEAD>, un <BODY>, ainsi que les divisions suivantes 'container', 'logo', 'titre', 'cadre', et 'footer'. Pour commencer, intéressons nous au <HEAD>, qui comporte une balise <META> permettant de définir la racine du site (ici <http://romaricche04.free.fr/cherbonr>). On trouve ensuite une balise <TITLE> classique, et une balise <LINK> permettant de définir une feuille de style .css et de la localiser.

Le <BODY> contiendra la division 'container' dans laquelle on insèrera la division 'logo', 'titre', 'cadre', et 'footer'. Dans 'logo' nous allons mettre un lien vers cette page afin de la recharger. Le 'titre' quant à lui contient comme son nom l'indique le titre (« Formulaire d'inscription »). Enfin le 'cadre' contiendra le corps du formulaire. Nous allons commencer

par créer un champ 'fieldset' avec une 'legend', dans lequel nous allons insérer une table avec plusieurs champs à remplir. Tout d'abord, une liste déroulante pour le Titre avec 3 valeurs (Madame, Mademoiselle, Monsieur), puis des zones de texte pour le Nom, Prénom, Adresse, Adresse2, Ville, et Code Postal, des boutons de type radio pour l'Entreprise, et enfin un 'textarea' pour la Description. Nous créons finalement un bouton 'Envoyer' pour soumettre les valeurs.

Il va maintenant falloir envoyer ces informations au serveur de base de données MySQL Free. Pour cela, nous exécutons la fonction PHP 'require_once' qui permet de récupérer les informations de connexion contenues dans le fichier 'connexion.php' qui, étant donnée le caractère confidentiel des informations contenues, est placé dans le répertoire 'cherbonn\http\interdit\'. Ce fichier recense les paramètres de connexion à <http://phpadmin.free.fr/>, ainsi que le nom de la base de données et du codage des caractères. De plus, grâce à 'mysql_query' on va insérer les valeurs récupérées avec la méthode POST, dans les champs de la base de données.

4 – Style.css

Afin de donner un design agréable à la page du formulaire, nous avons créé une feuille de style avec un style de police propre à chaque division, des images d'arrière plan, ainsi qu'une disposition adaptée.

5 – <http://myphpadmin.free.fr/>

Afin de récupérer, trier, et utiliser les données récoltées, il faut créer une table de base de données sur le site '<http://myphpadmin.free.fr/>'.











































































Nous allons donc créer un champ par information recueillie en attribuant un type de donnée (Texte, Nombre,...) à chacun, ainsi que la taille en octets à réserver.

Serveur: romaricche04.sql.free.fr ▶ Base de données: romaricche04 ▶ Table: cherbonnel_09_inscript

Afficher Structure SQL Rechercher Insérer Exporter Importer Opérations Vider

Supprimer

Champ	Type	Interclassement	Attributs	Null	Défaut	Extra	Action
<input type="checkbox"/> id	int(11)			Non	aucune	auto_increment	     
<input type="checkbox"/> date	datetime			Non	aucune		     
<input type="checkbox"/> Titre	enum('Mlle','Mme','Mr')	utf8_general_ci		Oui	NULL		     
<input type="checkbox"/> Nom	varchar(30)	utf8_general_ci		Oui	NULL		     
<input type="checkbox"/> Prenom	varchar(30)	utf8_general_ci		Oui	NULL		     
<input type="checkbox"/> Adresse	varchar(50)	utf8_general_ci		Oui	NULL		     
<input type="checkbox"/> Adresse2	varchar(50)	utf8_general_ci		Oui	NULL		     
<input type="checkbox"/> Ville	varchar(20)	utf8_general_ci		Oui	NULL		     
<input type="checkbox"/> CodePostal	int(5)			Oui	NULL		     
<input type="checkbox"/> Entreprise	enum('Pagora','CTP')	utf8_general_ci		Oui	NULL		     
<input type="checkbox"/> Description	text	utf8_general_ci		Oui	NULL		     

↑ Tout cocher / Tout décocher Pour la sélection :      









Version imprimable Suggérer des optimisations quant à la structure de la table ?




Ajouter 1 champ(s) En fin de table En début de table Après id Exécuter

+ Détails...

Ouvrir une nouvelle fenêtre phpMyAdmin

On pourra ensuite visualiser les informations recueillies auprès des différentes personnes inscrites dans le tableau suivant disponible sous l'onglet 'Affichage' :

	id	date	Titre	Nom	Prenom	Adresse	Adresse2	Ville	CodePostal	Entreprise	Description
<input type="checkbox"/>  	1	2009-11-13 09:14:21	Mr	Jean	Quentin	rue du Due	NULL	Grenoble	38000	Pagora	Gnagna
<input type="checkbox"/>  	3	2009-11-13 09:18:04	Mlle	Dougadoudou	Raïssa	quelque part	NULL	Bamako	0	NULL	NULL
<input type="checkbox"/>  	13	2009-11-13 10:58:25	Mr	cherbonnel	romaric	le bourg	centre	bion	50140	Pagora	youpi
<input type="checkbox"/>  	21	2009-11-13 11:15:58	Mr	Geek	Alan	computer	center	Taiwan	1010	Pagora	Geek pro

↑ Tout cocher / Tout décocher Pour la sélection :   

Afficher : 30 enregistrement(s) à partir de l'enregistrement n° 0

en mode horizontal et répéter les en-têtes à chaque groupe de 100

6 – Conclusion

Ce TP fut une bonne occasion de découvrir quelques rudiments de base de programmation en PHP, ainsi que de se remémorer les notions de HTML vues en deuxième année. De plus, la gestion de bases de données est une compétence utilisable dans plusieurs domaines industriels et personnels.