

Alexis

N3 Informatique

PLESSIER

08/01/2021

Technologies Web - Serveur

Rendu Projet Blog

## I. Démarrage du site

Pour lancer le blog, il suffit d'ouvrir le fichier **index.php** (**Alexis.blog/public**), l'architecture MVC et les implémentations de chaque fichier devraient suivre. Personnellement, j'ai créé un Virtual Host nommé **alexis.blog** grâce à WampServer et j'ai juste à rentrer ce nom dans mon URL.

⚠ Lors de chaque requête envoyée à la base de données, les fichiers appellent le fichier **connectDb.php** (**app/db**) qui se connecte à une base de données MySQL avec des logs précis. Cependant j'ai extrait cette base en brut dans le fichier **blogdb\_tthd.sql** (**app/db**) pour pouvoir la réutiliser. Je précise également ici que les mots de passes étant hashés, voici les logs des utilisateurs de la base de données :

Pseudo	Mot de passe
Wild	admin
ban	ban
sub	sub

## II. Choix Technologiques

Pour concevoir ce blog, j'ai décidé de coder comme suit :

- **Côté client** : HTML et CSS, ce qu'il y a de plus basique.
- **Côté serveur** : PHP, c'est un des langages dynamiques les plus répandus, je n'avais jamais utilisé ce langage, j'ai appris à en maîtriser les bases et son apprentissage est plutôt simple et intuitif mais possède également une bonne synergie avec mon SGBD.
- **SGBD** : MySQL, car au-delà du fait qu'il soit également très répandu, il combine bien avec PHP et le serveur utilisé et qu'il possède une interface graphique (*phpMyAdmin*) qui nous permet de consulter les changements dans notre base de données très rapidement et ainsi gagner du temps sur une possible erreur.

- **Serveur** : Apache (le tout grâce à WampServer) car le plus compatible avec les langages énoncés précédemment et le plus utile pour l'environnement de travail que j'utilise actuellement qui est Windows.

Pour l'architecture et la conception du blog, j'ai utilisé le modèle MVC (*Modèle-Vue-Contrôleur*) comme vu en cours car il correspond bien au projet et c'était une bonne occasion de travailler dessus.

△ Comme dit plus tôt, j'ai utilisé le logiciel WampServer pour mon projet et il présente quelques dysfonctionnements, notamment sur les *includes* et *require* en PHP ! C'est pourquoi j'ai également inclus le blog dans une architecture non MVC et un peu « sale » nommé **Alexis.blog\_Bis** au cas où des erreurs venaient à apparaître lors de la consultation du blog !

### III. Philosophie du blog

Lors de la création du blog, j'ai décidé de partir sur une page d'accueil simple présentant les différents articles avec leur auteur et date de création. On peut accéder à l'article et ses commentaires en cliquant dessus quel que soit le statut de l'utilisateur. Un utilisateur peut être banni (**ban**), abonné (**Subscriber**) ou bien administrateur (**admin**). J'ai donc légèrement modifié le système CRUD présenter dans le sujet et ajouté quelques contraintes :

- **Ban** : Un utilisateur peut être banni par un administrateur via une interface (que nous verrons plus tard). Dès lors qu'il est banni, un utilisateur ne peut plus publier d'article ni de commentaire, il ne peut que lire le contenu du blog. Il pourra cependant voir son statut être modifié par un administrateur dans l'avenir, ce n'est pas impossible.
- **Subscriber** : Un utilisateur Subscriber est le statut basique d'un utilisateur lorsqu'il crée son compte. Ce statut lui permet de consulter les articles et les commentaires mais aussi d'en publier. Le propriétaire d'un article ou d'un commentaire peut modifier ou supprimer son commentaire comme bon lui semble.
- **Admin** : Un utilisateur administrateur est un statut attribuer par un autre administrateur (à par le premier évidemment). Ce statut possède les mêmes

caractéristiques qu'un Subscriber à une différence prête que nous verrons après.

Tous les utilisateurs connectés auront accès à un onglet « **mon compte** » leur permettant d'avoir quelques informations sommaires sur leur compte, la possibilité de supprimer leur compte, ainsi que la liste de leurs articles. La suppression de leur compte entrainera la suppression de toute traces sur le blog. Si un utilisateur supprime son compte, cela supprimera ses articles et commentaires. C'est cette même page (**compte.php**) qui est utilisé pour avoir les informations d'un autre utilisateur dans le tableau des articles de la page d'accueil par exemple. Cependant, si l'utilisateur est un **administrateur**, il aura une interface lui permettant de changer le statut de n'importe quel personne (hors Admin, des querelles sont vites arrivées sur internet) mais également de supprimer un autre utilisateur (hors admin également) effaçant toute trace de ce dernier.

△ La suppression d'un article entraine bien sur la suppression de tous les commentaires liés à cet article.

#### IV. Problèmes rencontrés

Lors de ce projet, j'ai rencontré quelques problèmes tous plus différents les uns que les autres :

- Lors du script de la connexion à ma base de données en PHP, l'adresse 127.0.0.1 à été préférée à localhost.
- La conception UML du projet à dû être revue plusieurs fois pour obtenir la vision que j'avais du blog.
- La notion d'URL rewriting m'as pris du temps à prendre en main et je ne suis pas sur et certain de pouvoir l'exploiter comme je le voudrais à l'heure qu'il est.
- Le problème d'*include* et *require* en PHP m'a pris du temps et un peu découragé dans le sens où mon projet était terminé et qu'il ne me restait plus qu'à mettre les fichiers dans les bons dossiers pour respecter une architecture MVC.

## V. Conclusion

Pour conclure, ce projet à était une découverte totale des technologies côté serveur pour ma part. Manipuler des bases de données à été satisfaisant ainsi que l'apprentissage des langages que cela implique. J'ai pu me rendre compte à quel point les technologies web peuvent être vaste et l'investissement que nécessite un projet comme celui-ci.