

Département : Informatique

Filière : *SMI* Semestre : *S6*

TP 7 Web Dynamique

Développer une application PHP en respectant le modèle MVC - Partie 1

L'objectif de cette série est de développer une application PHP tout en séparant les 3 aspects de l'application, à savoir l'accès aux données (*Modèle*), l'affichage (*Vue*) et les traitements (*Contrôleur*).

Nous allons reprendre notre exemple habituel de l'application de gestion des étudiants. L'objectif final est de réorganiser notre code de sorte à respecter le modèle MVC.

Vous pouvez vous inspirer de la séance du cours pour arriver jusqu'au résultat fourni dans le dossier « step1/version5 » (voir attachement) ».

I- Fonctionnalités de l'application

Notre application doit offrir les fonctionnalités suivantes :

- 1- L'application affiche une page d'accueil (qui explique l'objet de l'application).
- 2- Elle affiche la liste des étudiants (stockés dans la base de données).
- 3- En cliquant sur un étudiant, on affiche plus de détails sur cet étudiant.
- 4- On peut ajouter des nouveaux étudiants
- 5- On peut modifier un étudiant.
- 6- On peut supprimer un étudiant.

Ces fonctionnalités sont accessibles via des URLs de type : index.php?action=....

Pour cela les liens de notre application sont tous de la forme suivante :

```
<a href ="index.php?action=index">Acceuil</a> <br />
<a href ="index.php?action=liste">Liste de étudiants</a> <br />
<a href ="index.php?action=ajouter">Ajouter un étudiant</a> <br />
<a href ="index.php?action=modifier&CodeE=E1">Modifier un étudiant</a> <br />
<a href ="index.php?action=supprimer&CodeE=E1">Supprimer un étudiant</a> <br />
<a href ="index.php?action=detail&CodeE=E1">Détail d'un étudiant</a> <br />
<a href ="index.php?action=detail&CodeE=E1">Détail d'un étudiant</a> <br />
```



Département : Informatique

Filière : *SMI* Semestre : *S6*

II- Fichiers et fonctions à créer

Pour commencer, créez l'arborescence de l'application. Nous aurons besoin pour cela de:

- a. Un fichier *index.php*, qui va servir du point d'entrée unique à l'application. Tous les liens pointent vers ce fichiers en lui passant une variable *\$_GET["action"]*. Selon la valeur de cette variable, le script décide quelle est l'action à exécuter.
- b. Un script *EtudiantManager()* (qui peut être mis en un dossier *Models*). Ce script contient toute les fonctions d'accès à la base de données nécessaires pour l'application. Par exemple, nous aurons besoin des fonctions suivantes :
 - i. getCn(), une fonction qui permet de se connecter à la base de données.
 - ii. getListeEtudiants(), qui permet de récupérer tous les étudiants
 - iii. *getDetailEtudiant(\$code)*, qui permet de récupérer l'étudiant dont le code est passé en paramètre.
 - iv. *supprimerEtudiant (\$code)*, qui permet de supprimer l'étudiant dont le code est passé en paramètre.
 - v. ajouterEtudiant (array \$t), qui permet d'insérer un étudiant.
 - vi. modifierEtudiant (array \$t), qui permet de mettre à jour un étudiant.
- c. Un ensemble de vues (qu'on peut mettre dans un dossier *Views*). Ce sont les scripts qui contiennent le code HTML pour afficher les différentes pages. Nous aurons dans notre cas :
 - i. *vIndex.php* qui contient le code HTML pour afficher la page d'accueil.
 - ii. vListe.php qui contient le code HTML pour afficher la liste des étudiants.
 - iii. vDetail.php qui contient le code HTML pour afficher le détail d'un étudiant.
 - iv. *vFormAjouter.php*: qui affiche le formulaire pour ajouter un étudiant.
 - v. *vFormModifier.php*: qui affiche le formulaire pour modifier un étudiant.
 - vi. Pour la suppression, on affiche la liste des étudiants après.
- d. Un fichier *EtudiantController.php* (qui peut être mis en un dossier *Controllers*). Ce script implémente les différentes actions sous forme de fonctions (par exemple, on peut les nommer comme suit :
 - i. indexAction()
 - ii. listeAction()
 - iii. detailAction()
 - iv. supprimerAction()
 - v. ajouterAction()
 - vi. modifierAction()



Département : Informatique

Filière : *SMI* Semestre : *S6*

III- Mise en place

Supposons que l'utilisateur a cliqué sur le lien « Liste des étudiants ». Alors, le script *index.php* est exécuté en lui passant la variable « *\$action* » par la méthode *GET*. Dans ce cas, ce script doit appeler la fonction *listeAction()* du contrôleur. La fonction *listeAction()* doit appeler la fonction *getListeEtudiants ()* (qui est définie dans le modèle) pour récupérer les étudiants de la BD. Le résultat doit être stocké dans une variable « *\$etudiants* ». Et en fin, l'action termine par afficher la vue *vListe.php*.

Donc, pour implémenter l'action « liste » vous devez :

- 1- Modifier le fichier *index.php* pour prendre l'action en considération. C'est-à-dire, appeler l'action *listeAction* quand la condition *\$_GET["action"]=="liste"* est vraie.
- 2- Modifier le script *EtudiantManager.php* pour y ajouter la fonction *getListeEtudiants()* qui cherche les étudiants dans la BD.
- 3- Ecrire la vue *vListe.php* qui boucle sur un tableau *\$etudiants* pour afficher son contenu.
- 4- En fin, modifier le contrôleur *EtudiantController.php* pour lui ajouter l'action *listeAction()*. Cette fonction fait le lien entre le modèle et la vue. Elle appelle la fonction du modèle *getListeEtudiants()* et stocke son résultat dans la bonne variable puis affiche la bonne vue.
- 5- Voilà. Vous pouvez tester maintenant la fonctionnalité « liste des étudiants »
- 6- Répéter les points de 1 à 5 avec les autres actions.