Modification affichage erreur PHP.

Mon tuteur de stage souhaite ne pas afficher les erreurs PHP pour sur le navigateur afin que le client ne soit pas exposé à des informations qui ne le concerne pas, et qui gâche les pages du site. Il m’a alors confié la tâche de désactiver l’affichage des erreurs PHP. J’ai alors étudié l’affichage des erreurs, et j’ai effectué une recherche sur tous les fichiers du message qui s’affiche à l’écran. Cette recherche m’a conduit à un fichier ne contenant que du html, n’était qu’en fait que la mise en forme des informations et qu’il n’était appelé par d’autre fichier. J’ai donc réalisé une autre recherche sur tous les fichiers, mais cette fois-ci avec le nom du fichier contenant le html, afin de savoir dans quel autre fichier il était appelé. Ce qui m’a conduit à un fichier « Exception.php » avec plusieurs fonctions PHP. Ces fonction était l’affichage d’erreur quel que soit, et une seconde l’affichage des erreurs PHP spécifiquement. J’ai alors réutilisé une des constantes (« DISPLAY\_ERROR\_ENABLED ») du fichier « constants.php », afin d’activer ou non l’affichage des erreurs. C’est la même constante que l’affichage des erreurs SQL. J’ai donc mit tout le contenu de la fonction dans une structure conditionnel, qui vérifie que la valeur de la constante soit bien à 1.

Mon tuteur de stage avait un problème sur un script PHP. Le but de ce script étant de stockées des informations de la base dans 4 fichiers textes différents, et de les envoyer en ftp sur un serveur. Mon tuteur de stage de a remarqué que certains même fichier était divisé en 2 fichiers distincts. De ce fait, ce n’était que le dernier fichier créé qui était envoyé en ftp sur le serveur. Mon tuteur soupçonné la configuration du serveur ou du « crontab ». Mon regarde extérieur m’a permis d’identifier rapidement l’erreur. Le nom des fichiers était une concaténation d’une chaine et de la date et heure avec les secondes. Il suffit que le serveur connaisse un période d’activité renforcé pour mettre du temps quant à l’exécution des requêtes. De plus le nom du fichier étant créée dans une boucle qui permettait à chaque passage d’écrire une nouvelle ligne dans le fichier, le temps de boucler et le nom du fichier diffère d’un une seconde. Il suffit donc de créer le nom du fichier hors de la boucle, avant celle-ci, a un moment fixe.

Toujours dans le même script, une erreur est journalisé dans les fichiers logs concernant la fonction date(). En effet les paramètres passés dans cette fonction ne sont pas entre guillemet, bien que la fonction retourne bien les valeurs attendues, cela génère une erreur. J’ai donc ajouté les guillemets manquants, aux paramètres à chaque fois que la fonction date est appelé.

Encore dans le même fichier, les fichiers créés avait un chemin définit directement dans le script. Mon tuteur de stage souhaitait tout d’abord enregistrer ces fichiers dans un endroit plus rationnel, car ils étaient dans le répertoire « www », donc accessible depuis internet. Il souhaite donc le mettre dans un répertoire stockant déjà d’autre exportation et ne se trouvant pas dans le répertoire « www ». De plus mon tuteur souhaitait utiliser une des constante du fichier « constants.php ». J’ai donc utilisé la constante « PATHCSV », qui était déjà présente mais inutilisé. J’y ai renseigné un chemin absolu correct.

Dans un fichier de log que je mis en place, une même erreur PHP survient à chaque exécution de requête SQL. L’erreur survient au moment de l’appel de la fonction set\_magic\_quotes\_runtime(). En effet cette fonction n’est plus utilisée depuis la version 5.3 de PHP, alors que sur le serveur c’est la version 5.3.14 qui est utilisée. De plus l’appel à cette fonction est censé désactiver l'option magic\_quotes\_runtime, alors qu’avec cette version de PHP, il est en natif désactivé.

Dans le fichier « cron.php », une fonction d’envoi de mail présente n’avait pas fini d’être développée. Mon tuteur de stage souhaitait donc que cette fonction soit fonctionnelle. Le mail qui est envoyé est soit pour stipulé au destinataire que l’exportation s’est bien passé, soit pour stipulé un problème lors de l’exportation. Le mail doit également doit indiquer le numéro des commandes concernées. Et enfin le mail doit être envoyé à un destinataire en copie-conforme, il faut donc créer une nouvelle constante dans le fichier « constants.php ». Je récupère le travail que j’ai déjà fait dans le fichier « cronExportVente.php ». Le retravaille pour m’adapter à la fonction, dans laquelle on passe 2 paramètres, qui sont l’objet et le corps du mail. Pour écrire les numéros de commande dans le mail, il me suffit d’utiliser l’instance « $this->data[‘commandes’] », dans laquelle est déjà stocker tous les commandes concernées par l’exportation. En effet, un peu avant l’appel de la fonction de l’envoi de mail, une fonction retourne toutes ces commandes et sont stockées dans « $this->data[‘commandes’] ». Comme c’est un tableau, l’utilisation d’un foreach permet de tourner autant qu’il y a de commandes. Et comme c’est un tableau je prends l’initiative d’écrire dans le mail le nombre de commandes concernées. Ceci est apprécié par mon tuteur de stage.