

## **Création du site « Enchère est en hausse ? » Statistiques, Forums et FAQ**

### **Objectifs du projet :**

- ‡Mettre en place les statistiques sur les connexions au site, sur les utilisateurs et les objets.
- ‡Mettre en place les forums avec l'ouverture automatique d'un forum dès la mise en vente d'un objet.
- ‡Mettre en place les Foires Aux Questions, ainsi que leurs mises à jour par l'administrateur

### **A / La Foire Aux Questions**

Elle se compose de 3 éléments : la FAQ à proprement parler, le formulaire de mise à jour et le script de mise à jour.

#### **1) FAQ : la présentation**

La page de présentation se décompose en 2 parties : La liste des questions en haut de la page et la question et sa réponse en bas de la page.

La question ( en haut de page ) est un lien pointant vers la même page mais au niveau de la question ( au bas de la page ) et de sa réponse.

Après chaque réponse, un lien permet de remonter à la liste des questions en haut de la page.

#### **2) FAQ : Le formulaire**

Le formulaire de mise à jour est disponible au niveau de la partie administration du site.

Composé de 2 champs ( question et réponse ), il appelle le script de mise à jour.

#### **3) FAQ : Le script**

Le script est écrit en Perl pour des raisons de facilité et d'adéquation entre le langage et l'objectif du script.

Le script récupère d'abord les données du formulaire et vérifie que les champs aient été non seulement remplis mais aussi correctement, pour des raisons de sécurité ( même si ce ne sera que l'administrateur qui y aura accès ).

Une fois que les données sont vérifiées, le script ouvre le fichier de présentation de la FAQ et le parcourt. Lorsqu'il rencontre la première question, il insère la nouvelle et crée le lien. Lorsqu'il rencontre la première réponse, il insère la nouvelle question et sa réponse, ainsi que le lien pour remonter sur la liste des questions.

## B / Les statistiques

### 1 ) Sur les connexions

Nous avons fait une petite recherche sur le web parce que des outils avaient déjà été développés. Nous avons donc bon nombre d'outils à notre disposition avec leurs avantages et leurs inconvénients :

- ‡Webtrends ( payant )
- ‡LogMetrix ( payant )
- ‡Xiti ( pas très complet )
- ‡Wysistat ( inscription nécessaire )
- ‡Webalizer ( complet, indépendant, gratuit )

Nous avons donc choisi d'installer Webalizer.

Webalizer est un logiciel qui génère des statistiques sur la consultation d'un serveur web à partir du fichier access.log d'Apache.

Pour générer les histogrammes ou camemberts, Webalizer utilise la librairie graphique de PHP : GD.

Voici les différentes options de compilation de Webalizer ( après avoir installé la librairie GD ) :

- ‡enable-dns : pour remplacer les IP par les noms de machine
- ‡prefix=DIR : pour spécifier le répertoire d'installation
- ‡with-gdlib=DIR /libgd.a : pour indiquer à Webalizer où se trouve ce fichier de la librairie GD
- ‡with-gd=DIR : pour indiquer à Webalizer où se trouve la librairie GD
- ‡with-language=french : pour spécifier la langue
- ‡etc-dir=DIR : pour spécifier le répertoire des fichiers de configuration de Webalizer

Il faut ensuite aller compléter les différents champs dans le fichier de configuration pour personnaliser l'affichage des statistiques.

### 2 ) Sur les enchères

#### [A – au final...](#)

#### 1.Fonctionnalités proposées

- Affichage du nombre d'objets en vente
  - par catégorie, lien vers les catégories
  - par utilisateur
- Répartition des utilisateurs
  - Par région (bug...)
  - Par pays
- Par nombre d'achats et de ventes
- Moyenne des prix

#### 2.Techniques utilisées pour la mise en oeuvre

Il n'y a aucun accès direct à la base : toutes les fonctions sont fournies par les groupes qui se sont occupés des bases => notre groupe n'a fait aucun accès aux bases.

L'affichage a été développé sous forme de fonctions ; ces fonctions appellent alors des fichiers php qui contiennent le code d'affichage (qu'ils gèrent dynamiquement.)

Au niveau organisation dans le répertoire stats/...

- Un fichier fonction.inc, qui contient les prototypes des fonctions avec les appels aux fichiers.
- Un dossier php/ et un dossier dat/.
- Dans le dossier php/ se trouvent tous les fichiers contenant du code à afficher dynamiquement

- Dans le dossier dat/ se trouvent des fichiers « de données » qui contiennent du code statique à afficher. (fichier afficherPageStats, qui propose les liens vers les fonctions d'affichage)

### Moralité : ce qu'on a fait et qu'on aurait pu faire mieux

Dans l'ensemble ça marche, mais les requêtes en boucle c'est pas très optimisé... Il y a donc un délai pour l'ouverture.

Solution (non mise en œuvre pour raison technique dans un premier temps, puis à cause du temps restant)

Faire comme webalizer : mettre à jour régulièrement et appeler un fichier au lieu de créer les stats dynamiquement et faire la requête sur la base à chaque appel.

## B. problèmes et solutions

### a. Déterminer les fonctions à demander...

On avait demandé un certain nombre de fonctions à l'origine, elles étaient sur le site. Mais à cette époque la, les groupes concernés n'étaient pas prêts à assurer qu'ils pourraient développer les fonctions. Un peu d'eau a coulé sous les ponts, on s'est préoccupé d'autre chose, et au moment de vraiment exploiter les fonctions... les groupes avaient quasiment fini et il était moins évident d'aller les voir en demandant des fonctions qu'ils n'avaient pas prévues...

On a donc eu recours à des rustines :

Les utilisateurs proposaient déjà la fonction obtenirDétail(\$identifiant). Avec une fonction obtenirTousLesPseudo(\$tableau) on pouvait déjà faire un certain nombre de stats, nous avons ensuite demandé les fonctions obtenirTousLesPays et obtenirTousLesCodepostal et nous avons fait le traitement statistique par du code (assez demandeur de ressources)

Pour les objets, ils proposaient une fonction d'affichage de tous les détails mais pas de fonction permettant de récupérer les détails afin de les exploiter. Nous avons donc décidé de faire moins de statistiques (et compte tenu du temps restant), et demandé quelques fonctions pour avoir des statistiques de base : obtenirLeNombreDObjetsEnVente(\$catégorie) par exemple qui permet d'obtenir le nombre d'objets en vente pour une catégorie spécifiée. (la signature dans le site est différente). Nous n'avions pas besoin d'une fonction envoyant les identifiants des catégories, vu qu'ils étaient fixés et ne changeront pas.

Enfin, les statistiques proposent l'affichage des moyennes des prix, donc une fonction moyennePrixActuel(\$id\_objet) a été demandée au groupe de gestion des transactions.

### b. faire de l'affichage de graphiques

On ne s'est pas beaucoup penché sur les graphismes, ce qui rend les statistiques fonctionnelles... mais pas lisibles.

C'est l'entière faute de Sozic, qui pensait avoir le temps de le faire elle même mais a été trop juste en temps à la fin.

La solution « facile » envisagée : mettre une image de couleur unie d'un pixel sur un pixel dans le répertoire, et afficher cette image de la taille voulue.

La solution plus difficile, en partie mise en œuvre mais pas terminée car on voyait peu l'intérêt de le faire sur toutes les statistiques à cause du temps restant : utiliser GD.

### c. pour la répartition par département... éliminer les doubles ?

Les statistiques proposent la répartition des utilisateurs par département. Méthode utilisée pour les calculer :

- Récupérer tous les codes postaux
- Récupérer les codes postaux par utilisateur
- On va ensuite chercher combien de fois les 2 premiers chiffres que l'on peut trouver parmi tous les

- codes postaux se retrouvent chez les utilisateurs.
- On commence donc par faire une division par 1000, on prend la partie entière, il ne reste que le numéro du département.
- C'est alors qu'on compare les codes des utilisateurs à ceux que l'on a trouvés.

Le problème se pose si différents codes postaux sont proposés dans une même région : on a des doubles, et impossibilité de les éliminer facilement. Nous avons essayé une « bidouille » en stockant les numéros de régions déjà comptés, mais le temps commençant à manquer nous en sommes restés là. Ce problème est aussi à l'origine du non-graphisme « simples ».

J'aurai donc retenu une fois de plus que ça ne sert à rien de se focaliser sur des détails, et qu'il vaut mieux s'intéresser à ce qui se voit en premier...

P.S. pour les stats : à propos du bug pendant la présentation.

Le bug est pour l'instant inexplicable car le numéro de la ligne ayant un « parse error » est la dernière ligne d'un fichier. Toutes les modifications apportées avant le bug ne portaient pas sur cette partie du fichier.

## C/LE FORUM

- 1 – Choix du forum
- 2 – Installation du forum
  - 2.1 – Etapes
  - 2.2 – Problèmes rencontrés
- 3 – Réalisation des fonctionnalités annexes
  - 3.1 – Les fonctionnalités choisies
  - 3.2 – La méthode d'implémentation
- 4 – Conclusion

### 1 - Choix du forum

Après une rapide recherche sur le net, nous avons trouvé trois forums qui nous semblaient intéressants.

IkonBoard : semblait être intéressant, mais il faut s'inscrire pour pouvoir le télécharger.

YabbForum : à l'avantage de ne pas nécessiter de base de données, par contre il faut avoir installé CGI et Perl. En outre, c'est le forum utilisé pour le site Insa, nous voulions changer un peu.

PhpBB : peut être utilisé avec n'importe quel type de base (MySQL, PostgreSQL, Access, ...). En outre son installation semblait assez facile, et une visite sur internet nous a permis de voir que beaucoup de sites utilisaient ce forum. C'est pourquoi nous avons décidé d'installer un forum phpBB.

### 2 – Installation du forum

#### 2.1 – Etapes

La première chose à faire est de récupérer les sources. On peut les trouver sur le site général de phpbb (<http://www.phpbb.com>) ou sur le site français (<http://www.phpbb-fr.com>). Dans le cas de phpBB, il fallait d'abord récupérer le .tar.gz contenant les bases du forum, et accessoirement l'archive du langage français.

Une fois cette opération réalisée, tout ce qu'il faut faire c'est décompresser l'archive dans le répertoire source des pages web (/var/www/ dans notre cas). Il suffit ensuite d'ouvrir dans un navigateur la page index du répertoire obtenu, et on obtient une page d'installation.

Auparavant il faut bien entendu avoir créé une base de donnée vide pour pouvoir indiquer à phpbb quelle base de donnée il va utiliser.

Un point important lors de l'installation, et qui est bien précisé dans la documentation, est qu'il faut mettre en accès libre total le fichier config de phpbb. En effet le programme d'installation va écrire dans ce fichier, et si celui-ci n'est pas en accès complet, alors on doit effectuer une manipulation supplémentaire qui n'est pas utile.

Au final le programme d'installation créé toutes les relations dont il a besoin dans la base, et le forum est alors opérationnel.

#### 2.2 – Problèmes Rencontrés

Nous avons rencontré quelques difficultés lors de l'installation, bien que celle-ci semble simple. La principale difficulté a été le fait que le programme d'installation n'a pas voulu marcher la première fois sur la base de donnée PostgreSQL, alors qu'il avait très bien marché sur une base MySQL. L'erreur indiquée ne nous apportant aucune précision, nous avons cherché sur internet une solution à notre problème. Après avoir consulté le forum de phpbb.com, nous avons découvert qu'il existait un bug au niveau du programme d'installation. En effet, la version précédente à celle que nous avons utilisé (la 1.4) semblait ne marcher qu'avec une base MySQL. Et la version que nous avons utilisée, qui est la plus récente (2.01), bien qu'elle soit censée fonctionner avec tout type de base de données, semble orientée principalement pour MySQL. Pour que l'installation marche avec une base PostgreSQL, il faut changer un 0 en 1 dans les fichiers de définitions des relations de la base. Cette petite modification nous a permis d'installer le forum sur enchere.ath.cx.

### **3 – Réalisation des fonctionnalités annexes**

#### **3.1 – Choix des fonctionnalités**

Pour bien faire le lien entre le site d'enchère et le forum, nous avons décidé de rajouter des fonctionnalités au forum.

Tout d'abord, lorsqu'un utilisateur s'inscrit sur le site d'enchère, il est automatiquement inscrit sur le forum, avec le même login et le même mot de passe.

La deuxième fonctionnalité que nous avons voulu ajouter, c'est la création automatique d'un topic lors de la mise en vente d'un objet. Le sujet de ce topic est le nom de l'objet, son premier message, qui est déposé sous le nom du vendeur, contient la description de l'objet.

#### **3.2 – Implémentation des fonctionnalités**

La première méthode à laquelle nous avons pensé était de se servir des fonctions déjà existantes dans les fichiers de phphbb. Malheureusement, ces fichiers sont assez illisibles, et nous n'avons pas réussi à retrouver les fonctions que nous cherchions.

Nous avons donc dû nous ramener à la deuxième solution, qui consiste à intervenir directement sur la base de données de phpBB. Toutefois, avant de pouvoir réaliser une quelconque fonction, il nous a d'abord fallu comprendre comment était organisé les données. Cela n'a pas posé trop de problèmes, que ce soit pour l'insertion d'un topic ou l'insertion d'un utilisateur. Le seul problème que nous avons rencontré concernait le mot de passe. En effet, phpBB ne conserve pas dans la base de données le mot de passe en clair, mais le crypte avant. Nous avons donc dû retourner voir dans le code de phphBB pour trouver cette fonction de cryptage.

Les fonctions en elle-même sont de simples fonctions php, faisant appel à des fonctions d'écriture et de lecture dans la base de données.

### **4 – Conclusion**

En conclusion, nous pouvons dire que l'installation des forums n'a pas été très difficile. Par contre, l'ajout de nouvelles fonctions a été rendu difficile par le fait que le code de phpBB était difficilement lisible.