



Examen final : Langage de script – PHP

Durée : 2h00

Les documents (livres, notes de cours, ...) ne sont pas autorisées

Exercice1 (5 point)

Nous nous intéressons ici à la définition et à la manipulation d'un tableau de notes (entiers). Écrire les fonctions PHP qui :

- affiche tous les éléments d'un tableau,
- calcule la moyenne des nombres contenus dans un tableau donné,
- teste si la note 20 est présente ou non,
- détermine la meilleure note obtenue,
- indique combien d'éléments sont supérieurs ou égaux à 10.

Exercice2 (15 points)

On voudrait réaliser une application web de gestion des employés avec HTML – PHP et MySQL. Pour ce faire, on va réaliser des pages web contenant, selon le besoin, des formulaires ou des tables HTML. Le traitement des données se fait avec PHP orienté objet et leur stockage se fait dans une base de données MySQL. L'application gère les employés et les départements auxquels ils sont affectés (Exemple : l'employé X est affecté au département Informatique).

La base de données « bd_employe » avec les tables suivantes est déjà créée :

- employes (mat_employe, nom, prenom, date_embauche, id_departement) ;
- departement (id_departement, nom_departement) ;

Travail à faire :

1. Réaliser une classe « ConnexionBDD » qui permet :
 - a. D'établir une connexion avec la base de données « bd_employe » : **Connexion()** ;
 - b. L'ajout d'un employé : **addEmploye(matricule, nom, prenom, date_embauche, id_departement)** ;
 - c. La recherche de tous les employés d'un département spécifier : **searchEmployes(id_departement)** ;
 - d. Afficher tous les départements : **listDepartement()** ;

Remarque : Les paramètres de connexion à la BDD sont: le serveur : localhost ; l'utilisateur : admin ; le mot de passe : admin ; le nom de la base de données : bd_employes

2. Réaliser un formulaire «**addEmploye.php**» : d'ajout d'un nouvel employé avec les éléments suivants:

- Un champ de texte pour la saisie du matricule de l'employé ;
- Une zone de texte pour la saisie du nom de l'employé ;
- Une zone de texte pour la saisie du prénom de l'employé
- Une liste déroulante pour le choix du département auquel on va affecter l'employé (la table département est considérée déjà remplie) ;
- Un bouton d'envoi de données ;

Remarque : La liste déroulante qui doit contenir les départements doit avoir comme valeur « **id_departement** » et comme étiquette « **nom_departement** »

3. Réaliser une page PHP « **addEmployeT.php** » pour le traitement des données du formulaire comme suit :

- a. Récupérer les données provenant du formulaire ;
- b. Faire appel à la classe connexion pour établir une connexion et appeler la méthode d'ajout d'un employé **addEmploye(matricule, nom, prenom, date_embauche, id_departement)** ; rediriger vers une page HTML « **listEmployes.php** ». (On ne vous demande pas d'implémenter la page **listEmployes.php**)

4. Réaliser une page : **searchEmploye.php** qui contient un formulaire avec les éléments suivants:

- a. Une liste déroulante pour choisir un département
- b. Un bouton d'envoi des données

Remarque : La liste déroulante qui doit contenir les départements doit avoir comme valeur « **id_departement** » et comme étiquette « **nom_departement** »

5. Réaliser une page **searchEmployeT.php** avec les fonctions suivantes :

- a. Récupérer le département sélectionné ;
- b. Appeler la méthode **searchEmploye(id_departement)** ;

Documentation PHP

Accès à la base de données MySQL

- `$cnx=new PDO('mysql:host=serveur;dbname=nom de la base ', 'login', 'motdepasse')` pour ouvrir une connexion avec la base de données MySQL.
- `$cnx->query(requete)` pour exécuter une requête select.
- Récupération des résultats: A travers la méthode `fetch(..)` , les valeurs possibles pour le paramètre de fetch : `PDO::FETCH_ASSOC`, `PDO::FETCH_NUM`, `PDO::FETCH_OBJ`
- `$cnx->exec(requete)` pour exécuter les requêtes de mis à jour (insert, update, delete).