

Rapport sur le projet php

Contrôle du document

Historique des versions

Date de création : Lundi 26 décembre 2012.

Date	Version	Contributeurs	Principales modifications
08/01/13	1.0	Sinnaeve Alexis	Création du document

Distribution

Document destiné à Monsieur Lannoy

État

Rédaction du document terminée.

Sécurité et Confidentialité

Travail réalisé par les étudiants de l'Université du Littoral Côte d'Opale. Toute reproduction, qu'elle soit partielle ou non, est interdite sans autorisation.

Participants

Responsabilité

Les informations de ce document sont des informations prévisionnelles. Elles sont susceptibles d'évoluer en fonction des divers événements pouvant survenir.

Commentaires

Aucun commentaire.

1. Structure du blog.

Avant de commencer le blog, nous avons reçu la structure en HTML et CSS faite en utilisant bootstrap. Cette structure a été placée dans deux fichiers nommés « haut.inc.php » et « bas.inc.php ». Ces deux fichiers seront inclus dans les autres fichiers afin d'avoir la structure du site. Cela permet également d'éviter d'avoir à recopier dans chaque fichier tout le code. Les fichiers sont donc moins lourds et lisibles.

2. La base de données.

Pour mettre en place le blog, nous avons eu besoin de créer une base de données. Dans cette base de données, nous avons commencé par créer la table article qui contient les champs suivants :

- Un identifiant qui est unique
- Un champ contenant le titre de l'article
- Un champ contenant le contenu du texte de l'article
- Un champ stockant la date de l'article

La seconde table que nous avons créée est la table utilisateurs. Elle permet d'enregistrer des personnes pour qu'ils puissent se connecter sur le blog. Elle contient donc :

- un identifiant unique
- Un champ texte qui contient l'email
- Un champ texte qui contient le mdp
- Le sid qui sera généré automatiquement

La troisième table enfin est la table tags qui contient les tags saisis par l'utilisateur lors de la création d'un article. Voici ses champs :

- Un identifiant unique
- Un champ texte tag qui stocke les tags en entrée
- articles_id qui permet de faire une liaison avec la table article

3. Le fichier « index.php »

Ce fichier est la page d'accueil du blog. On y affiche tous les articles. Chaque article contient un titre, une date de rédaction et le contenu de l'article. Ces éléments sont récupérés dans la base de données via les requêtes suivantes :

```
SELECT * FROM articles $where ORDER BY date DESC LIMIT $debut,$app"
```

Chaque article est suivi de boutons « modifier » et « supprimer ». Pour rendre le blog encore plus ergonomique, nous avons décidé de ne pas afficher l'intégralité de tous les articles sur une page, donc j'ai décidé d'afficher 3 articles par page. Ce qui nous a amené à un point fort du projet la pagination. La pagination a été l'un des éléments de programmation le plus difficile à mes yeux car il y a énormément de facteurs à prendre en compte. Il a fallu faire de nombreux tests comme pour savoir sur quelle page on se trouve ou encore ne pas rendre cliquables les liens « page suivant » et « page précédente » (voir photo)



Enfin voici un aperçu de la page index.



4. Article.php

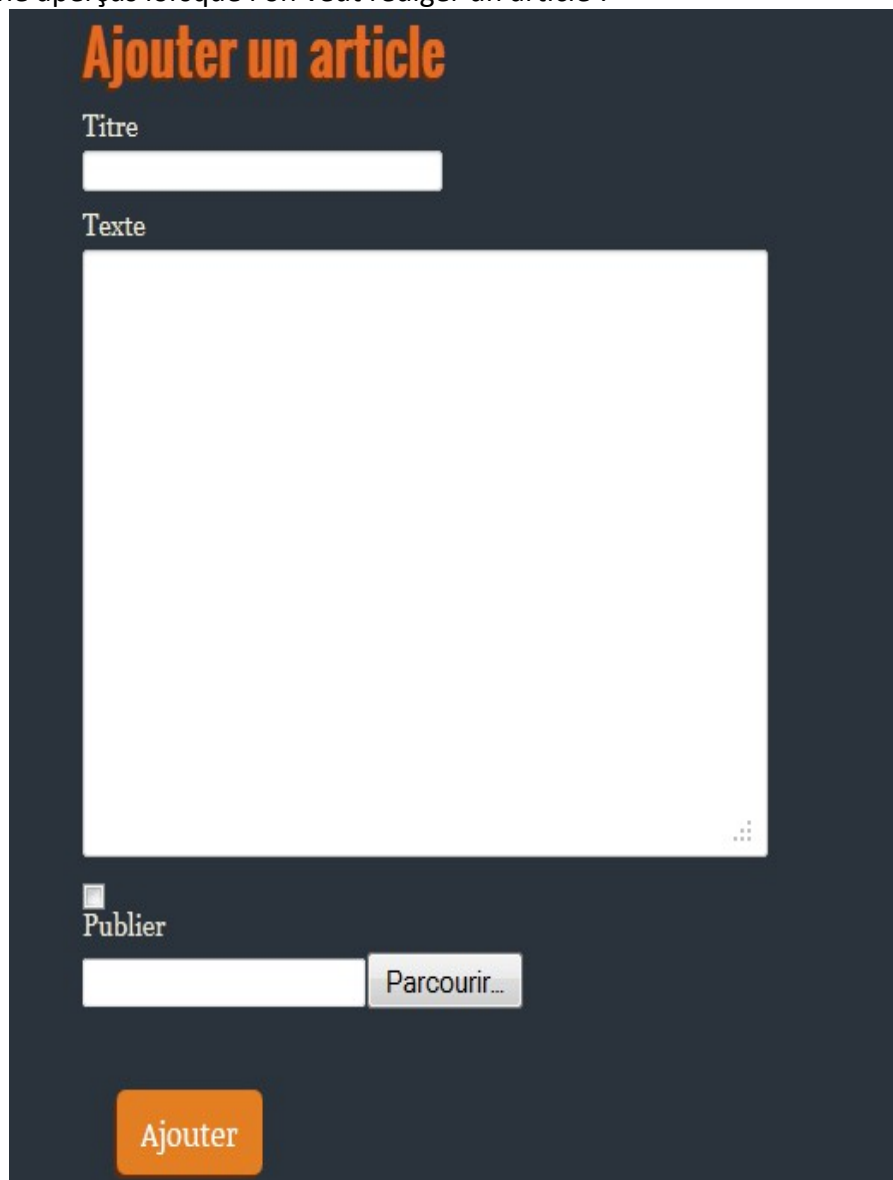
Ce fichier permet de rédiger ou de modifier un article. Il comprend donc un formulaire.

Pour accéder a ce formulaire, il faut cliquer sur rédiger un article dans le cas d'une rédaction et pour le cas d'une modification, il suffit d'appuyer sur le bouton modifier.

Nous avons donc un seul formulaire pour ces deux fonctions ce qui évite une redondance de code.

Si on modifie un article, il faut récupérer tous les champs de l'article, et les insérer dans le formulaire.

Voici une aperçus lorsque l'on veut rédiger un article :



The image shows a web form titled "Ajouter un article" in orange text on a dark blue background. Below the title, there is a label "Titre" followed by a white text input field. Below that is a label "Texte" followed by a large white text area. At the bottom left, there is a label "Publier" with a small square icon to its left, followed by a white text input field. To the right of this field is a button labeled "Parcourir...". At the very bottom, centered, is a large orange button labeled "Ajouter".

Et lorsque l'on modifie un article :



The screenshot shows a web form with a dark blue background. At the top, the title 'Modifier un article' is written in orange. Below it, there is a 'Titre' label and a text input field containing 'dfsdfsdfs'. Underneath is a 'Texte' label and a large text area containing 'efsdgf'. At the bottom left, there is a 'Publier' label and a small checkbox. To the right of the checkbox is a 'Parcourir...' button. At the very bottom, there is a large orange button labeled 'Modifier'.

5. Supprimer_article.php

Comme le titre l'indique, le but de cette page est de supprimer un article. Pour faire fonctionner cette page, il suffit simplement de cliquer sur le bouton supprimer présent sous chaque article. Le seul inconvénient de cette fonction c'est qu'une fois que vous cliquez sur supprimer, cela supprime directement sans vous prévenir !! Voici la requête SQL qui supprime donc un article :

```
DELETE FROM articles WHERE id='$id','article','supprimé')
```

6. Les notifications

Ce système ne demande pas un grand effort de codage mais donne un rendu ergonomique et moderne. Le but de la notification est de prévenir l'utilisateur des différentes actions qui se passent, comme votre article a été ajouté, votre article a bien été modifié, il y a une erreur, connexion impossible, etc....

Cette fonction prend en paramètre une requête SQL ainsi que deux chaînes de caractères. La première chaîne se réfère à la table dans laquelle la requête agit, et la deuxième correspond à l'action de la notification. Ces notifications sont ensuite gérées dans le fichier « notifications.inc.php » dans lequel nous faisons un switch selon le mot utilisé.

7. La connexion

Pour qu'un blog fonctionne bien, il faut des administrateurs pour rédiger du contenu. Pour ce faire, nous avons créé une page de connexion qui permettrait de se connecter pour rédiger des articles. Pour cette fonction, nous utilisons également les cookies dans lesquels on stocke le SID (un identifiant unique créé lors de la connexion). Nous avons également créé un petit script qui permet de vérifier les données du formulaire de connexion.

8. Les fonctions

Nous avons également créé des fonctions qui simplifient le code et le rendent plus lisible. Les fonctions que l'on a créées sont la fonction post qui permet l'obtention de données via la méthode post, la fonction get qui permet l'obtention de données via la méthode get et une fonction qui permet de gérer les notifications.

9. La recherche

Nous avons également implémenté sur le blog une fonction de recherche qui permet aux internautes de rechercher un article rapidement. Le champ de recherche récupère les données et consulte ensuite la base de données pour afficher tous les articles comprenant les termes entrés en recherche.

10. Upload une image

Pour finir les TP, nous avons créé un système qui permet au rédacteur de l'article d'uploader une photo avec un article. Pour ce faire, nous avons dû mettre des limites, comme par exemple le type de l'image (obligatoire une jpeg), une taille maximale de l'image, etc. Ces images sont ensuite stockées dans un dossier. Après de longues heures d'acharnement, l'upload fonctionne bien. Vous pouvez uploader une image et l'image est stockée dans un dossier.

11. Les tags

Le but de cette fonction est que lorsque l'on rédige un article, le blogueur peut y associer un tag. Ce tag sera donc lié à un article. J'ai essayé d'intégrer cette fonction à mon code, mais cela n'a pas vraiment fonctionné. Après de nombreuses heures passées dessus, j'ai donc décidé d'arrêter. Bien sûr, pour prouver que j'ai vraiment travaillé sur cette fonction, j'ai ajouté une page TAGS.PHP qui montre tout le code que j'avais écrit.

Conclusion

J'ai particulièrement adoré la façon dont nous avons élaboré ce blog via le biais de petit exercice puis une correction au tableau, ce qui permettait de corriger rapidement mais également de voir comment d'autres codeurs fonctionnent. Cette méthode d'apprentissage m'était très bénéfique.