

Projet TP

Créer un site d'« échange de messages » simple

Objectif principal

Créer une version simple d'un site permettant de partager des messages avec d'autres utilisateurs. « Site de messages » est une idée qui peut correspondre à plusieurs archétypes que vous connaissez :

- Mur facebook ou fil Twitter (*messages regroupés autour de leur auteur*)
- Forum de discussion (*files créés par les utilisateurs où plusieurs personnes discutent*)
- Site de Blogs (*Pages où un utilisateur écrit des articles sur ce qui le passionne*)
- etc.

Le but de ce projet est de réaliser une version simple, « primitive », de ces systèmes très répandus pour la communication en ligne.

Autres Objectifs

Donnez un « thème » à votre site : orientez votre site vers un domaine spécifique : forum de passionnés de foot, murs pour présenter un CV... il y a beaucoup d'options. N'hésitez pas à choisir un sujet qui vous intéresse et vous motive.

Si vous n'avez vraiment pas d'idée, faites une parodie d'un site existant (genre Twitter peut devenir «Touiteur »)

Penser aux visuels et à l'ergonomie : Sans exiger de vous des chefs-d'œuvre ou des interfaces optimisées pour économiser le maximum de clics, nous attendons un minimum sur la qualité visuelle et la facilité d'utilisation du site.

Les sites qui sont jolis et ergonomiques recevront un bonus allant jusqu'à 3 points.

Valorisation de la recherche et l'expérimentation : Les exigences décrites dans le cahier des charges ci-après sont les exigences **minimum**. Si elles sont parfaitement exécutées, vous aurez 14/20.

Pour monter au-delà, vous devrez aller plus loin, en répliquant des fonctionnalités communes sur le net, mais qui ne sont pas absolument obligatoires ici. La fin du document vous donne quelques idées sur ces moyens « d'aller plus loin ».

Rester sur HTML/CSS/PHP/Javascript : Si nous vous encourageons à aller plus loin, nous ne voulons pas non plus que vous sortiez des outils sur lesquels ce projet cherche à évaluer votre maîtrise!

Comme pour le micro-site, **le seul outil non vu en TD qui est autorisé est FLEX.**

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 :

- elle me propose de découvrir des contenus disponibles sur le site (*différents murs sur un facebook-like, threads sur un forum, etc...*)
- Un lien me propose également de créer un compte, et m'amène sur la page correspondante (*cf page. Création de compte*)
- Si l'utilisateur effectue un login réussi, il est automatiquement redirigé vers la page dont il est le propriétaire (*son mur/fil*).
- Sur les sites de types forums, il sera plutôt redirigé vers une page qui permet de voir tous les threads (*ou catégories de threads si vous en implémentez*).

- 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.
- L'utilisateur peut uploader un avatar. Ce dernier est automatiquement redimensionné à un format prédéterminé et adapté. (*du genre 256x256 pixels*)
- Si un formulaire incorrect est envoyé (*nom ou mot de passe vide, confirmation ratée...*), l'utilisateur est ramené sur la page.

- Page de consultation de contenu :

- Tous les contenus sont affichés par la même page PHP, qui utilise un paramètre transmis par GET pour identifier le contenu exact à afficher cette fois-ci (*ainsi, on peut partager l'adresse d'un contenu spécifique avec un autre utilisateur*)
- Les messages sont rangés dans un ordre pertinent pour le type de site (du plus récent au plus vieux sur les « réseaux sociaux », le contraire sur les forums)
- Au niveau de la base de données, les messages ont au minimum un titre, un texte de contenu, une date de post et un propriétaire/auteur.

- Les posts ayant une image doivent l'afficher. Pour l'objectif minimum, elles peuvent être affichées toujours à la même taille, ce qui simplifie le CSS.
- Si un utilisateur connecté est sur une page de contenu où il a le droit de poster, l'affichage de la page est modifié pour montrer un bouton « Nouveau post » *(ou autre nom selon le type de site : nouveau message, nouvel article, etc...)*
- Si l'utilisateur est connecté, chaque message/post dont il est l'auteur se voit 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 une pop-up de confirmation à la place, mais ce n'est pas "le minimum")

- 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 pré-remplis avec les valeurs précédentes.
- Donnez la possibilité d'uploader des images, et aussi de les modifier. Assurez vous qu'une modification détruit bien l'ancienne image dans le serveur !
- A noter que j'ai placé ces deux fonctionnalités ensemble à cause de la similitude visuelle et fonctionnelle, vu 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 :

- Une solution Javascript sera jugée meilleure, mais ne fait pas partie du strict minimum, qui est : demander la confirmation de l'effacement d'un post/message 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 la page précédente. Si on annule, on revient aussi à la page précédente, mais le post n'est pas effacé.

Quelques idées « pour aller plus loin »

Liste **non exhaustive** de choses que vous pouvez faire pour aller plus loin que les objectifs minimum, et avoir la possibilité d'une note supérieure à 14 sur 20

- Répartir le contenu en pages : Si une page devait afficher beaucoup de contenu, implémentez un système de pages qui permet, par exemple, de voir 10 posts à la fois. Vous pouvez implémenter un système « page suivante/précédente », ou alors un système AJAX qui va rajouter des posts et prolonger la page (ou bien en cliquant un bouton, ou bien automatiquement en arrivant en bas comme sur Facebook)

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

Sur un site de type Forum, ce seront les threads qui auront une catégorie, plutôt que les messages eux-mêmes.

- Utiliser Javascript pour l'esthétique et/ou l'ergonomie : Javascript a de nombreux usages potentiels pour rendre votre site plus pratique ou plus glamour ! Les options évidentes : enlever la page d'effacement de post et la remplacer par une pop-up, animations des boutons ou des textes (sans abuser!), etc...

- 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.

- Fonctionnalités de recherche : Ajoutez un formulaire sur la page d'accueil pour rechercher des contenus en fonction du nom de leur auteur. Vous pouvez aussi ajouter une fonction recherche pour les messages à partir de leur contenu et/ou seulement sur une page spécifique.

- Améliorer l'upload et l'affichage d'avatar/d'images : Il y a plusieurs manières d'améliorer l'upload des images dans votre site : empêcher l'upload d'images de taille trop grande, adapter l'affichage à des images de proportions différentes, etc.

- Utiliser AJAX pour éviter les changements de page : AJAX peut être utilisé pour limiter les changements de page et rendre la navigation plus ergonomique. L'écriture de message, pourrait, par exemple, être ajoutée dynamiquement à une page. Et ce n'est qu'une possibilité parmi beaucoup !