

Module : Programmation Web Avancée

Filière GI, Semestre : 4, AU : 2022-2023

Mini Projet "Gestion des demandes de modules libres"

Quelques instructions :

- Le mini projet doit être réalisé par un binôme d'étudiants.
- La production finale consiste en deux espaces (Admin et Etudiant)
- La production finale doit être prête à la date fixée et envoyée à l'adresse r.aitdaoud@usms.ac.ma
- Le mini-projet doit porter le nom Nom1Nom2_miniprojetWeb avec Nom1 et Nom2 sont les noms de famille des deux étudiants.
- Un simple tutoriel "PDO en PHP" est mis à votre disposition pour vous aider à s'initier avec l'accès aux bases de données avec PDO.

Objectifs pédagogiques :

L'objectif pédagogique de ce mini-projet consiste à valider les compétences acquises en cours et en TP du module Programmation Web Avancée, à savoir.

- Création des formulaires avec contrôle de saisie
- La mise en place des pages html
- Gestion et Upload des fichiers
- Notion des sessions
- L'authentification statique et dynamique
- Manipulation d'une base de données

L'objectif principal de ce mini projet et de savoir utiliser l'objet **PDO PHP** pour interagir avec une base de données en PHP.

Notez que, ce mini-projet et la suite de ce que nous avons vu dans les 3 Tps. Afin d'interagir avec une base de données MySQL, PHP propose trois méthodes différentes :

- + **Extension MySQL originale** : Il s'agit de l'extension MySQL native au PHP (elle est rendue obsolète depuis la version 5.5 de PHP).
- + **Extension MySQLi**: Il s'agit d'une extension un peu plus moderne.
- + **Objet PDO (PHP Data Objects)**: L'objet PDO constitue de loin la meilleure solution pour interagir avec **différentes** base de données (Mysql, Oracle, PostgreSQL, SQLServer, ...) en PHP.

Travail à faire :

1. Démarrer votre serveur web, puis accéder au phpMyAdmin.
2. Ensuite, donnez le nom d'utilisateur "root", le mot de passe correspondant.
3. Créer une base de données nommée **GestionDemandeModules**, avec les tables suivantes :
 - + Utilisateur (**iduser**, login, password, profil, statut)
 - + Etudiant (**idetud**, apogee, nom, prenom, datenaissance, statut, filiere)
 - + EventUser(**idevent**, *iduser*, dateevent, ipadress)
 - + Demande(**iddemande**, *idetud*, datedemande, modulesdemandees, file_releve, file_carte, iduser, reponseadmin, datereponse)
4. Mettre à jour la page index.php pour que l'authentification des étudiants soit réalisée à partir de la base de données (utiliser le couple Num Apogee et date de naissance).
5. Créer une page login.php qui permet aux personnels du service de scolarité de s'authentifier avec un login et un mot de passe.
6. Selon le profil de l'utilisateur (Chef scolarité ou Agent), la page login.php doit rediriger l'utilisateur à une page gestion.php dont le contenu est dynamique :

Liste des fonctionnalités fournies par la page gestion.php :

Pour le Chef de scolarité :

- Consultation, Ajout, modification, suppression du contenu de la table Etudiant.
- Consultation des demandes des étudiants (avec possibilité de voir les fichiers relevés de notes et carte étudiants de chaque étudiant).
- Faire des recherches sur les étudiants (par num apogee, et par filière)
- Répondre aux demandes des étudiants.
- Avoir une vision globale sur les demandes (des chiffres sous forme de tableau)
 - + Nombre total des étudiants
 - + Nombre total des demandes
 - + Nombre des demandes par filière
 - + Nombre de demandes sans réponse
- Clôturer l'opération des demandes de modules libres
- Ajout, modification, suppression d'un utilisateur

Pour l'Agent :

- Consultation des demandes des étudiants.
- Faire des recherches sur les étudiants (par num apogee, et par filière)
- Répondre aux demandes des étudiants.

7. Faire les traitements nécessaires afin d'alimenter les deux tables *EventUser* et *Demande* (N.B : Un étudiant ne peut envoyer qu'une seule demande, par contre, il peut modifier sa demande à tout moment).
8. N'oublier pas de s'assurer que les pages ne sont pas accessibles que si l'étudiant ou l'utilisateur passent via l'authentification.
9. Travailler sur la partie front end