

TP WEB-SERVEUR

BERTRAND Anthony, BANDERA Quentin

Table des matières

1	Introduction	1
2	Travail à effectuer	1
3	Travail fourni	1
4	Technologies utilisées	1
5	Démarrer le programme	2
6	Problèmes rencontrés	4
7	Conclusion	5

1 Introduction

Lors de nos études en L3 Informatique, nous avons dû créer un mini-blog en utilisant les technologies web de notre choix. Nous allons très brièvement revenir sur le travail demandé avant de parler du travail effectué ainsi que les choix technologiques. Nous terminerons en mentionnant les problèmes rencontrés ainsi qu’une conclusion sur ce projet avec nos ressentis personnels.

2 Travail à effectuer

L’objectif de ce projet en binôme était de faire un mini-blog ayant les fonctionnalités suivantes :

- Gestion des articles en CRUD (Create, Read, Update, Delete)
- Système de commentaire en CRUD
- Gestion des utilisateurs (pas forcément en CRUD, à vous de voir ce qui est utile)

A ceci s’ajoute quelques règles d’utilisation :

- Seul un administrateur peut créer/éditer/supprimer un article
- Un membre banni ne peut pas créer de commentaire
- Seuls les personnes connectées peuvent envoyer un message

3 Travail fourni

Nous avons implémenté la gestion des articles en ce qui concerne la création, la lecture et la suppression de ces derniers. La modification reste à faire. Bien que cette fonctionnalité semble simple à implémenter, la question de savoir si un administrateur peut modifier un article ne lui appartenant pas se pose. Ce n’est évidemment pas cette question qui nous empêcha de le faire. Nous avons préféré passer cette étape pour se concentrer sur les problèmes survenus (nous développons les détails dans une autre partie) et sur le travail global à effectuer. Une personne non-connectée ne peut que lire les articles et les commentaires. Une personne connectée et étant simplement membre peut en plus envoyer des commentaires. Seul les administrateurs peuvent créer et supprimer des articles.

4 Technologies utilisées

Nous avons décidé d’utiliser php car il a la possibilité d’être utilisé avec du html, permettant de changer la vue suivant certains paramètres. Mais il peut également servir pour les échanges avec le serveur en tant que langage orienté objet.

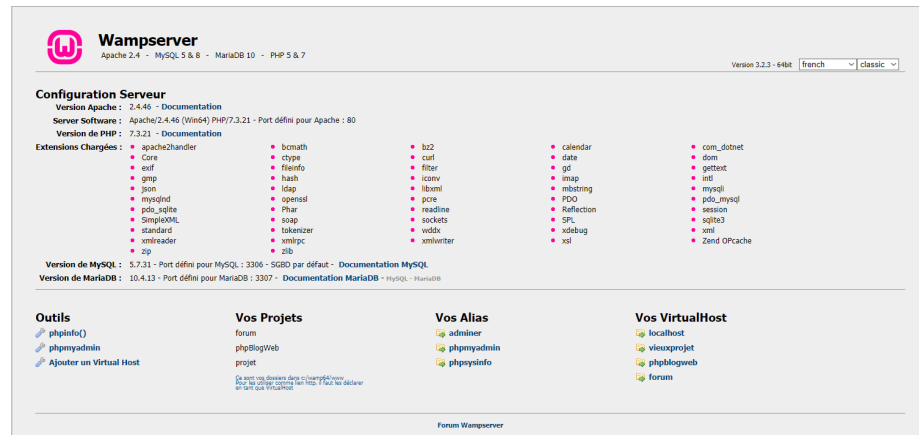
Pour la vue, nous avons utilisé html/css. Nous voulions quelque chose de simple que nous connaissions afin de ne pas passer trop de temps sur cet aspect. Le

but du projet étant de nous évaluer sur nos compétences en web coté serveur. Pour les requêtes en base de donnée, nous utilisons SQL ainsi que phpMyAdmin intégré à Wamp.

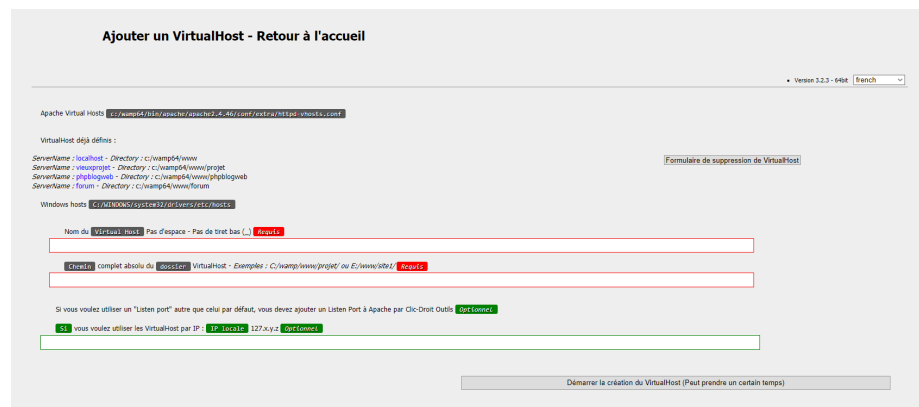
5 Démarrer le programme

(Après avoir importer notre projet, n'oubliez pas de lire le fichier README pour y trouver le compte administrateur).

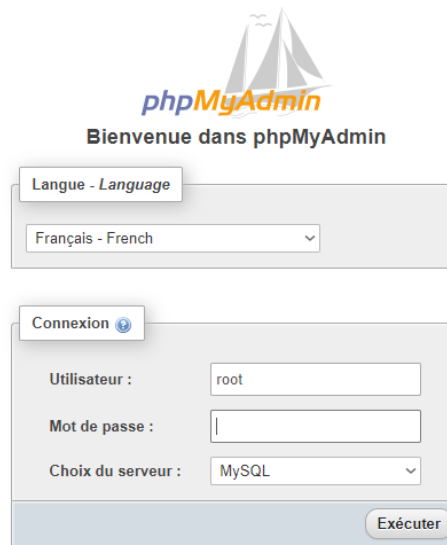
Nous avons travaillé avec le logiciel Wampserver. Pour démarrer notre projet, téléchargez WampServer, déposez le projet dans le dossier "www" de Wampserver, démarrez Wampserver et tapez localhost dans votre navigateur. Vous arrivez sur l'accueil de Wamp.



Cliquez sur "créer une virtualhost" et renseigné le chemin absolu du projet.



Il ne reste plus qu'à redémarrer wamp (clique gauche sur l'icone puis "rafraichir") et de revenir sur l'accueil de Wamp. Un lien pour le projet sera présent. Avant de cliquer dessus, il vous faudra importer la base de donnée fournie avec le projet. Pour se faire, cliquez sur phpmyadmin dans la catégorie "outils" de l'accueil. Vous serez redirigé sur une page de connexion.



phpMyAdmin

Bienvenue dans phpMyAdmin

Langue - *Language*

Français - French

Connexion ⓘ

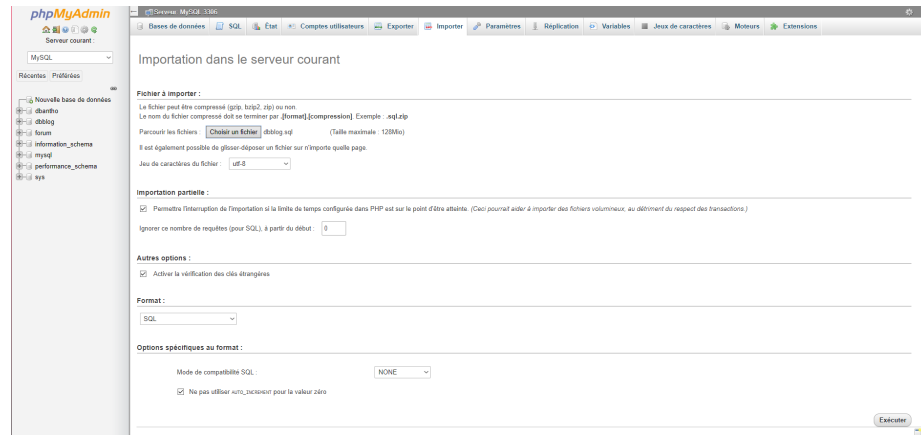
Utilisateur : root

Mot de passe : |

Choix du serveur : MySQL

Exécuter

L'identifiant est "root". Il n'y a pas de mot de passe (par défaut). Une fois sur phpMyAdmin, créez une base en la nommant "dbblog" puis cliquez sur "importer" et insérez le fichier "dbblog.sql" avant d'exécuter. Si tout se passe bien, la base de donnée devrait être opérationnelle. Le reste est pris en charge par notre projet.



6 Problèmes rencontrés

Le problème principal rencontré durant ce projet était le manque d'expériences et de connaissance dans le domaine, ce qui est normal. Ce projet était justement là pour nous donner cette expérience. Il a fallu apprendre sur le tas en se percutant à plusieurs obstacles. Nous avons commis une erreur dès le début de notre projet, mais nous l'avons découverte bien trop tard : Nous avons voulu gérer toutes les actions de l'utilisateur avec notre variable "action" et avec des formulaires. La variables "action" nous a permis d'avoir de la cohérence dans notre code, ainsi qu'une certaine routine dans l'ajout de fonctionnalités. Lorsque l'utilisateur clique sur un bouton, la variable \$action change et suivant sa valeur, le bon contrôleur est appelé pour traiter la demande. La suite est gérée par notre fichier simplemodele.php.

Nous trouvions cette approche très pratique de notre côté. Mais vers la fin du projet, nous nous sommes rendu compte des problèmes engendrés par cette méthode. Tout d'abord, l'utilisation abusive de formulaire en html (notamment des boutons submit) fait que lorsque l'utilisateur souhaite revenir en arrière en utilisant la touche "précédent" de son navigateur, ce dernier lui demandait de confirmer l'envoi du formulaire. Cette confirmation est très frustrante en tant qu'utilisateur. L'idée aurait été de garder les formulaires pour la connexion/l'inscription puis de gérer le reste sans. Mais notre variable \$action ne peut exister que dans un formulaire (une balise input de type hidden pour changer la valeur de cette variable).

Le deuxième problème engendré par cette méthode est que la variable \$action garde sa valeur même après le traitement de la requête. Pour faire simple : Quand un membre envoie un commentaire en cliquant sur "envoyer", la variable \$action prend la valeur "ajoutComm" ce qui permet au contrôleur Ctrl-Membre d'enregistrer le commentaire en base de donnée avant de rafraichir la

page. Si l'utilisateur rafraichit la page manuellement, tout le traitement expliqué ci-dessus est ré-exécuté, dupliquant son commentaire en base de donnée.

7 Conclusion

Dans l'ensemble, nous sommes ravis d'avoir travaillé sur ce projet. Bien que le résultat final ne soit pas à la hauteur de nos espérances, le projet nous a confronté à de multiples obstacles que nous avons dû passer avec plus ou moins d'élégance. Les problèmes majeurs de notre projet nous serviront de leçon pour nos futurs projets dans le domaine du web coté serveur.