

Documentation Impresora

- Pour un futur stagiaire -

Bonjour à toi jeune padawan ! Je me présente, Julie Meunier, actuellement étudiante en BTS SIO, option SLAM, au lycée Carcouët. Au moment où j'écris ces lignes, je termine ma cinquième et dernière semaine de stage au sein de la CIR du lycée Livet.

Si tu es là, c'est probablement pour continuer mon projet, et l'améliorer, c'est pour cela que je vais te faire une présentation de ce que j'ai pu faire à ce jour !

Sommaire

1. Présentation du projet.....	3
1.1. Outils utilisés.....	3
2. Impresora – Outils.....	4
2.1. La base de données.....	4
2.2. Schéma explicatif du site web Impresora.....	6
2.3. Script Python.....	7
2.4. Script PHP.....	7
3. Impresora – Site Web.....	9
3.1. Arborescence du site.....	9
3.2. Menu.....	10
3.2.1. La liste déroulante.....	10
3.2.2. Les formulaires.....	11
3.2.3. Les graphiques.....	11
3.2.3.1. Lien avec le menu.....	11
3.2.3.2. Les requêtes.....	11
3.2.3.3. Calcul.....	12
3.2.3.4. Images.....	12
3.3. Les formulaires.....	12
3.3.1. Formulaire d'ajout d'imprimantes.....	12
3.3.1.1. Formulaire.....	13
3.3.1.2. Traitement.....	13
3.3.2 Formulaire de suppressions.....	14
3.3.2.1. Formulaire.....	14
3.3.2.2. Traitement.....	14
Conclusion.....	15

1. Présentation du projet

J'ignore si Frédéric Volpi t'a un peu expliqué, mais voici ce que j'ai du faire.

Un ancien stagiaire a créé un script Python permettant de récupérer, grâce à un protocole SNMP, les compteurs des imprimantes de tout le lycée, pour les insérer dans des fichiers .CSV (un fichier correspondant à une imprimante), et les mettre à jour, tous les jours à midi.

A partir de ces fichiers, mon projet était de créer un site web affichant un graphique pour chaque imprimante, pour montrer la consommation des imprimantes par jour (papier et encre). En gros, afficher la différence de compteurs entre chaque jour.

J'espère que tu as compris ! Si c'est le cas, passons à la suite ! (Sinon demande plus de renseignements autour de toi ;))

1.1. Outils utilisés

Pour mener à bien ce projet, j'ai travaillé sur un environnement Ubuntu, sur une machine virtuelle Debian. Sur cette dernière, j'ai installé un LAMP (apache2, mysql, php5), et **TRES IMPORTANT**, la librairie JpGraph, qui permet de créer les graphiques !

2. Impresora – Outils

2.1. La base de données

Pour commencer, je vais te présenter la base de données qui se nomme « imprimantes_livet ».

Elle se compose de six tables que je vais te détailler ci-dessous, sous formes de relations.

IMPRIMANTES	(id_imprimante, nom_imprimante, adresse_ip, iso, batiment, salle, type_imprimante, marque_imprimante, modele_imprimante, format_imprimante)	
	id_imprimante	clé primaire
	type_imprimante	clé étrangère en référence à id_type de la table TYPE_IMPRIMANTE
	marque_imprimante	clé étrangère en référence à id_marque de la table MARQUE_IMPRIMANTE
	modele_imprimante	clé étrangère en référence à id_modele de la table MODELE_IMPRIMANTE
	format_imprimante	clé étrangère en référence à id_format de la table FORMAT_IMPRIMANTE

MARQUE_IMPRIMANTE	(id_marque, marque)	
	id_marque	clé primaire

MODELE_IMPRIMANTE	(id_modele, modele)	
	id_modele	clé primaire

TYPE_IMPRIMANTE	(id_type, type)	
------------------------	-----------------	--

id_type clé primaire

FORMAT_IMPRIMANTE (id_format, format)

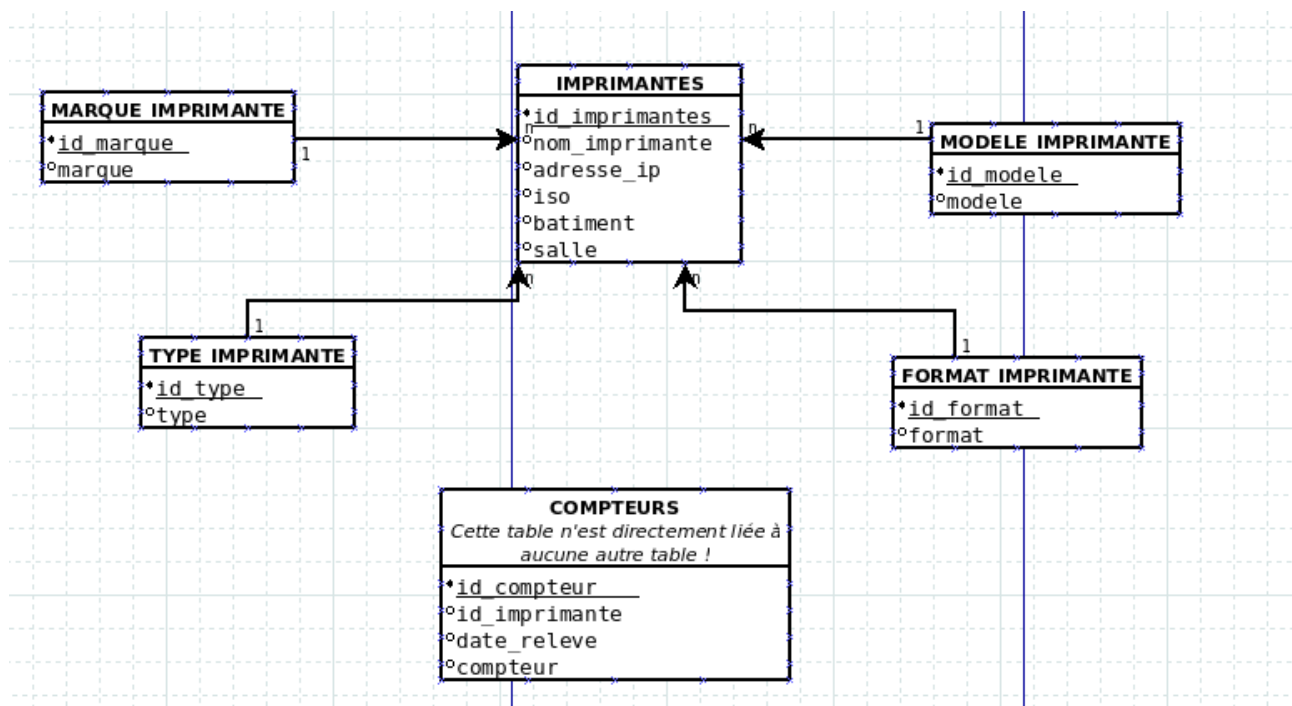
id_format clé primaire

COMPTEURS (id_compteur, id_imprimante, date_releve, compteur)

id_compteur clé primaire

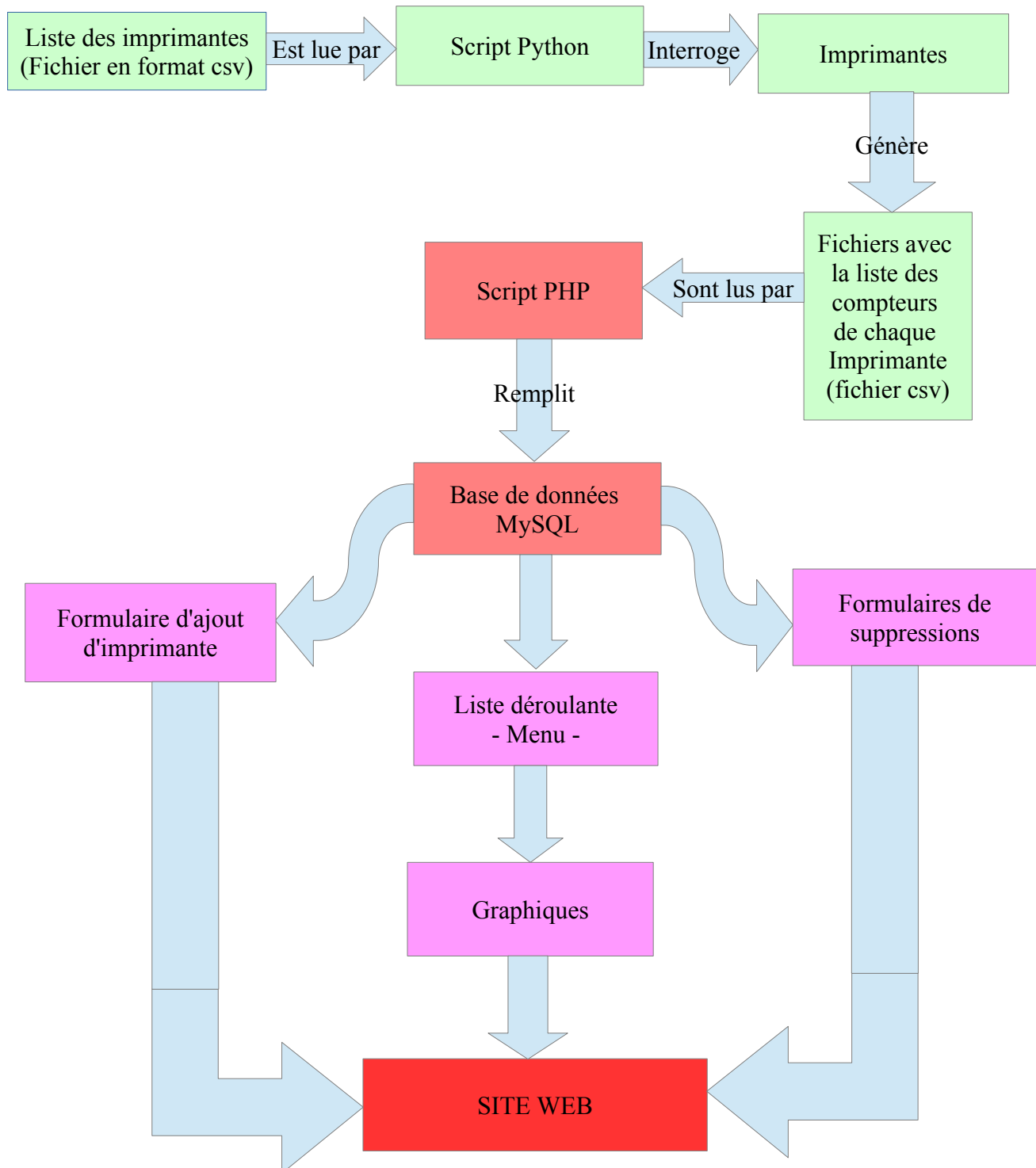
(/!\ Il n'y a pas de clé étrangère dans cette table ! /!\)

Schéma de la base :



2.2. Schéma explicatif du site web Impresora

Voici un « petit » schéma du déroulement des opérations d'*Impresora* :



J'espère que c'est plus ou moins clair pour toi à ce niveau là :)

Je vais maintenant te faire un explicatif plus détaillé !

2.3. Script Python

Bon, je vais passer rapidement sur ce point, car ce n'est pas moi qui ai écrit ce script, et je dois avouer que le Python et moi, ça fait deux !

Néanmoins, tout part de là, je suis donc bien obligée d'en parler ;)

Le premier fichier, que tu trouveras sur le serveur « srvimp », dans le dossier « script_python », se nomme « imprimantes_livet.csv ». C'est le fichier qui recense toutes les imprimantes du lycée sous forme de chaîne de caractère. Si tu l'ouvres, tu pourras voir qu'une chaîne de caractère se compose de l'imprimante (marque/modèle), de son ip, de son iso et de son nom (qui est composé, en général, du modèle et de l'endroit où elle se trouve dans le lycée). A ce jour, il y a 55 imprimantes répertoriées (oui ça fait beaucoup !) !

Ce fichier est lu par le script python, nommé « compt_print » (qui se trouve dans le même répertoire que la liste des imprimantes). Celui-ci va récupérer dans une boucle toutes les chaînes de caractères et créer un fichier pour chaque imprimante, nommé par l'adresse IP de chacune. Tous les jours, le protocole SNMP rajoutera à la suite de ces fichiers le compteur imprimante du jour (sachant qu'une imprimante éteinte ne donnera rien).

Les fichiers .CSV, qui nous seront utiles pour la suite, sont donc créés. Ils se trouvent dans le dossier « /script_python/historique/ ».

2.4. Script PHP

Le script PHP, que j'ai créé moi-même, cette fois-ci :p se nomme « entree_compteurs.php », qui se trouve dans le dossier « /var/www/imprimante » du serveur « srvimp » ou alors dans mon propre dossier stagiaire.

Ce script est important, car c'est lui qui va remplir la table « compteurs » de la base de données.

N'oublie pas d'ouvrir le fichier avant de lire la suite ! Ca te permettra de mieux comprendre ce que j'ai fais :p

1. Dans un premier temps, on se connecte à la base de données. Rien de bien compliqué en soi.
2. Ensuite, on vide entièrement la table « compteurs ».

Ce point sera sûrement à optimiser plus tard. Je dois avouer que je n'ai pas réussi à faire en sorte qu'une nouvelle entrée se mette dans la base à chaque fois, pour chaque imprimante. Du coup, on supprime tout, pour la re-remplir entièrement !

3. Ensuite, on intègre dans la variable \$repertoire le dossier que nous allons travailler par la suite. Ici, ce sera le dossier « /script_python/historique/ ».
4. Les fichiers se trouvant dans ce dossier vont être récupérés dans la variable \$MesFichiers, grâce à la fonction PHP « scandir » qui va lister les fichiers.

C'est maintenant que ça va devenir un peu plus compliqué. Rappelle-toi la composition de la table « compteurs ». Il y a quatre champs : id_compteur (qui se remplira en auto_increment, donc pas besoin d'y toucher), id_imprimante, date_releve et compteur.

Dans nos fichiers .CSV, nous avons des chaînes de caractère ressemblant à celle-ci : 31/01/14;29202. Nous en avons des dizaines et des dizaines par fichiers, soit les 55 fichiers. Je te laisse faire le calcul, c'est beaucoup !

1. On va d'abord ouvrir une première boucle (foreach) qui lancera le code suivant tant qu'il y aura des fichiers. Puis on ouvre une seconde boucle (if) qui s'occupera des fichiers, un par un.

Pour remplir les champs date_releve et compteur, nous avons nos données dans les chaînes. Toutefois, pour remplir le champ id_imprimante, cela va être différent. Souviens-toi je t'ai dit qu'il n'y avait pas de clé étrangère dans cette table. En réalité c'est le script PHP qui va créer le lien entre la table « compteurs » et la table « imprimantes ».

2. En effet, le code de la ligne 21 à la ligne 26 va récupérer le nom des fichiers, en enlevant l'extension .CSV, donc il va récupérer les IP de chaque imprimante, sous forme de nom. Ensuite, grâce à la requête SQL, on va faire le lien entre les adresses IP récupérées dans les noms de fichier et les adresses IP de la table imprimantes. Pour finir, on va décider d'afficher dans la variable \$result, l'identifiant qui correspond à chaque IP !
3. Maintenant qu'on a la variable qui va récupérer l'identifiant de l'imprimante, il va falloir récupérer les autres champs. On ouvre donc une troisième boucle (if) qui va travailler sur chaque fichier. Puis une dernière boucle (while) qui va lire chaque ligne d'un fichier.
4. Grâce à la fonction « fgetcsv », on va pouvoir découper chaque chaîne de caractère, à partir du délimiteur « ; ». Ensuite, il va falloir découper la date pour la remettre dans le bon format correspondant au format de la base de donnée (on a un format jj/mm/aa et on veut un format aaaa-mm-jj). On va donc redécouper la chaîne de caractère dans la variable \$explode qui correspond à la date, à partir du délimiteur « / », puis dans la variable \$date, on remet tout dans le bon sens.
5. Enfin, on va insérer nos données dans la base de données grâce à une requête SQL, avant de fermer le fichier.
6. Pour finir, on affiche la date du jour dans la variable \$date_jour ainsi que l'heure dans la variable \$heure avant de les afficher. Cela apparaîtra dans les fichiers log.

Voilà donc pour le Script PHP :)

Lycée Livet - Nantes

Si mon explication n'est pas très claire, je te laisse étudier plus en détail le code en question, et les fonctions qui ont été utilisées.

Une dernière chose, un Cron exécute ce script tous les jours à 13h, ce qui met automatiquement la base de données à jour, et donc les graphiques.

3. Impresora – Site Web

3.1. Arborescence du site

Nous allons maintenant passer à ce qui nous intéresse : le site.

Le site en lui-même se compose de trois pages :

- index.php
- page_formulaire_ajout.php
- page_forumaire_suppression.php

Néanmoins, j'ai créé dix fichiers PHP pour que le code soit moins lourds. Voici les sept autres fichiers :

- header.php
- menu.php
- suppression_imprimante.php
- suppression_marque.php
- suppression_modele.php
- suppression_type.php
- suppression_format.php

Comme tu peux le voir, la plupart des fichiers sont ceux des formulaires de suppression. Voici une arborescence qui t'expliquera comment les fichiers sont organisés entre eux, grâce à la fonction PHP « include » :

- index.php
 - header.php
 - menu.php
- page_formulaire_ajout.php

- header.php
- menu.php
- page_formulaire_suppression.php
 - header.php
 - menu.php
 - suppression_imprimante.php
 - suppression_marque.php
 - suppression_modele.php
 - suppression_type.php
 - suppression_format.php

De plus, les formulaires sont liés au menu par une association de liens.

3.2. Menu

Je vais maintenant t'expliquer le déroulement du menu. Il comporte le titre du site, la liste déroulante et les onglets permettant d'accéder aux formulaires.

3.2.1. La liste déroulante

Cette liste est ce qu'il y a de plus important dans le site web. En effet, c'est elle qui va afficher les graphiques, ce qui est quand même le but de ce projet ;)

La liste en elle-même est créée à partir de la base de données. J'ai créé un formulaire basique et y ai inséré une liste déroulante avec la balise HTML « select ». Toutefois, j'ai du intégré du PHP dans ce formulaire pour appeler la base de données.

Voici les étapes :

- Connexion à la base.
- Requête SQL pour faire appel à la table « imprimantes », dont les données sont triées par ordre alphabétique.

- Ouverture de la liste déroulante avec le « select ». Bien faire attention au « name » qui est donné. C'est grâce à lui que nous allons faire le lien avec les graphiques.
- Boucle permettant d'intégrer dans la liste les noms d'imprimante. Toutefois, elle prend en paramètre les identifiants des imprimantes, et affiche les noms.
- Fermeture de la boucle.
- Déconnexion de la base.
- Un bouton permettant l'envoi du formulaires

3.2.2. Les formulaires

Rien de bien exceptionnel ici.

J'ai créé un menu déroulant avec l'Accueil et les Formulaires. Ces derniers nous dirigent soit vers le formulaire d'ajout soit vers ceux de suppression.

3.2.3. Les graphiques

3.2.3.1. Lien avec le menu

Je les mets dans cette partie, car c'est grâce au menu qu'ils vont s'afficher.

Je te laisse ouvrir le fichier « graph_histo_impr.php » :)

Les graphiques sont créés grâce à la librairie PHP JpGraph. Si tu regardes dans le dossier, tu pourras y trouver de nombreux exemples de graphiques. Mais au final, nous n'avons pas spécialement besoin de ce dossier ici. Tout ce que nous avons à faire au début du fichier, c'est d'inclure les scripts tout faits que nous offre cette superbe librairie, grâce à la fonction « include ». Tu n'as, normalement, pas besoin d'y toucher ;)

Ensuite, nous avons la fameuse connexion à la base de données pour y récupérer ce qui nous intéresse. C'est à dire les dates et les compteurs d'imprimantes.

J'ai d'ailleurs créer deux tableaux pour recenser ces données : tableauDate et tableauCompteurs.

La ligne 23 est celle qui va faire le lien avec la liste déroulante du menu. **Comme tu peux le voir, on créé une variable (ici \$id_impr) qui va aller récupérer l'identifiant de l'imprimante choisi dans le menu** (rappelle-toi, les paramètres sont les identifiants et non pas les noms!). **Grâce au \$_GET, l'identifiant est récupéré grâce au « name » choisi dans la liste** (ici « imprimante »), ainsi nous avons fait le lien entre le menu et le graphique !

3.2.3.2. Les requêtes

Je ne m'attarderai pas sur ce point, car normalement les commentaires du script permettent de comprendre !

Il y a quatre requêtes de faites. La première permet de récupérer les compteurs et les dates de chaque imprimante en fonction de l'identifiant de l'imprimante (donc en fonction de l'imprimante choisie dans la liste). C'est la plus importante !

Ensuite, nous avons trois requêtes qui permettent de récupérer respectivement le nom de l'imprimante, son IP et le dernier compteur en date. Ces trois requêtes nous permettront donc de créer le titre de chaque graphique d'imprimante.

Après les requêtes, nous avons les récupérations de données grâce aux fonctions « `mysql_query` » et « `mysql_fetch_assoc` ». A faire pour chaque requête.

3.2.3.3. Calcul

A ce niveau là, tout ce que le graphique peut afficher, ce sont les compteurs à proprement parler. Mais ce n'est pas ce qui nous intéresse ici ! Non, nous, ce que l'on souhaite, c'est d'afficher la différence entre chaque jour.

On fait d'abord une première boucle qui nous permet de remplir les tableaux `tableauDate` et `tableauCompteurs` avec les données qui nous intéressent.

Puis, on va créer deux variables, `$i` et `$j`. La première sera l'indice (mis à 0), la seconde sera un tableau. Ces deux variables vont nous permettre de faire l'opération qui affichera la différence entre deux compteurs. Je te laisse étudier ça tout(e) seul(e) car ce n'est pas vraiment explicable ici ! ;)

3.2.3.4. Images

Après tout ça se trouve le code du graphique. Je ne te le détaillerai pas ici, c'est inutile puisque les commentaires sont là pour ça. De plus, c'est un code emprunté sur le tutoriel d'Eric Pommereau, donc même si je le voulais, ce serait difficile pour moi de te l'expliquer.

Seules les lignes 120, 139 et 150 nous intéressent vraiment.

- La première est celle qui permet d'intégrer les différences de compteurs (`$j`) sur l'axe des ordonnées, et de créer l'histogramme.
- La seconde est le titre avec tout ce que nous voulons y mettre (nom, ip et dernier compteur).
- La troisième ligne est l'affichage du `$tableauDate` sur l'axe des abscisses.

3.3. Les formulaires

3.3.1. Formulaire d'ajout d'imprimantes

Ce formulaire, comme son nom l'indique, permet d'ajouter une imprimante dans la base de données.

Toutefois, il y a un problème dont je me suis rendue compte un peu trop tard, c'est que **même si l'imprimante est ajoutée dans la base, elle n'est pas ajoutée dans le fichier « imprimantes_livet.csv » qui est notre fichier de base !** Il faut donc le faire à la main, ce qui est problématique. C'est donc un point qu'il faudrait améliorer par la suite.

Bref.

Le formulaire se compose de deux parties : le formulaire et le traitement.

Avant de passer au formulaire en lui-même, il ne faut pas oublier de se connecter à la base de données. Ici, j'ai utilisé la PDO au lieu d'une connexion basique.

3.3.1.1. Formulaire

Ce formulaire permet de faire plusieurs choses. Il permet d'ajouter une imprimante dans la table « imprimantes » (logique!), mais il permet également d'ajouter une marque, un modèle, un type et un format, dans les tables correspondantes.

Il y a trois parties : l'identité, les caractéristiques et la localisation.

L'identité se compose de trois « input ». Un nom, une adresse IP et un iso sont obligatoires.

Toujours bien faire attention aux « name » donnés, qui nous serviront dans le traitement.

La localisation se compose d'une liste déroulante écrite en brut qui montre les différents bâtiments du lycée, et d'un « input » qui permet d'entrer la salle.

Et enfin, les caractéristiques. C'est une partie très importante, car c'est elle qui va permettre d'ajouter des données dans les tables « marque_imprimante », « modele_imprimante », « type_imprimante » et « format_imprimante ». Mais nous verrons ceci plus en détail dans le traitement du formulaire.

Deux choix vont s'offrir à nous : soit choisir des données déjà existantes dans la base de données grâce aux listes déroulantes, soit entrer de nouvelles données.

Le premier choix n'est pas compliqué, c'est juste une liste déroulante qui récupère les données dans la base de données, de la même manière que les imprimantes dans le menu.

Pour le deuxième choix, des « input » sont créés pour pouvoir accueillir de nouvelles données.

Pour ces deux choix, il faut bien **faire attention aux « name »** qui vont nous permettre de traiter ces

deux données séparément dans le traitement. Pour bien les séparer, j'ai ajouté « List » en suffixe pour le « name » des listes, et « Free » pour celui des « input ».

3.3.1.2. Traitement

C'est dans le traitement du formulaire que les données seront intégrées dans la base de données.

Tout d'abord, on va récupérer les valeurs de l'identité et de la localisation grâce à \$_POST et aux « name » que je leur ai donné dans le formulaire. Puis on les met dans des variables,

Ensuite, on passe aux traitements des différents champs. Je vais prendre pour exemple le champ « marque », mais c'est la même chose pour les quatre.

Donc pour commencer, je crée une variable \$marque vide.

Puis j'ouvre une boucle if. Si le champ « marqueList », qui correspond à la liste déroulante des marques, n'est pas vide, la variable \$marque prend la valeur de la liste.

Par contre, si c'est le champ « marqueFree » qui n'est pas vide, la variable \$marque prend cette nouvelle valeur en compte. Si c'est le cas, on va insérer dans la table « marque_imprimante », dans le champ « marque », cette nouvelle donnée, en ajoutant un nouvel ID.

Si les deux sont vides (la liste et l'input), on affiche un message d'erreur.

Si la variable du message d'erreur, \$mess, ne contient rien, on va pouvoir insérer les données dans la table « imprimantes » grâce à une nouvelle requête SQL.

Une fois cela fait, le traitement est terminé. ;)

3.3.2 Formulaire de suppressions

Dans cette partie, je vais parler des cinq formulaires de suppression ensemble, car ils sont faits de la même façon.

3.3.2.1. Formulaire

Le formulaire se compose d'une simple liste déroulante qui récupère les données dans la table qui correspond, de la même façon que la liste déroulante du menu.

Rien de bien exceptionnel ici non plus :p

3.3.2.2. Traitement

La différence va se faire dans cette partie : le traitement.

Donc, de la même façon que précédemment, on va récupérer les données du formulaire grâce à

Lycée Livet - Nantes

\$_POST, et au « name » choisi dans le formulaire.

On se connecte à la base de données.

Puis on écrit une requête SQL qui va supprimer dans la base de données, l'enregistrement correspondant à l'identifiant de l'imprimante (ou de la marque/modèle/type/format).

Conclusion

Voilà, je pense que j'ai fais le tour et que tu as tous les outils en main pour refaire le projet s'il devait y avoir un problème ! (ce que je n'espère pas !)

Comme tu peux le constater il y a beaucoup d'amélioration à faire pour optimiser le site.

Je peux te donner quelques idées qui étaient prévues ou que j'ai eu :

- Ajouter un calendrier pour choisir une date particulière à laquelle afficher le graphique. La date choisie et les trente jours qui suivent cette date.
- Créer une boîte de dialogue pour la confirmation de suppression.
- Faire un lien entre l'ajout d'imprimante dans la base de données, et l'ajout d'imprimante dans le fichier de base « imprimantes_livet.csv », à l'aide d'un script.
- Optimiser le script d'entrée de compteurs dans la base pour ne pas avoir à la vider à chaque fois.

Pour conclure c'est un projet qui est loin d'être parfait, et qu'il va falloir améliorer ! Je te fais confiance pour ça ;)

Si tu as des questions, je reste disponible par mail, à cette adresse : julie_meunier44@yahoo.fr

Bon courage pour ton stage !

Julie