TD 3 – PHP

MVC :Modèle-Vue-Contrôleur

POO : Programmation Orientée Objet

Exercice 0 :

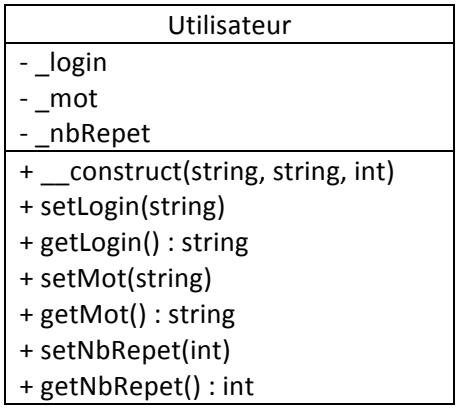
Récupérer le répertoire helloworld5 (helloworld5.zip). Visualiser les répertoires et les fichiers un par un, comprendre leur contenu et les compléter pour rendre le site fonctionnel :

a. Le répertoire config contient configuration.php, à compléter.
b. Les répertoires assets/css et assets/scripts contiennent des fichiers nécessaires pour utiliser le framework CSS Bootstrap. La compréhension de ces scripts n'est pas l'objet du module. Ils permettront juste de s'affranchir rapidement de la partie présentation du site pour se concentrer sur le PHP. Vous pourrez ajouter votre propre fichier CSS si nécessaire.
c. Le répertoire languages contient un fichier pour chaque langue proposée. Compléter le fichier languages/FR-fr.php.
d. Le fichier index.php à la racine du site charge le fichier de configuration et joue le rôle de routeur en choisissant le contrôleur selon la page demandée.
e. Le fichier controllers/404.php se contente d'appeler la vue views/404.php pour spécifier que la page demandée n'existe pas. Tester avec l'URL : http://…/helloworld5/index.php?page=toto
f. Toutes les vues doivent commencer par inclure header.php et terminer par inclure footer.php. Elles incluent alerte.php si un message d'erreur doit être affiché.
g. Le répertoire lib contiendra une bibliothèque de fonctions. Une fonction choixAlert permet de choisir le type d'alerte à afficher.
h. Quand aucune page n'est précisée dans l'URL on tombe sur le contrôleur index.php qui charge la vue index.php. Faîtes en sorte que cette vue affiche le formulaire qui sera traitée par helloworld5.php?page=hello.
i. Créer le contrôleur hello.php, qui :
    - fera appel au modèle hello.php pour accéder à la base de données et extraire de celle-ci les données souhaitées
    - fera appel à la vue hello.php à créer pour afficher les messages de bienvenue si l'utilisateur a bien été trouvé dans la base de données
    - affichera de nouveau le formulaire mais avec une alerte si l'utilisateur n'existe pas dans la base de données. Ce sera également le cas en cas de problème de connexion à la base de données ou si la requête se passe mal.

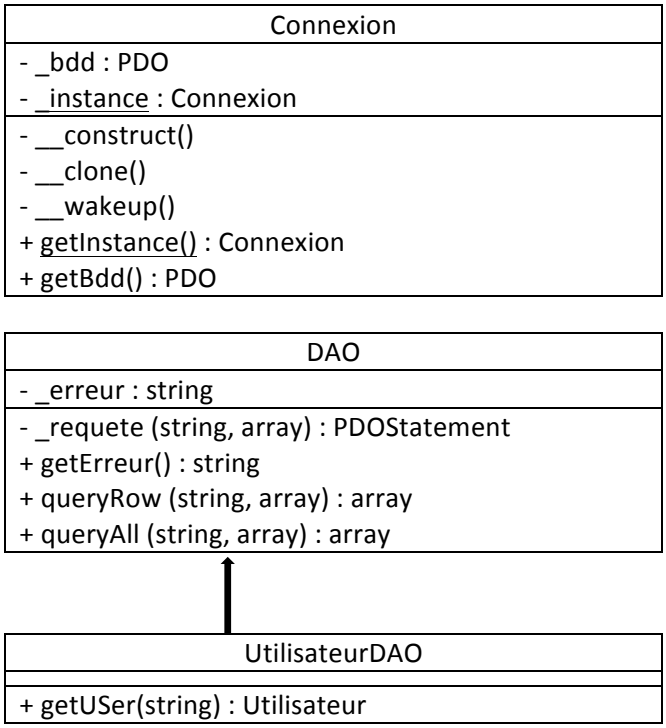
Objectif : Utilisation de la POO pour la partie Modèle et le transfert des données du modèle au contrôleur et du contrôleur à la vue.

Exercice 1 :

1. Créer un répertoire entities à la racine de votre site et une constante PATH_ENTITY. Dans ce répertoire entities, créer la classe Utilisateur.



2. Copier les classes Connexion et DAO dans le répertoire Model de votre site. Implémenter la classe UtilisateurDAO qui hérite de la classe DAO.



3. Modifier le contrôleur hello.php et la vue hello.php pour prendre en compte le nouveau modèle. On supprimera l'ancien modèle hello.php.

Exercice 2 :

Introduction :

Ce travail sera effectué en binôme. Il faudra que le résultat soit visible sur le site de l'IUT. Créer un nouveau projet privé dans gitlab avec comme membres le binôme et l'enseignant. Ajouter un readme à ce projet rappelant le nom du binôme et précisant l'URL du site sur le serveur de l'IUT. A chaque ajout d'une nouvelle fonctionnalité (chaque fin de question), effectuer un commit pour montrer votre progression. D'autres commits (correction d'un bug, amélioration de code …) peuvent bien sûr être effectués.

Problématique :

On souhaite utiliser une base de données de photos classées par catégorie et permettre aux internautes d'accéder au contenu de la base. La base de données est modélisée de la façon suivante :

Categorie(catId, nomCat)

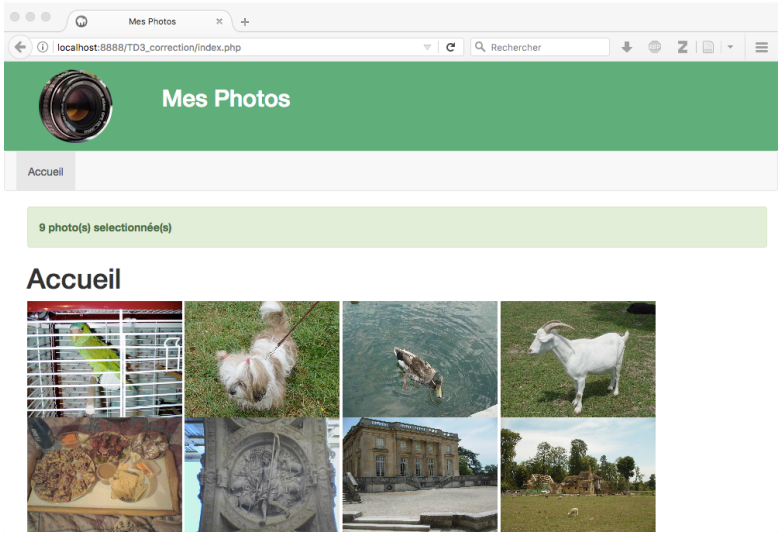
Photo (photoId, nomFich, description, #catId)

Les clés primaires sont soulignées et les clés étrangères sont précédées d'un dièse.

1. Pour commencer, télécharger le zip mvc_maison. Décompresser et renommer en TD3. Récupérer et sauvegarder à l'endroit adéquat les premières images à intégrer au site (images.zip).

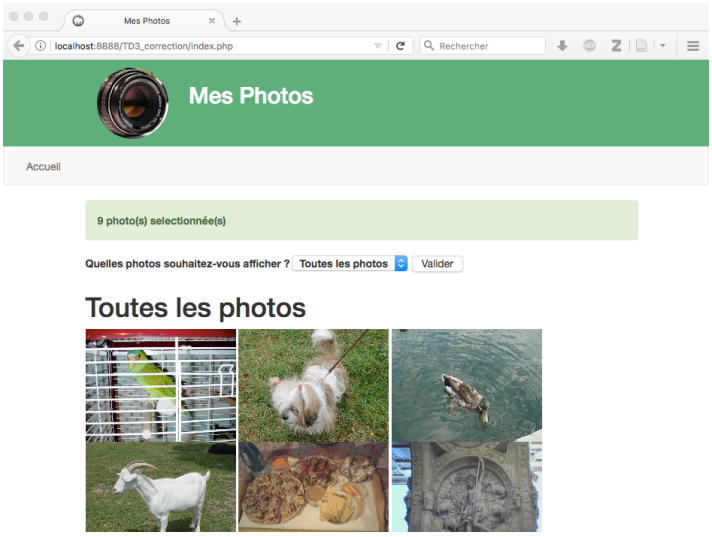
2. Créer les tables dans la base de données. Les identifiants sont des entiers. Les noms et la description sont des varchar de taille 250 max avec utf8_general_ci comme valeur d'interclassement. Importer les données dans les tables (tuplesPhoto.sql). Exporter les tables et leur contenu dans un fichier bdd.sql dans un répertoire bdd à la racine de votre site pour sauvegarde.

3. Dans un premier temps, on souhaite permettre à l'internaute de visualiser sur une page l'ensemble des photos référencées dans la base de données. Pour ça, on construit la page d'accueil qui donnera le résultat :

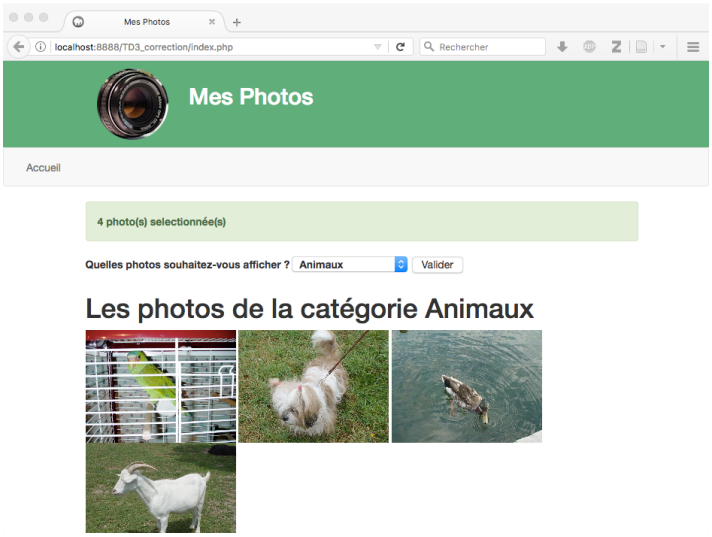


4. Dans un second temps, on veut pouvoir visualiser les photos pour une catégorie donnée.

Au lancement de la page :



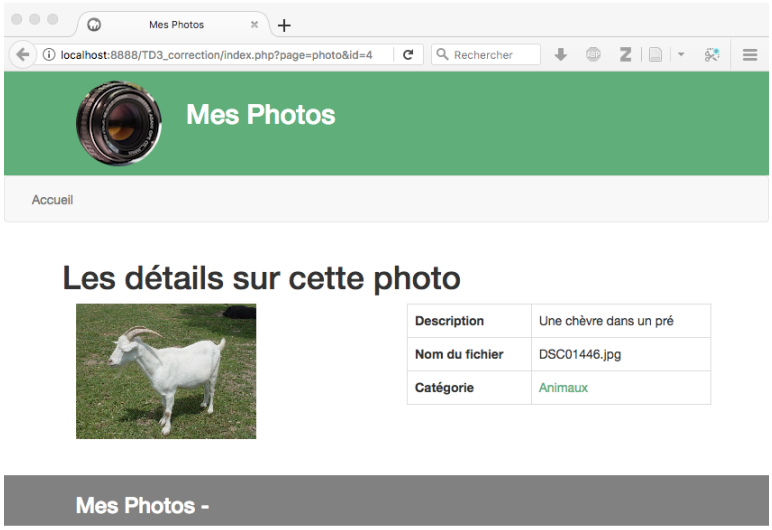
Après sélection d'une catégorie et validation :

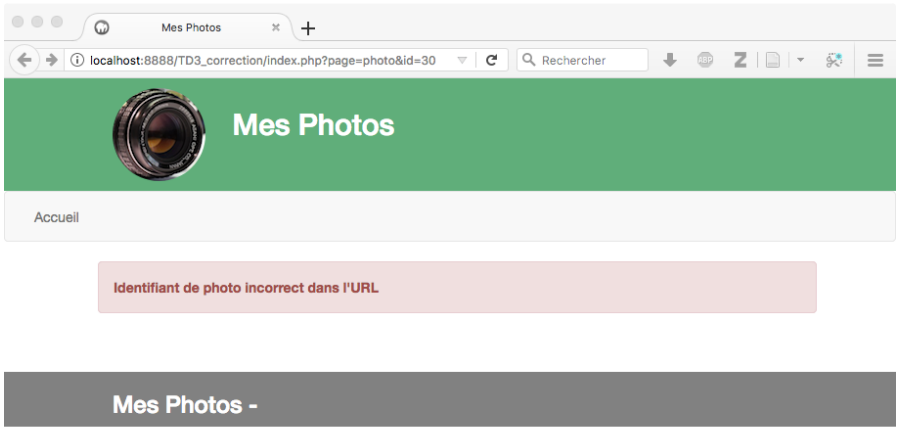


5. Pour finir, on souhaite pouvoir afficher le détail d'une photo en cliquant sur une image :

    a) Ce sera une page dédiée à cette image. On verra l'image bien sûr, mais aussi le nom du fichier correspondant sa description et sa catégorie.

    b) Des liens permettront de revenir sur la page qui affiche toutes les photos (accueil dans le menu) ou sur celle qui affiche toutes les photos de la même catégorie (en cliquant sur la catégorie).





Remarques sur l'utilisation de bootstrap :

1. La page d'une photo est obtenue en mettant l'image d'une part et la table d'autre part dans deux balises div : `<div class = "col-md-6 col-sm-6 col-xs-12">`

2. La présentation de la table est obtenue avec deux classes ainsi :
   `<table class="table table-bordered">`