

# **Évaluation pratique finale PHP / SQL**

**Temps imparti : 4h00**

## Notation

A la fin du temps imparti, vous devrez livrer l'intégralité de vos fichiers, et une note sur 20 sera attribuée selon le barème suivant :

- **4 points** pour l'exercice 1
- **4 points** pour l'exercice 2
- **6 points** pour l'exercice 3
- **4 points** pour l'exercice 4
- **2 points** sur les critères suivants :
  - Indentation du code
  - Commentaires
  - Pertinence du nom des variables
  - Lisibilité

**Bonus** : des points de bonus pourront être ajoutés si les fonctionnalités facultatives ont le temps d'être effectuées. A faire que si on a le temps !

## Conseils

- Lisez tout l'énoncé dès le début, pour savoir où vous allez.
- Prenez le temps de coder, en commentant votre code dès qu'il est nécessaire, pourquoi pas en copiant la consigne en commentaire.  
Il faut que le correcteur puisse comprendre ce que vous avez fait !
- Gardez des fonctions simples, qui ne font qu'une seule chose, pour mieux vous y retrouver.

## Fichiers :

Chacun des exercices font appels à des fichiers qui vous sont fournis au préalable. Ces fichiers sont classés par exercices et sont téléchargeables à cette adresse :

<http://www.wf3.fr/evaluations/finales/php/exercices.zip>

## Exercice 1 (4 points) : Le stagiaire

Un stagiaire, dont vous êtes responsable, doit récupérer une liste d'utilisateurs depuis une base de données mais également donner la possibilité d'en ajouter à l'aide d'un formulaire classique.

Le script principal se compose de deux parties :

- La liste des utilisateurs et leurs informations
- Le formulaire d'ajout d'un nouvel utilisateur

Le stagiaire ne fait de la programmation PHP que depuis peu de temps, il n'indente et ne commente pas son code. En tant que maître de stage, vous devez l'aider à corriger et faire fonctionner son code. Une bonne indentation du code est également nécessaire.

### Étapes de réalisation :

1. Déployez les fichiers *index.php* et *connect.php* sur votre serveur local en n'oubliant pas d'importer le script SQL *users-install.sql*. Celui-ci créera une nouvelle base de données contenant une table "users".
2. Aidez votre stagiaire à déboguer son code, corriger ses erreurs et le faire fonctionner.

## **Exercice 2 (4 points) : Authentification**

### **Phase 1 : l'inscription**

Un formulaire HTML d'inscription vous est fourni (inscription.html), comportant les champs nom, prénom, email et type (élève ou formateur).

Vous devez gérer l'enregistrement des inscriptions dans une table que vous aurez préalablement créée après avoir vérifié la validité des données :

1. tous les champs sont requis;
2. l'email devra être conforme et unique. Un utilisateur ne peut pas s'inscrire si le mail renseigné existe déjà.
3. Par ailleurs, le mot de passe devra être crypté à l'enregistrement.

### **Phase 2 : la connexion**

Vous devez gérer la connexion (formulaire HTML fourni, login.html) et déterminer le type d'utilisateur si ce dernier existe.

1. Si c'est un formateur : après s'être authentifié il doit être redirigé vers la page <http://www.wf3.fr/lecole/lequipe-pedagogique>.
2. Si c'est un élève : après s'être authentifié il doit être redirigé vers le site <http://wf3.apolearn.com/>

## Exercice 3 (6 points) : Ciné Phil

Une association d'amateurs de films vous demande de développer une page web sur son intranet.

Un membre de l'association accèdera à cette page, qui lui permettra de comptabiliser les films les plus visionnés.

La page se compose de deux parties :

1. Une liste de films, associés chacun à un bouton.
2. Un tableau affichant les 10 films les plus visionnés, mis à jour à chaque clic sur une des boutons.

Le formulaire, ainsi que l'affichage de l'ensemble des films, est déjà conçu : voir le fichier *movies-form.php*

Pour le moment, la page affiche la liste des films et soumet le formulaire à chaque clic sur un bouton.

*Note : La table 'genres' ne sert que pour les questions bonus. Vous pouvez l'ignorer pour le moment*

### Étapes de réalisation :

1. Mettez en place sur votre serveur les fichiers *form-movies.php*, *connect.php* et *functions.php* et importez la base de données *movies* depuis le fichier *movies.sql*
2. Affichez, dans la section *#most-viewed*, un tableau des 10 films les plus visionnés, du plus vu au moins vu. On affichera :
  - a. le titre du film (*title*)
  - b. la date de sortie (*release\_date*)
  - c. le nombre de vues (*nb\_viewers*).

*Note : Le nombre de vues est déjà précisé en base, pour vous permettre de tester votre code*

3. Faire en sorte que, si plusieurs films ont le même nombre d'entrées, le plus récent soit affiché en premier.
4. Mettre à jour le tableau à chaque clic sur un bouton.
5. Les genres des films sont précisés dans la table *genres*. Ajoutez, au-dessus du tableau, un bouton pour chaque genre de film. Chacun de ces boutons affichera le tableau des 5 films les plus visionnés de son genre. Ce nouveau tableau remplacera le tableau global.

### Bonus

Ajoutez un bouton pour réinitialiser les statistiques (passer tous les films à zéro vues).

## Exercice 4 (4 points) : Bibliothèque

Vous travaillez actuellement à la réalisation d'un site pour une bibliothèque en ligne. Votre collègue a déjà commencé à coder, et vous demande d'ajouter, à l'aide de son code, des livres dans la base de données. Il aurait besoin de 200 livres bidons dans la table "books" pour pouvoir continuer à avancer.

Pour ne pas le froisser, vous devez impérativement utiliser le code qu'il vous a fourni. Voici tout ce que vous en savez :

- La classe *Book* est utile pour créer des instances (vides) de livre. C'est le modèle de données.
- La classe *RandomBookDataGenerator* est utile pour générer des données de test (des chaînes, des noms, des date, etc.)
- La classe *BookManager* permet de réaliser les requêtes à la base de données, sur la table "books"

### Étapes de réalisation :

1. Mettez en place sur votre serveur le fichiers *insert\_books* et le dossier *Library*.
2. Créer 200 instances de la classe *Book* (une boucle peut être utile)
3. Hydrater chacune de ces instances avec des données aléatoires de test
4. Insérer chacune de ces instances en base de données

Vous devez réaliser votre code dans le fichier "insert\_books.php", et ne pas toucher aux autres fichiers !