Projet : Système de blog

Les blogs sont nombreux parce qu'ils sont très faciles à écrire – il suffit d'un système capable de mémoriser des dates et du contenu. Le blog que nous présenterons ici permet d'ajouter des billets, des commentaires et de visualiser les billets. Le système stocke les billets et les commentaires dans une base de données MySQL et utilise Smarty pour afficher des templates.

La table qui contient les billets s'appelle billets_blog :

```
CREATE TABLE billets_blog (
ID INT NOT NULL AUTO_INCREMENT ,

titre VARCHAR( 120 ) NOT NULL ,

contenu TEXT NOT NULL ,
annonce TINYTEXT NOT NULL ,

date_billet DATETIME NOT NULL ,

categorie VARCHAR( 12 ) NOT NULL , PRIMARY KEY ( ID )

) TYPE = INNODB ;
```

La signification de plupart de ces champs est évidente. Annonce est un extrait du billet, ne contenant aucune balise. Les commentaires sont stockés dans la table commentaires_blog :

```
CREATE TABLE commentaires_blog (

ID_comment INT AUTO_INCREMENT ,

nom VARCHAR( 50 ) NOT NULL ,

comment TEXT NOT NULL ,

date_comment timestamp,

ID INT NOT NULL ,

PRIMARY KEY ( ID_comment ) )

 TYPE = INNODB ;
```

La raison pour laquelle le champ date_comment est de type timestamp est que nous n'avons pas l'intention de permettre la modification des commentaires : une fois posté, il ne changera jamais et il n'y a donc pas besoin de configurer manuellement sa date.

Comme pour les systèmes précédents, nous utiliserons un fichier de configuration *config_blog.php* pour mettre en place la connexion MySQL et créer un objet Smarty. Avec cette configuration par défaut, les templates Smarty se trouveront dans le répertoire *templates* :

Voici un résumé des quatre scripts du système :

editer_blog.php : Ajoute et modifie les billets du blog

index_blog.php : Affiche la liste des billets du blog

afficher_blog.php : Affiche un billet individuel en intégralité

commenter_blog.php : Ajoute un commentaire à un billet du blog

Créations de billets

Avant de faire quoi que ce soit d'autre, il faut pouvoir ajouter du contenu. Nous allons donc commencer par l'éditeur de billet. Avec ce script, nous pourrons réellement nous rendre compte que l'utilisation de Smarty permet de séparer les templates HTML de PHP et que cela produit un code bien plus propre. Commençons par le template *templates/edition_blog.tpl* :

```
<html>

<head>

        <title>{$titre}</title>

{literal}
<style>
h1 {

font-family: sans-serif;

font-size: 20px; }

table.champs_saisie { font-family: sans-serif; font-size: 12px;

}
table.champs_saisie td {

vertical-align: top; }

</style>

{/literal}

</head>
<body>
<h1>Nouveau billet</h1>
```

```
<form method="post" action="editer_blog.php">
<table class="champs_saisie">
<tr> <td>Titre :</td><td><input name="titre" type="text" /></td> </tr> <tr> <td>Contenu
:</td>

<td><textarea name="contenu" rows="15" cols="40"></textarea></td> </tr>

<tr> <td>Catégorie :</td><td><input name="categorie" type="text" /> </tr>
<tr> <td /><td><input name="submit" type="submit" value="Poster" /></td> </tr> </table>
</form>
</body>
</html>
```

Ce template n'est rien de plus qu'un formulaire avec des champs titre, contenu et categorie envoyés à *editer_blog.php via* la méthode POST. La Figure ci-dessous montre ce formulaire affiché dans un navigateur.

Le script *editer_blog.php* fonctionne en deux modes. S'il prend ses entrées à partir du formulaire précédent, il nettoie ces entrées, ajoute le nouveau billet dans la base de données puis redirige l'utilisateur vers une page d'affichage de ce billet.

Affichage d'un billet

Le script *afficher_blog.php* doit afficher trois éléments : un billet, les commentaires sur ce billet et un formulaire permettant de saisir un nouveau commentaire.

Ces trois composants sont décrits dans *templates/afficher_blog.tpl*. La première partie contient des informations d'en-tête et une fonction JavaScript qui nous servira plus tard à afficher le formulaire pour les commentaires :

Pour les commentaires sur les billets, nous utiliserons la fonctionnalité {section} de Smarty car elle nous permet d'afficher un nombre quelconque de commentaires en parcourant un tableau. Ici, la variable $commentaires est un tableau de commentaires, chacun étant lui-même un tableau ayant pour clé nom, date et comment pour accéder aux différents contenus des commentaires. Smarty parcourt les commentaires et affiche le contenu de cette section pour chacun d'eux, ce qui permet une représentation très compacte de ce traitement.

Enfin, nous devons afficher le formulaire pour permettre aux utilisateurs d'ajouter leurs commentaires. Pour cela, on utilise une petite astuce : au lieu d'afficher directement le formulaire, on le cache jusqu'à ce que l'utilisateur clique sur le lien JavaScript "Ajouter un commentaire".

La Figure ci-dessous montre un exemple de billet avec un formulaire de commentaire caché.

Maintenant que nous nous sommes occupés du HTML, *affiche_blog.php* est très simple à écrire. Nous suivons le scénario classique consistant à vérifier le paramètre d'entrée ID et à rechercher le billet dans la base de données.

Ajout de commentaires

Le script qui ajoute des commentaires aux billets, *commenter_blog.php,* est le seul du système qui n'utilise pas de template car il n'affiche rien. Dans la section précédente, nous avons vu qu'il prenait trois paramètres : ID, commentaire et nom contenant, respectivement, l'identifiant du billet, le contenu du commentaire et le nom de celui qui a posté le commentaire. La première étape, comme d'habitude, consiste à vérifier que l'identifiant est correct et correspond à un billet existant :

Création d'un index des billets

La page d'accueil qui présente les billets les plus récents ressemble à la page d'affichage d'un billet car elle utilise la fonctionnalité {section} de Smarty pour réduire le code de l'itération.

Le fichier *templates/index_blog.tpl* commence par cette information d'en-tête classique :

```
<html>

<head>

<title>{$titre}</title>

{literal} <style>

table.billets { font-size: 12px;

}
table.billets td {

padding-bottom: 7px; }

.entete { font-size: 14px;

font-weight: bold; }

</style> {/literal} </head> <body>

<a href="index_blog.php">Mon blog</a>
```

Le script ***index_blog.php*** commence par vérifier l'existence d'un paramètre catégorie ; s'il existe, l'index n'affichera que les billets de cette catégorie. Pour ce faire, on nettoie d'abord ce paramètre et on ajoute une clause WHERE pour restreindre la requête que nous verrons bientôt. S'il n'existe pas, on initialise la clause WHERE et la variable Smarty avec des chaînes vides.

On termine en affectant le titre de la page et en affichant le template (voir la Figure ci-dessous pour le résultat final) :