

Projet TP

Créer un système de blog Simple

Objectif principal : Créer une version simple d'un site permettant de partager des messages avec d'autres utilisateurs. La fonctionnalité de ce site ressemblera aux premiers sites de ce type apparus vers la fin des années 1990. On peut décrire ces vieux blogs comme une version extrêmement basique d'un Twitter ou d'un mur Facebook.

Cahier des charges

Fonctionnalités exigées

Vous devez réaliser ces objectifs. Une démo que je vous montrerai en TD/TP vous aidera à mieux les visualiser

- **Page d'accueil** : me propose de découvrir les blogs d'un certain nombre d'utilisateurs choisis aléatoirement, ou de login avec un nom utilisateur. Un lien me propose également de créer un compte (*cf page. Création de compte*)

Si l'utilisateur effectue un login réussi, il est automatiquement redirigé vers le blog dont il est le propriétaire (*voir page de consultation de blog*)

- **Page de création de compte** : demande de saisir un nom et un mot de passe. Le mot de passe doit être saisi deux fois (« *confirmer le mot de passe* »).

Une fois le formulaire rempli correctement et envoyé, un utilisateur est créé dans la base de données, et l'utilisateur est ramené sur la page d'accueil.

Un message doit informer de la création réussie du compte.

Si un formulaire incorrect est envoyé (*nom ou mot de passe vide, confirmation ratée...*), l'utilisateur est ramené sur la page.

- **Page de consultation d'un blog** : tous les blogs sont affichés par la même page PHP, qui utilise un paramètre transmis par GET pour identifier le blog à afficher (*ainsi, on peut partager l'adresse d'un blog spécifique avec un autre utilisateur*)

Chaque page montre les 10 derniers messages postés par le propriétaire du blog, rangés du plus récent au plus ancien.

Au niveau de la base de données, les messages ont donc au minimum un titre, un texte de contenu, une date de post et un propriétaire/auteur.

Si un utilisateur connecté visite le blog dont il est propriétaire, l'affichage est modifié pour montrer un bouton « Nouveau post », et chaque message se voit lui-même ajouter deux boutons : modifier et effacer.

Cliquer sur « modifier » ouvre la page créer/modifier un post. Cliquer sur « effacer » ouvre la page pour demander confirmation de l'effacement.

(Si vous êtes à l'aise, vous pouvez utiliser Javascript pour ouvrir un pop-up de confirmation à la place, mais ce n'est pas exigé, voire section « Fonctionnalités pour aller plus loin »)

- **Page de création/modification d'un post** : Une page contenant un formulaire pour taper le titre et le contenu d'un post/message. Lorsque le formulaire est envoyé, le post est envoyé et ajouté au blog du propriétaire.

La différence visible entre « ajouter » et « modifier » est assez restreinte : dans un cas, les champs sont initialement vides, dans un autre, ils sont préremplis avec les valeurs précédentes.

A noter que j'ai placé ces deux fonctionnalités ensemble à cause de la similitude visuelle et fonctionnelle de la perspective de l'utilisateur. Mais si vous préférez avoir deux pages qui certes se ressemblent mais sont séparées (*une pour ajouter et une pour modifier*) vous en avez le droit.

- **Page de confirmation d'effacement** : Comme ce n'est pas certain que Javascript aura été vu en cours avant la fin de ce projet, on va demander la confirmation de l'effacement d'un post avec une page dédiée.

La page doit montrer le post qu'on s'apprête à effacer, pour éviter les erreurs/étourderies.

Si on confirme, le post est effacé et on revient à l'affichage du blog. Si on annule, on revient aussi au blog, mais le post n'est pas effacé.

Fonctionnalités « pour aller plus loin »

Non seulement elles ne sont pas exigées, mais ce sont des suggestions destinées aux élèves qui n'ont aucune difficulté avec les objectifs de base. En réalisant ces objectifs, vous prenez de fait de l'avance pour le final, où on vous demandera ces choses

- Archives et navigation au-delà de 10 messages : Dans le projet exigé, un blog n'a besoin que d'afficher les 10 messages les plus récents, et les autres sont donc inaccessibles. Vous pouvez changer cela en créant un système de « pages » à l'intérieur de chaque blog (*page 2 affiche les messages 11 à 20, etc...*)

A noter que ces « pages » ne sont pas des fichiers individuels : c'est notre page « d'affichage de blog » qui devra être modifiée pour prendre un ou des paramètre(s) supplémentaire et adapter l'affichage en conséquence.

- Catégories de messages/tags/mots-clés : Offrir la possibilité à l'utilisateur de ranger ses messages dans des catégories prédéterminées (*du genre « Travail », « Loisirs », etc. ...*)

- Utiliser Javascript pour éviter la page « confirmer effacement »: Javascript vous permet d'afficher une pop-up qui demande confirmation au lieu de forcer l'utilisateur sur une autre page. Implémenter cette fonction rendra le site plus agréable à utiliser.

- Rendre la saisie de message plus conviviale et intéressante : Il existe des « éditeurs de texte Javascript » gratuits qu'on peut réutiliser dans ses projets, pour permettre une saisie en gras, italique, etc. (*un peu comme dans Word*)
Vous pouvez essayer de vous procurer un de ces composants et de l'intégrer dans votre page.

- Uploader des images : Vous pouvez ajouter la possibilité pour un utilisateur d'importer des images dans son blog. Une image par message et/ou un avatar par exemple (*plusieurs images par message posent des difficultés de mise en page donc je vous déconseille de vous y essayer pour ce projet « rapide »*).

- Fonctionnalités de recherche : Ajoutez un formulaire sur la page d'accueil pour rechercher des blogs en fonction du nom de leur auteur. Vous pouvez aussi ajouter une fonction recherche pour les messages à l'intérieur d'un blog.