

# Livrables Fil Rouge

## Module Programmation de l'interface

### Seconde partie : PHP, JavaScript, HTML et CSS

#### Développeur Logiciel

### *Objectifs : Création d'une application Web avec accès à une base de données*

## Projet BEDETHEQUE

Utiliser les fichiers **structure\_cataloguebd.sql** et **donnees\_cataloguebd.sql** ainsi que le répertoire **/images/**.  
On considère que toutes les images sont référencées dans la table **bandesdessinees**.

### **Partie public**

- Créer une page affichant la liste de BD (auteur/titre/couverture avec une image miniaturisée).
- Ajouter à cette page des liens vers une nouvelle page qui affiche l'ensemble des informations liées à une BD avec notamment une couverture en grand format, la liste des thèmes et des commentaires associés.
- Ajouter un formulaire permettant de saisir un nouveau commentaire pour cette BD.

#### **Contraintes :**

Limiter l'affichage de la liste à 5 ouvrages, avec des liens permettant d'afficher la suite.

(**Option** : ajouter un moteur de recherche qui affiche une liste de BD en fonction du thème et/ou de l'auteur et/ou de mot(s) saisi(s).)

### **Partie administration**

- Créer une interface permettant de modérer les commentaires laissés par les internautes.  
Un nouveau commentaire (modération = "") ne doit pas s'afficher sans avoir été modéré.  
La modération permettra de les rendre visibles (moderation='V'), de les modifier (corriger le texte) ou de les supprimer physiquement. Il sera nécessaire de modifier la base pour gérer un champ **modération**, les valeurs possibles seront contrôlées par la base de données.
- Créer une interface permettant de saisir un nouvel auteur.
- Créer une interface permettant de saisir une nouvelle BD.  
(**Option** : cette interface permet l'Upload de fichier et la génération de miniatures.)
- Créer une interface permettant de lier une BD à un thème.
- Créer une interface permettant de supprimer une image et sa miniature du serveur et d'effacer la ligne correspondante de la base de données.

Login administrateur : admin

Mot de passe : afpa

## **Livrables**

Un répertoire nommé : **/DL999\_Fil\_Rouge\_Web/** contenant :

1. Un fichier script de création de la base nommé **DL999\_BDTheque.sql**, uniquement si vous avez apporté des modifications sur la base (le nom de la base est imposé : cataloguebd).
2. Un sous-répertoire nommé **/DL999\_BDTheque/** contenant tous les fichiers de code, avec IMPERATIVEMENT un fichier **index.php** ou **index.html** dans ce répertoire. C'est le SEUL point d'entrée de l'application.

TOUS les autres fichiers seront rangés dans les répertoires suivants, vous n'êtes pas obligés de respecter le modèle MVC mais dans ce cas vous devez utiliser au minimum **/css/**, **/ini/**, **/js/** et **/images/**. Créer uniquement les répertoires dont vous vous servez.

*RQ : Nom de la base, user de la base, mot de passe de la base et mot de passe administrateur du site doivent être enregistrés dans UN SEUL fichier d'initialisation dans le répertoire **/ini/**.*

3. Un dossier applicatif nommé **DL999\_BDTheque.pdf**.

## **Dossier applicatif**

Le document devra comporter les éléments suivants :

- Une page de garde (titre du projet, auteur, date, logo Afpa) et un sommaire,
- Une introduction pour chaque partie,
- Partie conception de l'application avec :
  - Le plan de navigation
  - La charte graphique (entête(s), bandeau(x), menu(s), pied de page, polices, couleurs ...)
- Partie développement avec :
  - Liste des dossiers et fichiers créés avec descriptif.
  - Liste des classes éventuelles et des fonctions avec description 'synthétique'.
- Partie installation en précisant l'ordre d'exécution, le nom de la base, du login et du mot de passe éventuel, une liaison ODBC éventuelle...
- Partie tests avec la description des contrôles, du comportement de l'application et des jeux d'essai.

## **Fonctionnement**

L'application doit être intuitive (pas besoin d'une formation pour l'utiliser), ergonomique (de type site WEB, pas d'application en mode caractère) et fonctionnelle (contrôles des champs de saisie, gestion des erreurs d'accès à la BDD, les erreurs sont gérées et retournées à l'utilisateur par des messages explicatifs et compréhensibles), ...

... Bref, tout ce que **VOUS** attendez d'une application WEB.

# STRUCTURE DE LA BASE

## TABLES ET CHAMPS DE LA BASE DE DONNEES :

### auteurs

- **aut\_id** int11) NOT NULL AUTO\_INCREMENT,
- ***aut\_nom*** varchar255) COLLATE latin1\_general\_ci NOT NULL, PRIMARY KEY aut\_id)

### bandesdessinees

- **bd\_id** int11) NOT NULL AUTO\_INCREMENT,
- ***bd\_titre*** varchar255) COLLATE latin1\_general\_ci NOT NULL,
- ***bd\_resume*** text COLLATE latin1\_general\_ci NOT NULL,
- ***bd\_image*** varchar255) COLLATE latin1\_general\_ci NOT NULL,
- ***bd\_auteur\_id*** int11) NOT NULL, PRIMARY KEY bd\_id), KEY fk\_aut\_bd bd\_auteur\_id)

### commentaires

les avis de l'internaute *com\_auteur* sur une bande dessinée d'id *com\_bd\_id*

- **com\_id** int11) NOT NULL AUTO\_INCREMENT,
- **com\_bd\_id** int11) NOT NULL,
- ***com\_date*** timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,
- ***com\_auteur*** varchar255) COLLATE latin1\_general\_ci NOT NULL,
- ***com\_texte*** text COLLATE latin1\_general\_ci NOT NULL, PRIMARY KEY com\_id), KEY fk\_bd\_comment com\_bd\_id)

### liens\_bd\_themes

- **lien\_bd\_id** int11) NOT NULL,
- **lien\_themes\_id** int11) NOT NULL, PRIMARY KEY lien\_bd\_id,lien\_themes\_id), KEY fk\_themes\_lien lien\_themes\_id)

### themes

- **th\_id** int11) NOT NULL AUTO\_INCREMENT,
- ***th\_intitule*** varchar255) COLLATE latin1\_general\_ci NOT NULL, PRIMARY KEY th\_id)

## CONTRAINTES SUR LES TABLES :

### bandesdessinees

- `fk_aut_bd FOREIGN KEY (bd_auteur_id) REFERENCES auteurs (aut_id);`

### commentaires

- `fk_bd_comment FOREIGN KEY (com_bd_id) REFERENCES bandesdessinees (bd_id);`

### liens\_bd\_themes

- `fk_bd_lien FOREIGN KEY (lien_bd_id) REFERENCES bandesdessinees (bd_id) ON DELETE CASCADE,`
- `fk_themes_lien FOREIGN KEY (lien_themes_id) REFERENCES themes (th_id);`



## QUELQUES CONSEILS

Utiliser les données transmises par GET pour créer des requêtes SELECT.

Utiliser POST pour les requêtes INSERT, DELETE et UPDATE.

Tester vos requêtes dans PHPMyAdmin avant de les mettre dans le code PHP.

## HTML

- `<input type= « hidden » name= « ? » value= « ? » />`  
ce champ n'est pas visible à l'affichage mais sa valeur sera transmise lors de la validation du formulaire.
- `<input type= « ? » value= “?”» name= “myname[]” />`  
Un name se terminant par [] sera transmis sous forme de tableau:  
on peut donc le réutiliser dans le même formulaire coté serveur `$_POST[myName]` sera un tableau associatif.
- `<input type= « file » name= « ? » value= « ? » />`  
est un champ permettant l'upload de fichier.  
Le formulaire contenant ce champ doit comprendre l'attribut **enctype="multipart/form-data"**.  
Pour récupérer ce fichier coté serveur utiliser `$_FILES`.

## MYSQL

- **CURDATE() ou CURRENT\_DATE() :**  
Fonction renvoyant la date en cours.
- **LAST\_INSERT\_ID() :**  
Fonction renvoyant le dernier champ auto-incrémenté créé (durant une session de connexion).
- Il n'existe pas de type booléen en MySQL, utiliser un champ TINYINT avec des valeurs 0 et 1.
- InnoDB est un moteur de Bases de données permettant la gestion des clés étrangères en MySQL.

## PHP

- **Var\_dump(\$var)** et **print\_r(\$var) :**  
affiche les infos contenues dans \$var , utiles notamment lorsque \$var est un tableau ou un objet.
- **move\_uploaded\_file(\$repertoire/fichier temporaire, \$repertoire/fichier cible)**  
copie un fichier uploadé vers un emplacement du serveur.
- Créer une miniature en jpeg avec php :

*//calcul des dimensions de la miniature à partir de celles de l'image d'origine*

```
list($width, $height) = getimagesize($copieImage);
```

```
$new_height = 150;
```

```
$new_width = ($width/$height)*$new_height;
```

*//creation d'une image miniature en mémoire vers \$image\_mini\_tmp*

```
$image_mini_tmp = imagecreatetruecolor($new_width, $new_height)
```

*//copie de l'image à dupliquer en mémoire vers \$image\_tmp*

```
$image_tmp= imagecreatefromjpeg($copieImage)      )
```

*// copie du contenu de \$image\_tmp, vers \$image\_mini\_tmp en appliquant les nouvelles dimensions*

```
imagecopyresampled($image_mini_tmp, $image_tmp, 0, 0, 0, 0, $new_width, $new_height, $width, $height)
```

*//enregistrement de la miniature de crée vers \$copieMini ;*

```
imagejpeg ( $image_mini_tmp , $copieMini)
```