

Cette partie du TP comprend la création et l'utilisation de la base de donnée pour créer une page destinée à l'affichage des avis.

1. **Création de la base de donnée**

Utilisez au choix Mysql ou PostgreSQL. Créez une base de donnée de « ratingdb »

Dans un premier temps cette base ne contiendra qu'une seule table :

avis (id, departement, promo, module, note, commentaire, etudiant, date)

Le champ « id » est un numéro de série auto-incrémenté. c'est aussi la clé primaire.

les champs « departement, promo, module, etudiant » sont des chaines de caractères.

Le champ « commentaire » est une chaine de caractères d'au moins 300 caractères.

La note est un entier de 1 à 5.

Par défaut « date » contient la date et l'heure de création.

Il est souhaitable d'ajouter des contraintes à votre base de donnée.

Insérez au moins une dizaine d'enregistrements dans la base.

2. **Paramètres d'accès à la base de donnée**

Créez, un script « config-db.php » dans « Models » qui renseignera les variables :

\$DBHOST, \$DBDRIVER, \$DBNAME, \$DBUSER, \$DBPASSWORD

Ce fichier sert simplement à isoler les variables définissant l'accès à la base de donnée.

3. **Connexion à la base de donnée**

Activez si nécessaire dans le « php.ini », l'extension de pilote PDO correspondant à votre base de donnée.

Créez un script « connect-db.php » dans « Models » pour instancier un objet PDO nommé \$PDO qui sera utilisé chaque fois qu'un accès à la base de donnée est nécessaire.

Utilisez pour le constructeur les informations de votre fichier « config-db.php »

En cas d'échec de la connexion générez une erreur (\$ERROR) et créez le message correspondant dans le tableau \$ERRO_MESS. Terminez le code par l'affichage de la vue error.

4. **Ajout d'un modèle « lister »**

Dans votre dossier « Models » ajoutez un script « avis-lister.php ».

Créez y une fonction « get_avis ».

Cette fonction fera appel à la variable globale \$PDO pour interroger la base de donnée. Gérez les inclusions de fichiers en conséquence.

Avec une requête préparée (même si ce n'est pas encore indispensable ici), récupérez le contenu de la table « avis » que vous retournerez sous la forme d'un tableau associatif.

5. **Ajout du contrôleur correspondant**

Dans le dossier « Controllers », créez un script « avis-lister.php ».

Dans ce fichier appelez la fonction « `get-avis()` » pour récupérer les avis avant d'inclure la vue correspondante.

6. Ajout de la vue

Dans le dossier « Views », créez la page « `avis-lister.php` ». Cette page doit construire un tableau html avec les données transmises par le contrôleur. Une ligne avec les titres de colonnes « id, departement... » sera ajoutée.

Renseignez les variables « `$title` » et « `$content` » et utilisez le gabarit. Les fonctions « `cell()`, `row()` et `table()` » de votre bibliothèque de tags vous permettront d'améliorer la lisibilité.

A ce stade le contrôleur doit fonctionner et l'appel de la page « <http://localhost/index.php?controller=avis-lister> » doit produire un affichage correct.

7. Améliorations

• Ajout d'un critère de tri

Ajouter un argument « `$tri` » à votre fonction `get_avis()` avec « id » comme valeur par défaut. Modifiez votre SQL pour en tenir compte.

Dans le contrôleur, vérifiez la présence du paramètre `$_GET['tri']` et si nécessaire ajoutez le.

Transmettez ce paramètre lors de l'appel de la fonction.

Vérifiez avec l'appel à « <http://localhost/index.php?controller=avis-lister&tri=note> » le bon fonctionnement de votre critère de tri.

Modifiez votre vue et transformez les titres de colonnes en liens hypertexte pour recharger la page avec des critères de tri différents. Vérifiez.

Si elle n'existe pas encore, créez une fonction « `check_array()` » dans votre bibliothèque de vérification. Cette fonction permet de vérifier si une valeur est présente dans un tableau. Elle retourne 0 (Ok) ou un numéro d'erreur en cas d'absence. Ajoutez le message correspondant dans le tableau `$ERROR_MESS`.

Modifiez le contrôleur pour vérifier la clef de tri avant utilisation et pour générer une vue d'erreur en cas de problème. Provoquez volontairement une erreur de paramètre pour vérifier.

• Limite des commentaires

Modifiez votre vue pour limiter à 50 caractères le commentaire affiché pour un avis. Un lien « plus » permettra d'appeler le contrôleur « `avis-detail` ». Ce contrôleur pourra être réalisé comme exercice complémentaire si vous disposez de temps.

• Ajout d'un nombre limite d'enregistrements à afficher.

Un nouvel argument « `max` » doit permettre de limiter le nombre d'avis affiché par page.

Vérifiez son existence (donnez une valeur par défaut), son contenu (un entier positif inférieur à 20).

Modifiez votre fonction « `get_avis()` » pour le prendre en compte.

Vérifiez avec « <http://localhost/index.php?controller=avis-lister&max=5> »

• Ajout d'un offset

Un nouvel argument « `offset` » va être utilisé pour construire un affichage page par page.

Vérifiez son existence (donnez une valeur par défaut), son contenu (un entier positif inférieur au

nombre d'enregistrement contenu dans la table). Pour cela, ajoutez une fonction count-avis() dans votre modèle.

Modifiez votre fonction « get_avis() » pour prendre en compte ce nouvel argument.

Vérifiez que « <http://localhost/index.php?controller=avis-lister&max=5&offset=10> » affiche 5 avis à partir du 10ème.

- **Barre de navigation**

Ajoutez une barre de navigation dans votre vue.

Elle peut contenir :

- des liens « précédent , suivant »,
- une numérotation des pages « 1, 2, 3, 4 »
- les liens « premier , dernier ».

Gérez le contenu de la variable « offset » en conséquence.

- **Ajout d'un formulaire de saisie du nombre d'avis/page**

Ajoutez à votre barre un formulaire pour modifier le nombre max d'enregistrements. Vérifiez la donnée transmise avec une fonction check_integer().

Agir sur la variable « max » qui doit être transmise de page en page (pas encore de session)

- **Ajout d'un filtre (requête préparée)**

Ajoutez un département par module sous la forme d'une liste de sélection qui permet d'afficher tous les avis ou de restreindre à d'un département particulier.

Vérifiez le paramètre transmis et modifiez la fonction get_avis() en conséquence. Ici la requête préparée prend toute son importance.

- **Amélioration de l'affichage avec une feuille de style.**

Ajoutez une feuille de style dans le dossier « css » et personnalisez l'affichage de la page.

Remplacez la note par l'affichage d'une série d'étoiles.

Supprimez les informations inutiles (id).