PHP-TP2

Structurer vos scripts de manière à avoir un répertoire par exercice. Déposez vos scripts bien commentés sur Claroline.

Remarque :
En plus des commentaires, votre code doit suivre au moins les 3 règles de bonnes pratiques suivantes :

Exercice 1
Dans de nombreuses applications, il apparaît rapidement qu'il est nécessaire de pouvoir identifier l'utilisateur. Pour cela, il est nécessaire de stocker des informations utilisateurs comme son login et son mot de passe.

1. Créer une table Personne contenant : Id (auto_increment), nom, mot de passe, et couleur. (choisir utf8 comme charset).
   Exemple Sur Workbench

```
CREATE TABLE Client (
 id int(11) NOT NULL,
      nom       varchar(25) DEFAULT NULL,
      prenom       varchar(25)  DEFAULT NULL,
      adresse       varchar(25)  DEFAULT NULL,
      tel       int(11) DEFAULT NULL,
      dateNaissance       varchar(25)  DEFAULT NULL,
      sexe       varchar(25)  DEFAULT NULL,
 PRIMARY KEY (       id       )
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;
```

2. Peupler la table (le mot de passe sera stocké en clair pour le moment). Dans nom, prenom et adresse mettez des accents et des caractères non latins.

Rappel : Le nom de votre BD est votre numéro d'étudiant (p123456789), vous pouvez y accédez dans PhPMyAdmin ou en utilisant MysqlWorkbench. Votre login est p123456789 et votre mot de passe est 1123456789. Vous pouvez y ajouter des tables, mais pas créer de nouvelle base.

3. Créez un fichier params.inc.php contenant les informations de connexion. Pour l'IUT, cela doit ressembler à :

```
<?php                    // params.inc.php
$host="localhost";
$user="p123456789";
$password="1123456789";
$dbname="p123456789";
?>
```

4. Écrivez une fonction PHP qui permet de vous connecter à la base de données. N'oubliez pas de gérer les erreurs.

5. Écrivez un script PHP qui affiche le contenu de la table Personne dans un tableau. Répondre à cette question en testant les différentes méthodes d'accès aux données ainsi qu'avec les différentes méthodes d'indexation vues en cours.

6. Avant d'aller plus loin, résoudre les problèmes éventuels de Charset. Si vos accents ne s'affichent pas correctement, il faut configurer le charset de la session en exécutant la commande SQL « SET NAMES utf8 » via votre script PHP (il faut également vérifier que le script est également en utf_8)

Exercice 2

Créez un formulaire HTML sur une page de login, qui demande à l'utilisateur son login et son mot de passe. Écrivez ensuite le script PHP qui traite le formulaire de login (traitelogin.php). Le script affiche
Bonjour login
Merci de patienter nous allons vérifier vos données
Ensuite, il effectue une vérification de l'existence du login puis vérification du mot de passe. Si l'utilisateur est reconnu, affichez un récapitulatif de ses données. Sinon, renvoyez-le à la page de login en lui indiquant qu'il s'est mal identifié. S'il s'agit de l'administrateur, on affichera les informations de tous les utilisateurs enregistrés (cf. exercice 1). Effectuez les modifications nécessaires pour qu'une fois l'utilisateur identifié, une session soit créée (son numéro sera affiché lors du récapitulatif de ses données).
Ajoutez un lien permettant à l'utilisateur de se déconnecter. Ce lien lancera logout.php (à écrire).

Exercice 3

Dans la base, modifiez par exemple le champ couleur pour un utilisateur en tapant exactement orange. Que se passe-t-il si vous faites afficher ces informations avec le script précédent ? Modifiez-le pour que cela ne s'affiche pas en italique.
Dans le formulaire, tapez maintenant un nom d'utilisateur existant, mais essayer de faire une injection SQL sur le mot de passe en tapant " ' OR 1 = '1 " ou ses variantes. Modifiez le code pour protéger les accès.
On visitera avec profit ces sites relativement accessibles sur la sécurité PHP
http://www.eric-couchelou.net/securite-securiser-son-site-sous-phpmysql
et http://fr.openclassrooms.com/informatique/cours/securite-php-securiser-les-flux-de-donnees

Exercice 4

Testez la sécurité de votre site avec l'inspecteur de code du navigateur. Regardez par exemple si vous pouvez modifier le formulaire, ajouter/supprimer des champs, ajouter supprimer des personnes dans la liste affichée par votre script « traiterlogin.php »

Vérifier également si l'utilisateur peut exécuter un script via votre formulaire. Exemple, lors de l'affichage « Bonjour login » que se passe-t-il si au lieu du login l'utilisateur saisi le code javascript suivant :

```
<script>alert('Il y a une faille XSS')</script>
```

Ceci est le principe des failles XSS qui peuvent être très dangereuses (vol d'information, vol de session, etc.)

https://openclassrooms.com/courses/protegez-vous-efficacement-contre-les-failles-web/

Exercice 5

Pour introduire un peu plus de sécurité, le mot de passe ne sera pas stocké en clair dans la base de données, mais sera "haché" avant d'être stocké. Modifiez alors la procédure d'identification.

Attention : Il faut comparer les mots de passe hachés ! Donc pensez bien à les modifier dans la base en modifiant vos Insert avec des valeurs hachées pour les mots de passe. Vous pouvez également le faire avec PHPMyAdmin en utilisant un générateur de hachage selon l'algorithme que vous allez utiliser (md5, sha1, ne sont plus recommandés, préférer BlowFish). Augmenter éventuellement la taille du champ pour stocker le mot de passe haché (255 caractères par exemple).

Exercice 6

- Reprendre le formulaire HTML du TP1 permettant à un nouvel utilisateur de s'inscrire, et de saisir toutes les informations relatives à la table *Personne*.
- Écrivez un script *inscription.php* qui sera appelé par le formulaire, et qui effectuera le traitement des données soumises. Plusieurs cas doivent être traités par le script : si le "nom", ou le "mot_de_passe" n'ont pas été saisis, un message d'erreur doit être affiché, ainsi qu'un lien pointant vers le formulaire. Dans le cas contraire, un récapitulatif des données saisies doit être affiché à l'utilisateur. Pensez à demander de saisir 2 fois le mot de passe pour parer à une éventuelle faute de frappe.
- Si tout est bon, créez l'utilisateur et enregistrez ces informations dans la table (n'oubliez pas de crypter le mot de passe avant).
- Comme précédemment, rajouter les protections nécessaires (voir addslashes, addcslashes() ou quote() de PDO))
- Modifiez le script *inscription*.php, de manière à automatiquement rediriger l'utilisateur vers le formulaire si les champs obligatoires n'ont pas été saisis.