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

***Travail pratique #2 : PHP/MySQL***

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

***Ce travail doit être fait en respect des équipes constituées pour le TP1***

**Date de remise du TP2**

Au plus tard : lundi le **11 Avril 2016 avant 11:00**

**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**

Votre travail consiste à améliorer votre site de CMS en ajoutant des fonctionnalités en JavaScript et en AJAX afin de rendre le site plus convivial. Vous y inclurez une visionneuse d’images, une validation des champs de saisie du formulaire (augmenté) ainsi qu’une fonctionnalité de recherche.

**Directives générales**

**LÉGENDE :**

**VERT-» Pascal**

**BLEU –» Alexandre**

**TURQUOISE –» Pas mal fait**

1. Dans le fichier "**index.php**" (c-à-d, dans le fichier d’accueil), vous devez ajouter une **visionneuse** qui mettra en évidence les derniers ajouts sur le site. La visionneuse devra répondre aux critères suivants :

Une liste sous forme de points ou de vignettes doit permettre d’accéder directement à l’image qui nous intéresse. Cette liste doit aussi mettre en évidence l’image qui est présentement affichée dans la visionneuse.

L’image affichée doit changer à toutes les x secondes (on doit avancer d'une image dans la liste en même temps que l’image change).

Lorsqu’on survole l’image affichée de la visionneuse, cela doit interrompre le défilement automatique des images. Lorsqu’on ne survole plus l’image, le défilement automatique doit redémarrer.

Lorsqu’on clique sur l’image affichée de la visionneuse, cela doit changer de page et afficher la page concernée par l’item courant dans la visionneuse.

Vous pouvez utiliser les fonctions utilitaires contenues dans les fichiers se trouvant dans les exemples et les exercices.

2. Vous devez modifier le formulaire d’inscription en y ajoutant les champs « date de naissance », « adresse », « numéro de téléphone » et « courriel ». **Vous modifierez la base de données en conséquence**, mais vous n’avez pas besoin de créer de nouvelle table. Il vous faudra également et surtout assurer une **validation côté client complète** sur le formulaire permettant l’inscription d’un nouvel usager.

Vous devez retirer les espaces blancs inutiles au début et à la fin des champs de saisie.

Vous devez vérifier immédiatement et sans recharger la page si **le pseudo d’utilisateur désiré est disponible** ou non.

Lorsqu'on clique sur le bouton d’inscription, on met en évidence (en utilisant une couleur d'arrière-plan différente) tous les champs erronés (incluant l'étiquette du champ) et on affiche un message d'erreur pertinent et précis (un résumé de toutes les erreurs) en haut des informations personnelles.

Immédiatement après avoir modifié un champ (une fois que le curseur n'est plus dans le champ), on remet l'arrière-plan original pour ce champ si la nouvelle valeur du champ est valide. Si l'utilisateur revient dans un champ correctement rempli et cause une erreur, on doit immédiatement afficher le champ comme étant en erreur.

Lorsque tous les champs ont été corrigés, on efface le message d'erreur; ce qui indique à l'utilisateur que l’inscription peut maintenant être envoyée.

Lorsque l’utilisateur remplit le formulaire pour la première fois, on ne doit pas afficher d'erreurs tant et aussi longtemps que l'utilisateur n'a pas cliqué sur le bouton d’inscription (sauf pour le pseudo).

3. Vous devez ajouter une **fenêtre de confirmation** pour les diverses suppressions de votre site : film et utilisateur.

4. Vous devez programmer en JavaScript en utilisant uniquement les fonctionnalités du langage ECMAScript et du DOM. Vous devez utiliser le standard **DOM-2** pour la gestion des événements. Vous devez utiliser le format de données **JSON** pour les données retournées en Ajax. Vous ne devez pas utiliser d’extensions propriétaires.

5. Vous devez respecter les directives relatives aux codes html et css définies pour le TP1 :

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

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.

La section de contenu de chaque page doit débuter par un titre de niveau 1.

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

**Directives concernant la fonction de Recherche**

**Vous devez ajouter un champ de recherche dans la page de gestion des usagers.**

Vous devez ajouter une liste déroulante **par catégories d’utilisateurs** (type d’accès, membre ou admin) qui s’affiche sous le champ recherche après qu’un utilisateur ait tapé un **minimum de 3 caractères** dans le champ de recherche. On doit pouvoir sélectionner une suggestion de la liste pour effectuer la recherche. Cette liste doit répondre aux critères suivants :

Les suggestions doivent s’afficher par catégories et en ordre alphabétique de nom d’utilisateur.

La catégorie du membre doit être affichée ainsi que le nom complet du membre et son login.

Un clic sur chacun des membres affichés dans la liste devra mener directement à la page de modification de ce membre.

Si l’utilisateur ne sélectionne pas un élément de la liste, mais clique sur le bouton « Rechercher », on doit alors afficher tous les résultats de recherche dans la page.

**Directives concernant le code JavaScript**

1. Vous devez décomposer votre code **JavaScript** en plusieurs **petites fonctions** réalisant un travail précis et utilisant des paramètres lorsque possible. Ajoutez des variables globales seulement si nécessaire.

2. Vous devez **commenter** en détails le **code JavaScript** en spécifiant clairement, entre autres, le rôle de chaque fonction et de ces paramètres ainsi que des structures conditionnelles et des boucles.

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

4. 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**.

5. Vos fichiers doivent être organisés efficacement selon une arborescence Web comme expliqué en classe.

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

7. Portez une attention particulière à la structure de votre programme ... pensez à l’efficacité de votre code et 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 !*