**Programmation Web 420-03D-FX**

***Travail pratique #1 : PHP/MySQL***

***Prof.: Matthieu Hermet Hiver 2016***

***Ce travail peut être fait en équipe de 2 (fortement suggéré)***

**Date de remise du TP1**

Au plus tard : mercredi le **2 Mars 2016 avant 18:00**

**Pondération et Conditions de Réalisation**

Ce travail compte pour 16% de la note finale. Tous les documents produits dans le cadre de ce travail doivent s'afficher correctement avec les deux navigateurs Web **Mozilla Firefox** et **Chrome**.

**Objectif**

Vous avez la charge de l’administration d’un site de partage de films (Nous sommes installés au Panama). Des utilisateurs identifiés (pseudo et mot de passe) peuvent consulter une page affichant la liste des films disponibles. Un ou des administrateurs peuvent téléverser des descriptions de films. À vous d’implémenter ce site, qui contient les pages suivantes :

1. Connexion : nom d’utilisateur + mot de passe
   1. Route vers la page principale si l’utilisateur est reconnu
   2. Offre un lien vers un sign-in si l’utilisateur est inconnu
2. Page principale : Depuis Connexion :
   1. Si le signataire est un utilisateur :
      1. Version de la page Consultation
   2. Si le signataire est administrateur :
      1. Version de la page Admin : l’admin peut ajouter ou supprimer des films.
3. Page Film : présente un film.

Note : l’hébergement des images se fait en local, dans un sous-dossier du site. Voir : <http://www.w3schools.com/php/php_file_upload.asp>. Seul le nom et le chemin de l’image passent en base de données.

**Directives générales**

1. Un script de base de données est fourni. Vous devez vous créer une base de données nommée **tp1\_nom1\_nom2** avant d’importer le script et utiliser l’interclassement « utf8\_general\_ci ». Vous avez le droit de la modifier si vous le désirez mais faites attention avant de vous aventurer en terrain inconnu!

2. Les contrôles des formulaires doivent être correctement **alignés** et les étiquettes doivent être **liées** à leur contrôle respectif.

3. Toutes vos pages doivent contenir au minimum **5 parties** : page, entête, menu, contenu et pied-de-page.

Le menu doit être créé à l’aide d’une liste : il ne contient que 2 liens ! Page de connexion et page principale.

Pensez à utiliser des fichiers séparés pour les différentes sections d’une page, tel que vu en exemple au cours 4.

4. La **section** de contenu de chaque page doit débuter par un titre de niveau 1.

5. Le titre (*title*) de chaque page doit être complet et pertinent.

6. Toutes les données envoyées par formulaire doivent être **validées** du côté **serveur** de façon précise comme vu en classe. Les champs doivent être conservés en PHP. Vous devez prévoir toutes les validations standards.

7. Vous devez utiliser, lorsque nécessaire, les fonctions qui feront en sorte que les utilisateurs ne pourront écrire du code HTML ou JavaScript dans les contrôles des formulaires.

8. Dans les cas où on ajoute, modifie ou supprime un enregistrement dans une table, vous devez faire apparaître un **message de confirmation** indiquant si l’opération a réussi ou non.

9. Vous devez créer un fichier **CSS pour une présentation graphique uniforme et professionnelle** : pas de bannière, mais un bon zonage spatial (en-tête, menu, contenu, pied-de-page), et une coloration d’arrière-plan.

10. Validez que le schéma (« *outlining* ») de vos pages est correct en utilisant le module complémentaire de Firefox *HTML 5 Outliner*.

11. Vous devez aussi respecter les **standards suivants**:

**HTML 5 Polyglot** (valider avec *Total Validator*)

**CSS** (http://jigsaw.w3.org/css-validator/#validate\_by\_upload)

12. Pour la connexion à la base de données MySQL et pour réaliser les différentes opérations sur vos tables, vous devez utiliser l’extension **PDO** telle que vue en classe. La **connexion à la BD** devra toujours se faire à l’aide du **fichier externe** « param\_bd.inc » (voir plus bas) non accessible par l’ensemble des internautes.

13. Pour les requêtes, les insertions, les mises à jour et les suppressions dans vos tables, vous devez utiliser les **requêtes préparées avec marqueurs nominatifs**.

14. Vous devez prévoir une **page 404 pertinente** et la rendre opérationnelle.

15. Les mots de passe doivent être **encryptés.**

16. Vous devez créer des **fichiers** et/ou des **fonctions** au besoin.

**Directives concernant la base de données**

17. Dans **phpMyAdmin**, vous devez créer un nouvel utilisateur qui servira pour la connexion à votre base de données (et à la correction). Il doit avoir les caractéristiques suivantes :

Utilisateur : garneau;

Serveur : localhost;

Mot de passe : qwerty123;

Base de données pour cet utilisateur : aucune;

Privilèges globaux : choisir les options pertinentes;

Limites des ressources : 0;

18. Un modèle de la base de données vous est fourni en annexe.

19. De plus, vous devez créer un **fichier externe au site** « param\_bd.inc » contenant les paramètres de connexion que vous devrez inclure dans toutes les pages où vous devez vous connecter à la base de données. Ce fichier doit inclure les paramètres suivants :

$dbHote = "localhost";

$dbUtilisateur = "garneau";

$dbMotPasse = "qwerty123";

$dbNom = "tp1\_nom1\_nom2".

**Directives concernant les pages PHP**

20. On doit pouvoir se connecter et se déconnecter au besoin de façon conviviale. On doit avoir la possibilité de **conserver le « login »** si on le désire.

21. Les éléments suivants doivent être présents :

Authentification des clients (sécurité)

o Le formulaire de connexion doit être une page complète (et non pas juste dans le coin droit) afin de vous permettre de bien **séparer vos pages sécurisées SSL** de celles qui ne le sont pas!

o On doit pouvoir se connecter et se déconnecter au besoin de façon conviviale.

Il y a 2 types d’usagers connectés en plus d’un utilisateur non connecté : *admin* et *membre*.

Seule la section Contenu (en page principale) changera, selon l’option sélectionnée dans le menu. Vous pouvez utiliser l’url afin de transmettre la page ou la section choisie (ex. index.php?page=2).

La section Contenu devra être générée dynamiquement à partir de la base de données selon l’état de la liste des films.

Vous devez **construire une section administrative** qui sera accessible seulement lorsqu’on s’est connecté correctement. Le contenu de l’administrateur sera donc différent du contenu standard. Cette section devra contenir les pages suivantes:

o Gérer le contenu :

Doit permettre d’ajouter, modifier et supprimer des pages de contenu (descriptions de films). N’oubliez pas qu’on doit pouvoir ajouter des images et donc qu’on doit les transférer sur le serveur et les redimensionner physiquement selon qu’elles sont affichées en page principale ou en page de description spécifique (stocker une image pleine grandeur et diminuer la taille en html si besoin est – pas l’inverse).

o Gérer les usagers :

Doit permettre d’ajouter, modifier et supprimer des usagers. Un usager a simplement besoin d’un nom d’utilisateur (« *login* ») et d’un mot de passe. Le mot de passe sera conservé encrypté dans la base de données.

Les pages de formulaire seront « hard-codée » donc leur contenu ne sera pas dans la base de données.

**Directives concernant le code PHP et HTML**

23. N'utilisez **pas d'espace, de majuscules, de symboles ni d'accent** dans les noms de fichiers et de dossiers. Nommez vos fichiers avec des noms significatifs. Séparer les mots de vos noms de fichiers par **le trait d'union**.

24. Vous devez créer des **fichiers** et/ou des **fonctions** au besoin et précéder ces dernières d’un **entête** de fonction standard.

25. Vous devez créer des dossiers et sous-dossiers pour bien **structurer votre projet** (exemple : mettez vos fonctions et/ou fichiers utilitaires dans un dossier « includes »).

26. Vous devez programmer une (des) **classe(s) de gestion** pour les accès à la base de données.

27. Vous devez commenter les fonctions ainsi que le code qui demande un peu plus de réflexion et/ou d’explications.

28. Les **noms** donnés aux champs des formulaires, aux fonctions et variables doivent être **significatifs** et en **français**.

29. Vous devez, bien sûr, respecter les **règles de base de la programmation** (décalage des blocs, utilisation des instructions appropriées, etc.)

30. Portez une attention particulière à la structure de votre programme ... pensez à la *simplicité* avant tout !

**À remettre**

Exportez votre base de données (.sql). Enregistrez ce fichier dans le sous-dossier *bd* de votre application Web.

Copiez les fichiers de configuration de votre serveur dans le sous-dossier *config* de votre application Web.

Compressez le dossier contenant votre site puis déposez-le dans LÉA (section Travaux) ou envoyez-le-moi par MIO si c’est trop gros.

*Bon travail !*