

## TD 2 - PHP

### Accès aux bases de données - Introduction de MVC

#### Objectif :

Utilisation de phpMyAdmin et PHP pour accéder à une base de données.

#### Exercices :

On souhaite utiliser une base de données pour personnaliser le mot d'accueil en fonction du login de l'utilisateur. Selon le login, le message d'accueil sera différent et répété un certain nombre de fois.

Login toto	Valider
<ul style="list-style-type: none"><li>• Hello toto</li><li>• Hello toto</li><li>• Hello toto</li><li>• Hello toto</li><li>• Hello toto</li><li>• Hello toto</li><li>• Hello toto</li></ul>	

Login titi	Valider
<ul style="list-style-type: none"><li>• Bonjour titi</li><li>• Bonjour titi</li><li>• Bonjour titi</li><li>• Bonjour titi</li><li>• Bonjour titi</li></ul>	

Login tata	Valider
tata : Login inconnu	

1. Utiliser phpMyAdmin pour créer la table Utilisateur (login, mot, nbRepet). nbRepet est un entier. login et mot sont des varchar, de taille maximale 50, avec utf8\_general\_ci comme valeur d'interclassement. Créer à partir de phpMyAdmin deux utilisateurs associés à des messages d'accueil différents et des nombres de répétitions différents également.
2. Dans helloworld5.php un champ de saisie permet à l'utilisateur de préciser son login. Le formulaire de la page pointe sur la page elle même. Le message d'accueil, formé à partir du mot dans la base de données et du login, s'affiche uniquement si le login existe dans la base de données. Dans le cas contraire, un message d'erreur s'affichera.
3. Récupérer le répertoire helloworld5 (helloworld5.zip). Visualiser les répertoires et les fichiers un par un, comprendre leur contenu et les compléter pour rendre le site fonctionnel :
  - a. Le répertoire config contient configuration.php, à compléter.
  - b. Les répertoires assets/css et assets/scripts contiennent des fichiers nécessaires pour utiliser le framework CSS Bootstrap. La compréhension de ces scripts n'est pas l'objet du module. Ils permettront juste de s'affranchir rapidement de la partie présentation du site pour se concentrer sur le PHP. Vous pourrez ajouter votre propre fichier CSS si nécessaire.
  - c. Le répertoire languages contient un fichier pour chaque langue proposée. Compléter le fichier languages/FR-fr.php.
  - d. Le fichier index.php à la racine du site charge le fichier de configuration et joue le rôle de routeur en choisissant le contrôleur selon la page demandée.
  - e. Le fichier controllers/404.php se contente d'appeler la vue views/404.php pour spécifier que la page demandée n'existe pas. Tester avec l'URL :

`http://.../helloworld5/index.php?page=toto`

- f. Toutes les vues doivent commencer par `include v_header.php` et terminer par `include v_footer.php`. Elles incluent `v_alert.php` si un message d'erreur doit être affiché.
- g. Le répertoire `lib` contiendra une bibliothèque de fonctions. Une fonction `choixAlert` permet de choisir le type d'alerte à afficher.
- h. Quand aucune `page` n'est précisée dans l'URL, on tombe sur le contrôleur `c_accueil.php` qui charge la vue `v_accueil.php`. Faites en sorte que cette vue affiche le formulaire qui sera traité par `index.php?page=hello`.
- i. Compléter le contrôleur `c_hello.php`, qui :
  - fera appel au modèle `m_hello.php` pour accéder à la base de données et extraire de celle-ci les données souhaitées
  - fera appel à la vue `v_hello.php` à compléter pour afficher les messages de bienvenue si l'utilisateur a bien été trouvé dans la base de données
  - affichera de nouveau le formulaire mais avec une alerte si l'utilisateur n'existe pas dans la base de données. Ce sera également le cas en cas de problème de connexion à la base de données ou si la requête se passe mal.