

TP 4 : bases de données

I. Mise en place d'une base de données

L'objectif de cet exercice est de mettre en place une base de données qui sera utilisée dans les exercices suivants.

1. Lancez *phpMyAdmin* (une fois votre serveur Web démarré) et loguez-vous.
2. Créez une nouvelle base de données :
 - a. Choisissez un nom (tout en minuscules, pas de caractères spéciaux, pas d'espaces) : *tp4*, par exemple ;
 - b. Pour l'interclassement, choisissez *utf8_general_ci* (par exemple).
3. Créez une nouvelle table nommée *users* contenant 3 « colonnes ». Elle doit contenir les champs suivants :
 - a. « id » de type UNSIGNED INT, index « PRIMARY », auto-incrément (A_I)
 - b. « login » de type VARCHAR(100)
 - c. « password » de type VARCHAR(100)
4. Ajoutez plusieurs utilisateurs dans votre table (onglet « Insérer »).
5. Vérifiez le contenu de la table (onglet « Parcourir »).

Dans une application destinée à la production, les mots de passe doivent être chiffrés. Nous ne nous intéresserons pas à cette partie. En effet, les *frameworks* PHP (comme *Laravel*) intègrent déjà cette fonctionnalité.

II. La classe *MyPDO*

Comme vu en TD, nous allons écrire une classe *MyPDO* pour nous aider à créer et maintenir une connexion vers la base de données.

1. Créez un répertoire « *Config* » contenant un fichier nommé *DBConfig.php*
2. Dans ce fichier de configuration, spécifiez les éléments nécessaires pour se connecter à votre base de données :
 - DB_HOST : localhost
 - DB_NAME : le nom choisi dans l'exercice précédent (*tp4* ?)
 - DB_LOGIN et DB_PASSWORD : les identifiants utilisés pour se connecter à votre base
3. Écrivez un script PHP permettant d'afficher dans une page HTML le contenu de la table *users* :
 - a. Utilisez la méthode *getInstance* pour récupérer la connexion.
 - b. Exécutez une requête de type SELECT pour sélectionner les données.
 - c. Récupérez-les avec la méthode *fetch* et affichez-les dans une table HTML.
4. Que se passe-t-il lorsque vous saisissez l'URL *localhost/XXX/config/* (où XXX est le répertoire de votre projet) ?
5. Ajoutez le fichier nommé *.htaccess* et qui possède le contenu suivant :

```
DENY FROM ALL
```

6. Et maintenant, que se passe-t-il si vous essayez d'accéder au répertoire de configuration ?

Vous êtes maintenant prêt pour exploiter la base de données dans une application Web.

III. Connexion d'un utilisateur

Dans le TP précédent, nous avons développé une classe *User* pour gérer les utilisateurs. Les méthodes *connexion/deconnexion* permettent de gérer la connexion/déconnexion de l'utilisateur à l'aide d'une variable de session.

1. Reprenez vos scripts. Si ce n'est pas le cas, assurez-vous qu'ils fonctionnent bien.
2. Écrivez la classe *UserModel* qui contient les méthodes classiques d'une classe CRUD avec en plus la méthode *fromArray*.

3. À l'aide d'un script, vérifiez que toutes vos méthodes sont fonctionnelles.
4. Modifiez maintenant la méthode `check` de la classe `User` : au lieu d'utiliser les constantes de classe, elle exploite maintenant la base de données via la classe `UserModel` pour vérifier la connexion d'un utilisateur.

IV. Retour sur les actualités

Nous souhaitons écrire une application qui affiche des actualités. Une actualité est caractérisée par un identifiant unique, un titre, un message et une date. Elle est associée à un l'utilisateur qui l'a créée. Sur la page principale, l'ensemble des actualités est affiché. Un formulaire permet à l'utilisateur de se connecté (sauf s'il l'est déjà). Si l'utilisateur est connecté, un bouton permet d'accéder à une page qui permet d'ajouter une nouvelle actualité.

1. Proposez une base de données pour l'application.
2. Écrivez les classes nécessaires (ou récupérez-les de précédents exercices...).
3. Écrivez les modèles nécessaires.
4. Écrivez les scripts PHP pour l'application.