

# INTERNET ET OUTILS

Pierre Gimalac, Stéphane Xu

*Licences de Mathématiques et d'Informatique*

*Janvier - Mai 2017*

Rapport du projet semestriel

# 1 Description du site et de ses fonctionnalités

## 1.1 WhyNot

Notre site est nommé *WhyNot*. Pourquoi *WhyNot* ?

Pourquoi pas ?

## 1.2 Description du site

### 1.2.1 Principe du site

Il s'agit d'un genre de blog ou de forum public dont le but est d'écrire et de publier des articles dans des catégories prédéfinies pour que tous les visiteurs (inscrits ou non) puissent les lire. On peut aussi rechercher dans le site des articles, messages ou utilisateurs.

Une fois connecté, on peut également commenter et noter les articles postés, en écrire ou échanger en privée des messages avec d'autres utilisateurs.

### 1.2.2 Page d'accueil

La page d'accueil du site affiche pour chaque catégorie contenant des articles un aperçu des 3 derniers articles ayant été postés.

### 1.2.3 Menu de navigation

La partie gauche de la page d'accueil (ainsi que de la plupart des pages) est un menu de navigation qui permet d'accéder rapidement aux 10 derniers articles postés ainsi qu'à chaque catégorie (où l'on peut voir l'ensemble des articles appartenant à cette catégorie).

Ce menu de navigation contient également un 'mini-formulaire' de recherche permettant de rechercher un article par son titre, ainsi qu'un lien pour accéder à la page de recherche (dont le formulaire est plus complet).

Enfin, si l'on est connecté, ce menu contient un lien vers la messagerie (qui permet de communiquer en privée avec un autre utilisateur).

### 1.2.4 Menu complémentaire

La partie droite de la plupart des pages contient des statistiques sur le site (nombre de pages chargées, d'articles postés, de personnes connectées,...) ainsi que des 'publicités' (pour le *CROUS*, *Paris Diderot* ou encore *OpenClassroom*).

### 1.2.5 En-tête et pied de page

L'en-tête des pages du site contient des liens, si l'on n'est pas connecté ce sont des liens pour se connecter, se créer un compte et signaler que l'on a oublié ses identifiants, sinon ce sont des liens pour voir son compte, accéder à l'espace connecté et se déconnecter.

Le pied de page contient divers liens vers une page annexe contenant des informations sur le site ("*Qui sommes-nous ?*", "*Nos partenaires*", "*Code source*", "*Transparence*", "*Règle du site*", "*Foire aux questions*", "*Wiki*", "*Contactez-nous*").

### 1.2.6 Création de compte

Le formulaire de création de compte demande le nom et le prénom de l'utilisateur, ainsi qu'une adresse mail et un pseudonyme. Nous n'avons pas eu le temps d'afficher les contraintes à respecter en javascript (voir *difficultés* page 7) donc nous avons fait une page spéciale pour les afficher (il faut cliquer sur "Créer un compte" une fois dans le formulaire).

### 1.2.7 Espace connecté

L'espace connecté est disponible pour les personnes connectées et affiche la liste des articles en attente de validation (que l'on peut lire mais ni noter ni commenter tant qu'ils ne sont pas publiés) ainsi que la liste des articles que l'utilisateur est en train d'écrire.

Il possède également un lien pour commencer à écrire un nouvel article.

### 1.2.8 Messagerie

La messagerie affiche en premier lieu la liste des utilisateurs auxquels l'utilisateur connecté a déjà envoyé un message, ainsi que le nombre de messages non lus que chaque utilisateur lui a envoyés (affiché entre parenthèses à côté du pseudonyme) et la date du dernier message échangé (l'heure si ce message date de moins de 24 heures et la date sinon).

Le menu contenant la liste des utilisateurs contient également en haut un formulaire de recherche d'utilisateur (qui renvoie sur la page de recherche).

Si un *id* est fourni en URL la liste des messages envoyés à la personne dont l'identifiant est indiqué est également affichée ainsi qu'un formulaire servant à envoyer un message à cette personne.

La messagerie ne contient (presque) pas de *javascript* (le peu qui soit utilisé sert à arriver en bas de la liste des messages), ainsi poster un message nécessite de charger la page (et voir de nouveaux messages de même).

### 1.2.9 Recherche

La page de recherche contient un formulaire permettant de rechercher un article par son titre ou son contenu mais aussi un utilisateur ou un message, en choisissant des paramètres de recherche très variés ainsi que l'ordre d'affichage des résultats (pouvant être la date, le titre d'un article, le nombre de commentaires ou de lectures d'un article, la note attribuée à un article, etc).

Les paramètres (variant selon ce que l'on recherche grâce à du *javascript*) sont le nombre de commentaires et d'articles (écrit par un utilisateur par exemple), il y a également des conditions temporelles : on peut indiquer que l'on cherche un article, message ou utilisateur plus récent ou plus ancien qu'un certain temps ou même si l'on est connecté plus récent ou plus ancien que l'avant dernière connexion (qui est sauvegardée dans la base de données).

De plus, on peut indiquer un auteur (si l'on cherche un article ou un commentaire) ainsi que sélectionner des catégories pour un article.

## 2 Inspirations

Le principe du site est inspiré d'un site qui n'existe plus et où l'on pouvait de même poster des articles une fois connecté, il y avait également une page 'espace connecté'.

Pour l'accueil du site nous nous sommes inspirés de *zone-telechargement* (pour l'affichage des catégories et de leurs derniers éléments) de même que pour le formulaire de recherche.

La messagerie est inspirée de celle de Facebook (affichage des contacts à gauche avec formulaire de recherche en haut et affichage des messages à droite, sur la droite de cette partie si l'on est l'auteur et sur la gauche sinon ; avec les plus récents en bas).

Enfin nous nous sommes inspiré de différents styles donnés en exemple dans des sites pour apprendre *PHP* ou *CSS* (les couleurs principales du site ou encore la forme des tableaux dans les menus sur les cotés).

## 3 Fonctionnement du site

### 3.1 Pages

Le site possède 3 'super-pages' contenant toutes les autres pages et renvoyant l'utilisateur sur la bonne 'sous-page' (selon la valeur de DO fournie par l'url).

- La page d'**index** contient les pages permettant d'afficher les articles, catégories et utilisateurs, ainsi que la page de recherche.
- La page de **login** contient les pages en lien avec le compte lui-même, c'est-à-dire les pages de connexion, déconnexion, création de compte, oubli des identifiants, affichage et changement des informations du compte et enfin de messagerie.
- La page d'**espace connecté** contient les pages en lien avec les actions effectuelles si l'on est connecté : écrire un article et lire un article en attente de validation.

Nous avons choisi de répartir le site en de nombreuses 'sous-pages' qui sont incluses sur demande, ce qui permet d'éviter d'avoir plusieurs fois le même code dans plusieurs pages.

Nous avons aussi fait une *page 404*, cependant dans les cas où un article demandé n'existe pas par exemple on préfère afficher que l'article n'existe pas plutôt que de renvoyer sur cette page 404 qui indique simplement que la page n'existe pas.

### 3.2 Fonctionnement

Sans être connecté, on peut simplement lire des articles, avoir un aperçu du compte d'un utilisateur et faire des recherches sur le contenu du site.

Si l'on est connecté, on peut écrire un article, commenter et noter les articles publiés existants, ou encore échanger des messages avec les autres utilisateurs.

Il existe cependant une particularité : certains comptes sont des 'master', c'est à dire qu'ils peuvent publier un article en attente de validation (ou bien le supprimer s'ils jugent qu'il n'est pas publiable) mais également supprimer des commentaires et même des utilisateurs (et en même temps leurs articles, messages, notes et commentaires).

## 4 Expérience et difficultés

### 4.1 Création du site

Nous avons fait le site tout au long du semestre en fonction de ce que l'on voyait en cours, mais parfois nous avons utilisé des cours trouvés sur internet pour avancer plus vite.

En effet nous avons fait tout ce qui nécessitait uniquement de l'*HTML*, du *CSS* et du *PHP* au début (donc sans la base de données).

Mais à un moment nous n'avions plus rien à faire (et il restait plusieurs semaines avant le début du cours sur *SQL*) donc nous avons appris les bases de *SQL* pour pouvoir continuer, ce qui explique que nous ayons utilisé *PDO* puisque nous ne savions pas que nous allions utiliser *mysqli* en cours.

### 4.2 Choix techniques

#### 4.2.1 Feuilles de style

Le site ne contient qu'une seule grande feuille de style contenant la plupart des styles (certains ont été mis directement dans le code *html*).

Dans cette feuille de style chaque page est séparée pour que ce soit plus clair, de plus il y a des styles ne s'appliquent que si l'écran est trop petit. Ainsi on obtient un site visuellement tout à fait différent si l'écran est trop petit.

#### 4.2.2 Requêtes SQL

Nous avons principalement utilisé des requêtes préparées qui permettent d'exécuter directement les requêtes en donnant les arguments sans avoir à faire de traitement.

De plus, nous avons choisi de faire de nombreuses requêtes plutôt qu'un nombre moindre de requêtes plus modulables. Bien que ce ne soit pas très efficace (il y a de nombreuses requêtes) nous avons trouvé que ce serait plus simple comme ça.

En effet nous pouvons ainsi exécuter les requêtes sans donner trop d'argument (puisque chaque requête est faite pour une action précise).

#### 4.2.3 Base de données

La base de données du site contient 7 tables : une table pour les utilisateurs, une table pour les articles, une pour les catégories, une pour les commentaires, une pour les messages, une pour les notes et enfin une pour stocker les personnes connectées (pour les statistiques du site par exemple).

Ces tables sont reliées les unes aux autres par des clés étrangères (un article doit être écrit par un utilisateur existant par exemple) et possèdent toutes des clés primaires pour identifier chaque élément qu'elles contiennent.

Les clés étrangères permettent par exemple que lorsque l'on supprime un utilisateur, ses articles, commentaires, notes et messages soient supprimés avec lui.

### 4.3 Sécurité

Pour coder les mots de passes dans la base de données, on utilise une clé située dans un fichier qui sert de salt pour coder le pseudonyme de l'utilisateur. On utilise ensuite le résultat obtenu comme salt pour coder le mot de passe lui-même.

Ainsi ce codage est très sécurisé puisque le salt du mot de passe dépend du pseudonyme mais également du contenu d'un fichier situé sur le serveur. Il faut donc avoir accès à la base de données, à la clé et connaître la méthode de codage pour avoir une chance de trouver un mot de passe.

Nous avons utilisé un éditeur de texte enrichi pour l'écriture des articles, ce qui permet d'avoir des articles plus complexes mais n'est pas très sécurisé (le texte final est en html) puisqu'on affiche tel quel le texte fourni à la fin et que l'on a pas vraiment de manière de vérifier que ce qui est fourni vient bien de l'éditeur de texte (et nous n'avons pas trouvé la fonction javascript de l'éditeur servant à passer d'un texte fourni en entrée au texte html dans ses fichiers).

### 4.4 Difficultés

La principale difficulté que nous avons rencontrée fut d'envoyer des mails (nous n'avons pas réussi), ce qui explique qu'aucun mail ne soit envoyé pour valider le compte et que le formulaire d'oubli des identifiants ne soit pas réellement traité.

De plus, nous avons du utiliser javascript à certains moments et parfois nous avons renoncé à faire certaines choses qui auraient nécessité l'utilisation de javascript : par exemple afficher les conditions de validité dans le formulaire de création de compte pour montrer en direct à l'utilisateur si ce qu'il a rempli est valide ou non ou encore pouvoir complètement changer les couleurs du site et voir directement les changements.