Projet PHP/MySQL

L'objectif de ce projet est de créer un site de *microblogging* intitulé Brèves de Couloirs, inspiré du très célèbre Twitter. L'application Web développée permettra ainsi de :

- saisir des messages courts (ne dépassant pas les 140 caractères) contenant éventuellement des hashtags (chaînes de caractères précédées par un #
- s'abonner à des utilisateurs pour voir leurs messages postés
- rechercher des utilisateurs ou des messages selon différentes caractéristiques

Le schéma relationnel que nous allons utiliser est le suivant :

```
UTILISATEUR(cu, nom, description, ville, image, pwd, datei)
MESSAGE(cm, texte, auteur#, datem)
HASHTAG(ch, hashtag)
HASHTAG_MESSAGE(ch#, cm#)
SUIVRE(csuiveur#, csuivi#)
AIME_MESSAGE(cm#,cu#)
```

Travail préliminaire : Mise en place de l'environnement de travail.

- Créez un répertoire `bdc` sous `C:/Program Files/Easy PHP/www`. Ce répertoire contiendra tous les fichiers de votre site Web (pages php et images).

Attention, l'environnement de travail EasyPHP impose que les fichiers PHP se trouvent sous `C:/Program Files/Easy PHP/www`. Les données sur le disque C: ne font cependant pas partie de votre compte et sont donc susceptibles d'être détruites d'une séance de TP sur l'autre. A la fin de chaque séance, vous sauvegarderez donc vos données sous votre espace personnel.

- Téléchargez une première version du site qui vous servira comme point de départ pour ce projet à l'adresse : http://www.irit.fr/~Yoann.Pitarch/Docs/L2/projet.zip

- Décompressez l'archive dans le répertoire `C:/Program Files/Easy PHP/www/bdc`. Un fichier README.txt présent dans l'archive décrit le rôle des différents répertoires présents dans l'archive.

- Via phpMyAdmin, créer la base `bdc`

- Exécutez les ordres SQL présents dans les fichiers `creabase.sql` et `inserbase.sql` (dans le répertoire `bd` de l'archive)

Vous utiliserez un éditeur de texte tel que Notepad++, Atom ou SublimeText pour éditer et créez les fichiers nécessaires à la réalisation du projet. Si vous êtes familiers avec d'autres éditeurs de HTML, CSS et PHP, vous pouvez bien sûr les utiliser.

EVALUATION PROJET

Le barème d'évaluation prend en compte la quantité et la qualité du travail.

- La quantité du travail peut être quantifiée en termes de TPs ou sous-parties TPs terminées.

- La qualité du travail est évaluée sur la somme de plusieurs critères tels que :

    - Qualité du code HTML
    - Qualité du code CSS
    - Bonne séparation du contenu et du style
    - Bon fonctionnement des sous-parties du projet
    - Bonne compréhension du sujet
    - Tests correctement effectués

Pour avoir la note maximale, je vous conseille de :

- Résoudre toutes les parties obligatoires correctement
- Résoudre les parties optionnelles
- Faire du code bien structuré et facile à lire
- Prouver en séance TP que le travail vous appartient

Je vous conseille de travailler en binôme, mais ce n'est pas obligatoire.

Vous devez utiliser le navigateur Mozilla Firefox, Safari ou Chrome pour visualiser les pages Web, car le projet sera corrigé sur un de ces navigateurs. Vous préciserez lequel a été utilisé pour le projet.

TP #1 : Mise en place de la structure du site et création de la page d'accueil

L'archive téléchargée contient déjà une première version de la page d'accueil. Celle-ci est néanmoins incomplète. Il vous faudra rajouter les éléments suivants décrits ci-dessous.

- Vous veillerez à ce que les éléments suivants soient présents dans le menu et pointent vers les pages adéquates
  - Accueil : *accueil.php*
  - Mur : *mur.php*
  - Recherche :
    - Message : *chercheMessage.php*
    - Utilisateur : *chercheUtilisateur.php*
  - Profil : *profil.php*
  - Login : *login.php*
- Vous ajouterez, sous *les top hashtags*, la liste des 3 derniers utilisateurs inscrits
- Vous modifierez l'affichage d'un message de telle sorte à afficher la photo de l'utilisateur ayant émis le message et le nombre de personnes aimant le message. Vous pouvez également modifier l'affichage d'un message selon vos goûts.

Attention : La partie entête et pied de page (si présente) devra se retrouver sur toutes les pages que vous allez créer pour ce projet.

Quelques remarques :

Vous pouvez utiliser toutes les mises en forme que vous souhaitez, mais vos mises en forme doivent être effectuées par une feuille de style CSS.

Pour voir votre page, il suffit de l'ouvrir avec un navigateur web.

Evitez des couleurs trop fortes. Cela rend le contenu difficilement lisible.

Chaque page que vous allez créer devra être validée avec le validateur W3C: http://validator.w3.org/. **Attention :** Pour les pages php, vous devez ne devez valider que le code source généré côté navigateur (clic droite sur la page et ensuite trouvez l'option « visualiser code source »).

Vous pouvez utiliser le site http://www.colourco.de pour trouver des combinaisons de couleurs qui vont ensemble.

Dès que vous êtes satisfaits de l'aspect de votre page (qui servira en fait pour tout le site), préparez vous à la création des pages *php*. Prenez le code constituant l'entête de vos pages (minimum le titre, le sous-titre et menu) et copiez-le dans une page nommée *entete.php*. Si vous avez une partie pied de page, faites de même et créez une page nommée *pied.php*.
Ensuite créez les pages pointées dans le menu que vous avez créé. Chacune de ces pagese doit contenir l'entête et le pied de page que vous allez inclure à travers les instructions *php* suivantes :

```
< ?php  include("entete.php") ; ?>
```

```
< ?php  include("pied.php"); ?>
```

Notez que pour que l'affichage du menu diffère en fonction de la page consultée, il faudra écrire une fonction php qui prendra comme paramètre la page active et qui ajoutera la classe `active` à l'élément adéquat.

Liste de remarques importantes :

- Les pages php ne peuvent pas être ouvertes directement (ex. double click). Il faut passer par le serveur Apache qui va les interpréter et générer le code HTML correspondant. Pour visualiser les pages .php, il faut inclure dans l'URL l'adresse du serveur, le nom de l'application et le nom de la page. Par exemple, la page *mur.php* qui se trouve sur le serveur http://127.0.0.1/ sera lisible à travers l'adresse : http://127.0.0.1/bdc/mur.php. Sur le serveur vous pouvez également accéder aux pages statiques HTML (par ex. http://127.0.0.1/dpp/accueil.html).

- A partir de maintenant, on suppose que l'utilisateur de l'application est *Homer Simpson* (code profil 1). Il est logué en tant que tel et toutes les démarches qui seront effectuées le concernent.

- Faites attention aux images. Les images avec un chemin absolu ne s'ouvriront pas quand le projet sera corrigé. Utilisez le chemin relatif.

TP #2 : Dynamisons la page d'accueil

Maintenant que vous êtes contents de votre page d'accueil, il est temps de dynamiser son contenu en insérant les données appropriées dans les éléments de l'interface (top hashtags, dernier message et derniers utilisateurs inscrits).

1. Dans un premier temps, créez un fichier `connect.php` qui contiendra les données nécessaires à la connexion à la base de données.

2. Testez l'appel à ce fichier via `accueil.php`. Toujours dans cette page, prenez en compte un échec de l'essai de connexion et prévoyez un message d'erreur.

3. Créez un fichier f*onctions.php* qui contiendra toutes vos fonctions utilisateurs.
Ecrivez les fonctions :

- `topHashtags` qui affiche les 3 hashtags les plus utilisés
- `dernierMessage` qui affiche le dernier message émis par un utilisateur suivi par Homer
- `derniersUtilisateurs` qui affiche les 3 utilisateurs les plus récents (selon la date d'inscription)

TP# 3: Le mur

Etape 1 : Contenu et affichage

En supposant que l'utilisateur de notre application s'appelle « Homer Simpson » (profil numéro 1), on veut modifier la page `mur.php` pour qu'elle n'affiche que les messages créés par cet utilisateur ou que cet utilisateur suit. Les messages doivent être triés par ordre chronologique.

Créez une fonction `afficheMur` dans `fonctions.php` qui effectue ces affichages. Les histoires créées par cet utilisateur doivent contenir du texte du genre « vous avez partagé cette histoire ». Dans ce cas, il ne faut pas afficher le nom de l'utilisateur.

Etape 2 : Premier formulaire (partager un message)

Ajoutez, en haut du mur (sous le bandeau de titre), un formulaire qui permet à l'utilisateur d'ajouter un message comportant du texte. Faites-en sorte que le message soit enregistré dans la base de données avec la bonne date de création et comme auteur « Homer Simpson » (comme convenu). Vous veillerez aussi à extraire les hashtags du texte pour les relier correctement dans la base de données.

Le formulaire peut revenir sur la même page ou une autre page *php* qui nous confirmera si l'insertion est faite avec succès. Il faut interdire l'insertion d'un message sans texte.
Un peu d'aide :

- Comment peut-on s'assurer que le contenu ne dépassera pas la longueur maximale permise dans la base de données ? Eviter les problèmes en imposant le nombre maximal de caractères dans le champ de texte (attribut *maxlength*).

- Pour récupérer la date d'aujourd'hui et l'heure vous pouvez utilisez ce morceau de code *PHP* **:** `$date = date("Y-m-d H:i:s");`

Attention : Le texte avec des apostrophes ne sera pas inséré comme il faut dans la base de données. Il faut se rappeler que les apostrophes servent de délimitateurs en MySQL.

Conseil : Quand vos requêtes SQL générées automatiquement ne marchent pas, vous pouvez les afficher sur la page php afin de comprendre la cause de l'erreur. Une fois le problème résolu, vous devez enlever l'affichage.

TP #4: Recherche d'utilisateurs

Créez un formulaire dans la page `chercheUtilisateur.php` permettant d'effectuer une recherche des profils qui correspondent à des critères fournis par l'utilisateur. La recherche doit satisfaire tous les critères de recherche saisis.

Le formulaire de recherche contiendra

- Un champ texte servant à saisir le nom de la personne
- Une liste déroulante qui contient toutes les villes existant dans la base de données
- Une case à cocher pour indiquer si la recherche ne doit s'effectuer que sur les utilisateurs suivis par l'utilisateur (on suppose toujours que l'utilisateur est Homer Simpson)
- Un champ texte contenant des mots contenus dans la description de l'utilisateur recherché

Selon les cas proposez les profils qui correspondent aux personnes cherchées. Affichez les résultats dans un tableau HTML qui contient nom, ville, description et la photo de la personne concernée. Pour faire ceci, utilisez une fonction `afficherUtilisateur` que vous allez définir dans `fonctions.php`.

Des instructions à respecter :

- Si le profil n'a pas d'image associée, on affiche l'image NA.jpg du répertoire images.

- Si aucun critère de recherche n'est saisi, il faut retourner la liste de toutes les personnes.

<u>Questions subsidiaires :</u>

1. Modifiez l'affichage en ajoutant à chaque personne au maximum deux messages (les plus récents). Les messages doivent se retrouver dans le tableau qui contient les personnes.

2. Ajouter un formulaire par personne pour permettre d'ajouter/supprimer la personne de la liste des personnes qu'il suit.

TP #5 : Recherche de messages

On veut proposer un formulaire qui permet de chercher les messages déjà existants dans la base de données selon des critères définis par l'utilisateur :

1. Créez dans la page `chercheMessage.php` un formulaire permettant à l'utilisateur de chercher des histoires en fonction des critères :

    1. Mot X qui doit se retrouver dans le nom de l'auteur.
    2. Ville de l'auteur
    3. Phrase/mot *A* qui doit se retrouver dans l'histoire
    4. Hashtag qui doit être présent dans le message

Ce formulaire appelle la page `resultatChercheMessage.php` ou la page même `chercheMessage.php` qui affiche le résultat (liste de messages). Les messages retournés doivent rencontrer au moins un des critères (1, 2, 3 ou 4) si saisis. Les résultats doivent être triés selon la date de création (les plus récents apparaissent en premier).

2. Créez une fonction `afficheMessage` pour qu'elle affiche le message. L'affichage sera identique à celui du message le plus récent sur la page d'accueil.

3. Modifiez la fonction `afficheMessage` en ajoutant à chaque message affiché un formulaire qui permet à l'utilisateur d'aimer le message.

TP #6 : Profil

Nous souhaitons proposer, à travers la page `profil.php`, une interface pour visualiser les informations du profil de l'utilisateur. Ici encore, nous supposons que l'utilisateur est l'utilisateur 1 (Homer Simpson). Réaliser le code permettant de répondre à ce besoin.

Créez un formulaire sur la page `modifierProfil.php` qui permettra de modifier les informations du profil de l'utilisateur. Toutes les informations sont modifiables sauf le nom de l'utilisateur qui lui est fixe. Ce formulaire appelle la page `resultatModifierProfil.php` ou la page même `modifierProfil.php` pour indiquer si les informations du profil ont bien été enregistrées dans la base de données.

BONUS (2 points de plus sur projet fini)

Si vous avez fini toutes les questions du projet y compris les questions subsidiaires, vous pouvez demander une partie supplémentaire de projet qui vous permettra de gagner 2 points de plus et d'être notés sur 22.

Voici un exemple de tâches que vous pouvez traiter comme bonus. D'autres sont possibles et devront être définies en accord avec l'enseignant.

TP Bonus : Gestion de la connexion

Proposez une solution dans la page `login.php` pour que seule une personne disposant du bon mot de passe (saisi au travers d'un formulaire) puisse accéder au site.

Proposez un formulaire qui demande :

- Le nom de la personne
- Le mot de passe

Le but est de permettre l'accès à certaine partie du site à travers une authentification de l'utilisateur. En autres termes, l'utilisateur doit saisir les bons nom et mot de passe. Les pages lues depuis la page de login doivent s'adapter au nouvel utilisateur. On rappelle que jusqu'à ici on supposait que l'utilisateur était Homer Simpson. Ce n'est plus toujours le cas. Homer Simpson restera l'utilisateur par défaut sauf si on accède aux pages à travers l'authentification. Deux solutions sont possibles avec les connaissances que vous possédez:

Solution 1 :

Proposez un bouton pour la soumission du formulaire. Le formulaire appelle la page *prive.php.* Si le mot de passe ne correspond pas à la bonne personne affichez un message d'erreur toute en re-proposant le formulaire d'authentification. Sinon proposez un menu qui contient des liens aux pages *mur.php, chercheMessage.php, chercheUtilisateur.php et profil.php.* Ces liens doivent être modifiés pour passer comme paramètre le code profil de l'utilisateur connecté (eg. *mur.php?cprofil=2*).

Solution 2 :

Le formulaire est accompagné avec plusieurs boutons

Bouton 1 : Accéder au mur
Bouton 2 : Accéder à la recherche de message
Bouton 3 : Accéder à la recherche d'utilisateurs
Bouton 4 : Accéder à l'affichage et à la modification du profil

Si l'utilisateur clique sur le bouton 1 et il a tapé le mauvais mot de passe on affiche un message d'erreur. Si le mot de passe est bon alors affichez pour la personne les messages qui doivent apparaitre sur son mur. Faites la même chose pour les trois autres boutons.

Attention : Si vous arrivez faire les deux solutions ou que vous trouvez d'autres solutions intéressantes, appelez les différentes solutions *login2.php, login3.php, ...* et envoyez-les toutes en précisant que vous avait fait plusieurs solutions.

Attention 2 : Dans la réalité, la gestion de l'authentification n'est pas gérée comme ça, mais à l'aide de sessions.

RENDU DE PROJET

Pour rendre le projet, vous devez compresser le contenu de votre projet dans un archive *.zip*. Le nom de l'archive contiendra nécessairement le nom des étudiants. L'archive doit être déposée la date limite qui vous sera donnée pendant les séances TP. Tout retard sera pénalisé au prorata du retard.

Faites attention :

- Faites attention aux images. Les images avec un chemin absolu ne s'ouvriront pas. Utilisez le chemin relatif.
- Le projet sera ouvert avec la base de données qui a le schéma donné au départ. Donc, évitez de changer le schéma. Les données de la base de données ne seront pas forcement les mêmes que les vôtres. Donc, testez bien vos fonctions et vos pages *php*.
- N'oubliez pas de mentionner les membres de votre groupe dans votre rendu.