Université Bordeaux Ségalen

Conception de Sites Web Dynamiques : TD 6

Commentaires sur une page web avec PHP et MySQL

Une documentation de la création d'une base de données avec MySQL et EasyPHP et accessible sur http://www.labri.fr/~preuter/cswd2012/mysql.pdf.

Pièges à éviter tout au long des TD :

- Vérifiez que EasyPHP est bien démarré (pour avoir un serveur Apache).
- Les extensions des fichiers PHP doivent être toujours .php
- Tous vos fichiers PHP qui utilisent la base de données doivent inclure le fichier connect_db.php en tout début de page.
- Votre dossier de travail doit être dans P:_APP\www\cswd.
- Dans le navigateur, l'URL pour afficher vos fichiers .php doit toujours commencer par http:// et non pas par file://.

Exercice 6.1 Outils de travail

1. Reprenez votre "feuille de triche" et mettez-le à côté de vous. N'hésitez pas à le compléter sur une deuxième page pendant ce TD.

2. Reprenez votre feuille de TD 5 et mettez-le également à côté de vous.

3. Aujourd'hui, nous travaillons dans un dossier td6. N'oubliez pas que vos dossier de travail avec EasyPHP doivent être dans le dossier P:_APP\www\cswd.

Exercice 6.2 Création d'une base de données MySQL

Pour cet exercice, regarder le support du cours ¹. 1. Si vous ne l'avez pas encore fait lors de derniers TDs, créez une base de données MySQL qui s'appelle base_cswd.

Exercice 6.3 Création d'une table dans une base de données MySQL

1. Si vous ne l'avez pas encore fait lors de derniers TDs, créez une nouvelle table livreor dans la base de données base_cswd avec un champs INT qui s'appelle id avec les attributs clé primaire ainsi que "auto-increment", un champs TEXT qui s'appelle nom, et un autre champs TEXT qui s'appelle texte, ainsi qu'un champs DATETIME qui s'appelle date.

 $^{^{1} \}rm http://www.labri.fr/\sim~preuter/cswd2012/mysql.pdf$

Exercice 6.4 Pages web XHTML statique

Considérez la page web suivante :



Dans cette exercice, vous écrirez le contenu du fichier XHTML saisir-commentaire.php de cette page web, et plus particulièrement :

1. Le titre de la page (Commentez).

2. L'inclusion d'un ficher form.css qui permet de définir la mise en forme.

3. Le titre de niveau 3 "Saisir votre commentaire".

4. Un formulaire qui renvoie vers la page enregistrer-commentaire.php avec la méthode get.

5. Une table avec les étiquettes et les champs d'entrées du texte (appelé *nom*), d'une zone de texte (appelée *commentaire*) ainsi qu'un bouton qui envoie le formulaire.

Exercice 6.5 Evaluation d'un formulaire

1. Créer un fichier avec Notepad++ qui s'appelle enregistrer-commentaire.php, récupérez les variables *nom* et *commentaire* du formulaire et stockez les dans des variables **\$nom** et **\$commentaire**. Affichez le contenu de ces variables, chaque variable sur une ligne.

2. Imaginez qu'un utilisateur aurait entré le nom "Martin" et le commentaire "Site internet très ergonomique !". Dans ce cas là, l'affectation d'une variable **\$sql** contenant une requête pour insérer un enregistrement aurait été :

```
$sql = "INSERT INTO livreor ( id , nom , texte, date )
VALUES ('', 'Martin', 'Site internet très ergonomique !', NOW()); ";
```

Affectez la variable **\$sq1** pour que vous construisez une requête à partir des données du formulaire. Notez que vous devez faire des concatenations de chaînes de caractères (vous ne pouvez pas faire de changement de contextes).

3. Exécuter votre requête en utilisant l'instruction suivante :

\$resultat = qdb(\$sql);

Afin d'avoir accès à la fonction qdb, n'oubliez pas que tous vos fichiers qui utilisent la base de données doivent inclure le fichier connect_db.php en tout début de page avec

<?php require_once("connect_db.php"); ?>

4. Tester votre formulaire et vérifiez si votre table livreor contient bien un nouvel enregistrement correspondant aux données saisies.

5. Tester votre formulaire avec un commentaire qui contient le caractère ' (par exemple je l'aime).

6. Pourquoi vous obtenez un message d'erreur ? Afin de sécuriser vos formulaires, protegez ses données en échappant ses commentaires (après la récupération des données du formulaire, et avant la création de la requête SQL dans la variable **\$sql** avec les instructions suivantes :

```
$nom = mysql_escape_string($nom);
$commentaire = mysql_escape_string($commentaire);
```

Exercice 6.6 Affichage des commentaires

Comme déjà dans le TD 5, dans cet exercice, vous allez créer une page permettant d'afficher tous les commentaires.

1. Créer une page XHTML dans le fichier livreor.php qui permet d'afficher tous les enregistrement d'un livre d'or. Un exemple pour afficher tous les enregistrements (sans mise en forme) se trouve ci-dessous. Essayez cet exemple dans le corps de votre document XHTML. N'oubliez pas d'inclure le fichier connect_db.php en tout début de page.

```
<?php
// exécuter une requete MySQL
$sql = "SELECT * FROM livreor;";
$resultat = qdb($sql);
// connaitre le nombre d'enregistrements
$nombre_resultats = mysql_num_rows($resultat);
// parcourir le tableau des enregistrements
while ($enregistrements = mysql_fetch_array($resultat))
{
    echo $enregistrements['nom'];
    echo $enregistrements['texte'];
    echo $enregistrements['date'];
}
?>
```

2. Pour la mise en forme, utilisez des tables de 3 colonnes (nom, texte, date).

3. Créer un lien hypertexte vers la page qui permet de saisir un nouveau commentaire.

Exercice 6.7 Evaluation de commentaires

Dans cet exercice, nous allons donner la possibilité aux visiteurs de votre site de donner une évaluation sur les commentaires dans le livre d'or. 1. Dans votre table MySQL s'appelant livreor, rajouter un nouveau champs INT qui s'appelle evaluation. Ce champs réfletera par la suite le nombre d'avis positif moins le nombre d'avis négatifs.

2. Dans la table HTML de votre fichier livreor.php, rajoutez deux colonnes. Dans la première de ces colonnes, affichez la valeur de votre champs evaluation. Dans la deuxième, affichez deux liens hypertextes qui permettent de renvoyer vers cette même page livreor.php, en y passant en paramètres la variable positif (resp. negatif) avec comme valeur l'identifiant id du commentaire en question. Voici une illustration d'un aperçu de cet affichage.

Il y a 2 enregistrements dans la table.				
Fred	Dansles fichiers XHTML, toutes les balises s'écrivent en miniscules.	2012-03-09 09:43:01	0	<u>positif</u> <u>négatif</u>
Gaël	Il ne faut pas oublier d'inclure le fichier connect_db.php en tout début de page	2012-03-09 09:50:24	0	positif <u>négatif</u>

3. Au debut de votre fichier livreor.php, vérifiez si vous avez un paramètre GET de positif avec la fonction isset. Si oui, incrémentez le champs évaluation du commentaire en question de 1 en construisant la requête nécessaire. Notez que par exemple, pour incréménter le commentaire ayant l'identifiant 2, la requête serait :

\$sql = "UPDATE livreor SET evaluation=evaluation+1 WHERE id=2;";

4. Vérifiez maintenant le paramètre negatif, et décrementez l'évaluation respectivement.

5. Remplacez les textes "positif" et "négatif" par des images (par example avec une pouce pointant vers le haut / pointant vers le bas).

6. Affichez la valeur du champs évaluation en couleur. Pour une valeur positif, prenez vert. Pour une valeur négatif, prenez rouge. Pour cela, définissez 2 classes CSS dans la feuille de style, et appelez la bonne classe selon la valeur.

7. Trier l'affichage des commentaires par le champs evaluation par ordre décroissant.

Exercice 6.8 Fonctions de mise en forme de date

1. Etudiez la fonction PHP date() sur le manuel de PHP (http://www.php.net/manual/fr). Essayez de trouver une meilleure mise en forme pour la date du commentaire postée, de type "Lundi, 5 Mars, à 10h00". Pour cela, définissez les deux tableaux suivants :

Exercice 6.9 Site Web

Réserver un espace WEB chez un hebergeur **gratuit** ou alors chez votre fournisseur d'accès à Internet (FAI) qui supporte le langage **PHP** et qui vous autorise d'utiliser au moins une base de données **MySQL**. Un exemple est olympe-network.com, mais il y en a peut-être d'autres plus intéressants.