

Projet PHP

Nous allons construire un mini-blog.

Ce qu'on va utiliser

Ce projet va utiliser beaucoup d'éléments abordés depuis le début du cursus :

- HTML, CSS
- Algorithmique
- PHP
- MVC
- CRUD
- ...

Rappels

- Vous avez accès à des fonctions pour déboguer, comme `var_dump`. N'hésitez pas à les utiliser pour vérifier les valeurs de vos variables si besoin est !
- Chercher sur Google n'est pas une honte ! Vous avez le droit de le faire ! (Après, évitez le copier-coller sans réfléchir, souvent ça ne fonctionne pas^^')
- Codez proprement ! Pensez à indenter votre code, à bien nommer vos variables et vos fonctions, ...
- Faites ce que vous pouvez dans le temps que vous avez

Jour 1

Mise en place de l'environnement

1. VS Code et serveur local

Créer un dossier pour le projet, l'ouvrir dans VS Code et ouvrir l'URL dans le navigateur.

2. Architecture MVC

Mettre en place une architecture MVC "classique".

3. Base de données

Importer la base de données fournie via PHPMyAdmin.

Une route d'accueil

En respectant l'architecture MVC, mettre en place une route d'accueil, qui contient du lorem ipsum et un menu de navigation.

Le CRUD des articles

1. Les Retrieve

a. En liste

Afficher une route qui liste les articles de la table **article** de la base de données.
Ajouter un lien vers cette route dans le menu de navigation.

b. Au détail

Créer une route qui affiche **un** article de la table **article** de la base de données. Pour savoir quel article afficher, la route reçoit un paramètre d'URL.

Ajouter les liens adéquats dans la route qui liste les articles pour pointer sur cette page de détails.

2. Le Create

Créer une route qui propose un formulaire pour ajouter un article.

Ajouter un lien vers cette route dans le menu de navigation.

A la soumission du formulaire, l'article doit être inséré dans la base de données.

3. Le Update

Créer une route qui propose un formulaire pour modifier un article. Pour savoir quel article modifier, la route reçoit un paramètre d'URL.

A la soumission du formulaire, l'article doit être modifié dans la base de données.

Ajouter un lien vers le formulaire de modification à chaque article (dans la liste comme dans le détail).

4. Le Delete

Créer une route qui supprime un article. Pour savoir quel article supprimer, la route reçoit un paramètre d'URL.

Ajouter un bouton qui pointe vers la route de suppression à chaque article (dans la liste comme dans le détail).

Jour 2

La connexion utilisateur

1. La route de connexion

Créer une route de qui propose un formulaire de connexion.

A la soumission du formulaire, on vérifie l'existence en base de données d'un utilisateur avec le même identifiant. S'il existe, on vérifie son mot de passe (hashé).

Si l'utilisateur est identifié, une session est créée et il est redirigé vers la route d'accueil.

Sinon, il est redirigé vers la route de connexion.

Ajouter un lien vers cette route dans le menu de navigation.

2. La route de déconnexion

Créer une route dont le seul rôle est de détruire la session d'un utilisateur.

A la fin, l'utilisateur est redirigé vers la route d'accueil.

Ajouter un lien vers cette route dans le menu de navigation.

3. La route de création de compte

Créer une route de qui propose un formulaire de création de compte.

A la soumission du formulaire, l'utilisateur doit être inséré dans la base de données.

A la fin, l'utilisateur doit être redirigé vers la route de connexion.

Ajouter un lien vers cette route sur la page de connexion.

4. Se souvenir de moi

Ajouter une checkbox au formulaire de connexion.

A la soumission du formulaire, si la checkbox est cochée, en plus de créer une session, un cookie doit être créé, durer un mois et contenir l'id de l'utilisateur qui vient de se connecter.

La gestion de l'utilisateur connecté

1. Dans le menu de navigation

a. Les bons liens

Modifier le menu de navigation pour qu'un utilisateur connecté ne voie pas le lien pour se connecter et qu'inversement un utilisateur non connecté ne voie que le lien pour se déconnecter.

b. L'utilisateur connecté

Modifier le menu de navigation pour afficher, en marge des liens de navigation, le pseudo et l'avatar de l'utilisateur connecté.

2. Pour gérer les articles

a. L'affichage

Modifier l'affichage des liens de création / modification / suppression d'article pour qu'ils ne s'affichent que si l'utilisateur a le rôle `admin`.

b. La logique

Empêcher le fonctionnement des pages de création / modification / suppression d'article si l'utilisateur n'a pas le rôle `admin`.

Jour 3

Les commentaires

1. Afficher les commentaires

Afficher, sur la page présentant **un** article, les commentaires qui lui sont liés.

Pour savoir quels commentaires sont liés à quels articles, la colonne **id_article** de la table **commentaire** reprend l'id de l'article qui est lié au commentaire.

2. Poster un commentaire

Faire un formulaire en bas de la page de chaque article, qui permet d'écrire un commentaire.

A la soumission de ce formulaire, le commentaire doit être inséré en base de données.

A la fin, on redirige vers la page de l'article de laquelle il vient.

3. Lier à l'article

Lorsqu'un commentaire est posté, il n'est, par défaut, pas lié à un article.

Trouver un moyen d'indiquer à quel article est lié le commentaire nouvellement posté.

4. Restreindre à l'utilisateur connecté

Bloquer l'affichage et la logique de postage de commentaire afin que seul un utilisateur connecté puisse poster un nouveau commentaire.

Les bonus

Si vous êtes arrivé jusqu'ici : bravo ! C'est du bon boulot !

Vous pouvez à présent faire ce qui vous passe par la tête pour améliorer le blog.

Quelques suggestions :

- Afficher le nom de l'auteur de l'article
- Remplir le nom de l'auteur à la création d'un article
- Afficher les auteurs des commentaires
- Remplir l'id de l'auteur d'un commentaire au postage
- Ne pas afficher les articles si la date de publication n'est pas encore atteinte
- Afficher les articles triés par date décroissante
- Faire du style CSS
- ...