

Présentation du projet

Création à l'aide d'osCommerce d'une boutique en ligne de fournitures pour les beaux arts, avec une interface d'administration du site, sécurisée par mot de passe, avec gestion de droits différents pour l'administrateur et le responsable de la boutique qui lui n'interviendra que pour la saisie et la modification des articles et les catégories, ainsi que la personnalisation d'un texte de la page d'accueil pour indiquer les news générales du site.

→ Utilisation de la contribution fckeditor pour l'éditeur WYSIWYG

→ Utilisation de la contribution "Admin Account with Access Level"

Le site proposera les actualités des arts plastiques dans la région lyonnaise via un flux RSS fourni par le site acryom. Lien du flux: <http://www.acryom.com/rss-departement-69.xml>

→ Création d'un nouveau bloc 'flux_rss'

Fonctionnement de la boutique:

- La boutique sera en langue française, le paiement ne sera possible qu'en euros.
- Le catalogue est géré par des liens vers les différentes catégories et sous catégories
- Recherche possible par mots clés et par marque (liste déroulante), ainsi que recherche avancée - Certaines catégories auront une description et une photo

→ utilisation de la contribution categorie_description

Fonctionnement du compte client:

- L'internaute devra être inscrit pour pouvoir commander. Toutes les pages contenant des données personnelles seront sécurisées : inscription, gestion du compte, et paiement.

→ Utilisation du ssl

- Moyens de paiement: CB, chèque. Utilisation de la contribution ATOS pour assurer la compatibilité entre oscommerce et le kit livré par l'établissement bancaire (fonctionne avec la plupart des banques) – à voir s'il est possible de l'utiliser dans ce cas où la boutique n'est pas destinée à la production. Il est possible sinon d'utiliser le module de base pour le paiement par carte de crédit (validation théorique du n° de carte saisi)

- Modules de livraison: module "Frais au total", en fonction du prix de la commande: 8€ si commande jusqu'à 25€, 5€ si commande entre 25 et 50€, gratuite au dessus.

- Possibilité de recevoir des notifications de produits par mail, d'envoyer l'info d'un produit à un ami de recevoir la newsletter.

Réalisation du projet

Le site a été développé sur un PC windows en local sous apache2 et php5 (php5 pour permettre l'utilisation de l'API DOM pour l'affichage des news), avec openSSL installé sur le poste, ainsi que l'envoi de mail activé.

J'ai essayé dans la mesure du possible de coder dans le respect de l'architecture d'OsCommerce afin de permettre l'évolution vers un site multilingue.

- Modification de l'aspect graphique:

- *Ajout du box de recherche à côté du bandeau* : Fusion des boxs search et manufacturer.

Ajout d'un bouton ok pour la recherche par marque.

- *Création du box flux_rss:*

- Création également d'une fonction javascript popup() : include/bulle.js.php pour permettre un affichage dynamique du contenu des news au survol d'un élément. Cette fonction est également utilisée pour l'affichage de certains textes d'aide géré par osCommerce par une popup « classique ».

- La récupération et l'affichage des news est effectué avec l'API DOM de php5 ainsi que Xpath pour l'affichage des 6 dernières news. Cela aurait été mieux de le faire avec le parsing sax, mais

dans ce cas, on ne peut gérer ni le nombre de résultats souhaités, ni l'affichage dans les bulles. L'arbre doit donc être en mémoire.

- Cette fonctionnalité est assez lourde du fait de la récupération du flux en ligne. J'ai donc créé une fonction permettant le cache de ce box, qui sera réactualisé toutes les 24 heures.

- Création du box compte :

A- Lorsque l'internaute n'est pas identifié:

- Création d'une bulle pour l'affichage des infos sur le panier membre. Utilisation des classes créées pour le flux rss avec quelques modifications au niveau des css sur la position de la bulle.

1- Formulaire de la page de login.php.

L'attribut action du formulaire doit être la page courante, cela permet la connexion sans passer par une page spéciale. L'inconvénient étant que l'envoi des données du formulaire lors de l'identification n'est pas sécurisée.

- Création d'un script qui permet la connexion lors du clic sur le bouton ok du formulaire de connexion.

2- Ajout des liens "mot de passe oublié", et "Créer un compte".

B- Lorsque l'internaute est identifié:

1- Ajout du texte "Vous êtes connecté en tant que: prénom nom".

2- Ajout du lien "Accéder à mon compte",

3- Ajout du lien "Me déconnecter". Le lien est également celui de la page courante avec une variable action=logoff ajoutée à l'url. Il a donc fallu créer un script basé sur le script de la page logoff.php. Le test sur cette variable au chargement de la page permet l'inclusion ou non du script, et donc la déconnexion de l'internaute.

- Ajout d'un bloc promotions

Création à partir du module nouveautés et de la page specials.php. Affichage uniquement s'il y a des promotions.

- Intégration de la contribution Admin access with level:

- Ajout de 3 tables dans la base de données : admin, admin_files, admin_groups

- Ajout et modification de plusieurs scripts dans la partie admin

L'interface de gestion est maintenant sécurisée dans le répertoire admin_sand.

- Ajout d'un groupe gestion_boutique, pour permettre aux responsables de gérer la boutique. Ce groupe aura accès : à la partie catalogue (gestion des produits et des catégories), à la partie client (détail des clients, et commandes), à la partie statistique (reports), à la page permettant d'écrire un mail et à celle permettant l'édition de la newsletter.

- Intégration des catégories avec description:

Modifications supplémentaires:

- Dont l'ajout d'une fonction test_cat() qui permet de tester si la catégorie courante a une description ou non.

1- Dans la partie gestion:

- Ajout d'un champ 'manufacturer_id' dans la table, et d'un champ de formulaire pour la gestion de catégorie: Les produits d'une catégorie avec description auront tous cette marque.

- Modification des fonctions tep_get_products_name et tep_get_products_description de la partie admin avec ajout d'un paramètre : la catégorie courante, ce qui permet de tester avec la fonction test_cat s'il faut le nom / description de la catégorie (nom en cas de nouveau produit, description dans tous les cas). Ces fonctions sont utilisées lors de la création d'un nouveau produit.

2- Dans la partie boutique:

- Ajout d'une fonction test_cat() qui teste si la catégorie est une catégorie avec description

permettant :

- l'affichage du descriptif de la catégorie au lieu du descriptif du produit (qui lui est vide) :
- l'affichage de l'image de la catégorie au lieu de l'image du produit si celle-ci n'existe pas (cas des supports et de certains pinceaux) :
- Ajout d'une fonction `tep_get_attributs` permettant de récupérer les attributs facultatifs d'un produit.
- Fonctionnalité non développée : recherche dans la description des catégories en cas de recherche dans la description de produit.

- Intégration de l'éditeur WYSIWYG FCKeditor:

- Intégration partielle du fait que je ne souhaitais pas gérer les images (des produits et des catégories) avec l'éditeur. Ajout du dossier FCKeditor dans le répertoire d'administration. Ce dossier contient les éléments et classes de l'éditeur. Ajout de fonctions dans le script `admin_sand/includes/function/html_output.html`, permettant l'écriture des éléments html.
 - Remplacement des appels à la fonction `tep_draw_textarea_field()` de la page `admin_sand/categories.php` par appels à la fonction `tep_draw_fckeditor()`. Il s'agit donc des champs de description des catégories (non prévu par la contribution) et des produits. Il a fallu modifier également la fonction `Create()` de la classe FCKeditor qui écrivait directement dans la page alors que je souhaitais un return pour pouvoir écrire le code plus tard dans la page (pour les description de categories qui passent par un `infoBox()`)
- Les script `admin_sand/mail.php` ainsi que `admin_sand/newsletters.php` ont également été modifié.

- Ajout du texte d'accueil modifiable par le responsable de la boutique

accessible par le box « outils » de la partie gestion.

Cette page comporte la date de dernière modification du fichier, un textarea créé avec la fonction `tep_draw_fckeditor()`. Le contenu du textarea est le fichier texte contenant les news . Lors de la modification, mise à jour du fichier après avoir validé l'aperçu. Un message confirme la mise à jour du fichier.

- Intégration des attributs dans l'application

La principale difficulté a été de concevoir la gestion des attributs et surtout le cas de la contenance. J'ai longtemps hésité à utiliser la contribution Qtprom qui permet la gestion du stock et des attributs, mais elle ne correspondait pas vraiment à ce que je voulais faire, donc j'ai fait le choix de développer cette fonctionnalité par moi-même, ce qui m'a pris beaucoup de temps.

1-Pour la partie gestion:

- Modification de la page de gestion des attributs: `products_attributes.php`:

Seul le premier tableau (gestion des options) a été gardé. Il permet la gestion des attributs. Il a fallu pour cela modifier les tables sur lesquelles s'effectuent les requêtes, en utilisant la table `products_attributes`, et en cas de demande d'effacement, requête à la table `products_attributes_valeurs` également

- Dans le formulaire d'ajout/édition de produit

Récupération de la liste des attributs facultatifs dans la base, et écriture d'un champ par attribut. Ce champ a le nom '`products_attributes_values`' [`id_attribut`], ce qui permet lors du post, la récupération de la valeur du champ et de l'id pour l'insertion dans la base.

- Modification de la page `admin_sand/categories.php` pour chaque cas (insertion mise à jour...). Pour la mise à jour, le choix à été fait d'effacer les attributs existants dans la table avant, puis de réinsérer les nouveaux, du fait que ceux ci sont facultatifs.

2-Pour la partie boutique:

- Dans la page de détail du produit `product_info.php`, requête sur les tables `products_attributes` et `products_attributes_valeurs` pour récupérer le nom des attributs et leurs valeurs

pour le produit sélectionné. L'affichage se fait ensuite (sous la description), par une boucle.

- Particularité pour les contenances

1-Pour la partie gestion:

- Ajout d'un bouton "nouvelle contenance" lors du clic sur un produit dans la partie gestion de la boutique de l'administration (admin_sand/categories.php)

Le formulaire contient les données (nom, description, marque, photo et attributs) du produit d'origine grâce au passage par get de l'id. Les champs nom, marque, photo, les attributs (sauf la contenance) et éventuellement description, ne pourront pas être modifiées (ajout d'un readonly=true sur ces champs). Le paramètre action de l'url est 'new_product_contenance'.

- Le fait de prévisualiser le produit ajoutera l'id du produit d'origine dans un champ caché. C'est grâce à ce champ que l'on sait que l'on ajoute une contenance.

- Si retour est cliqué, le paramètre GET action sera insert_product_contenance, et l'id produit d'origine reviendra en GET.

- Lors de l'insertion du produit, on récupère l'id commun s'il existe, sinon on le crée (= id du produit d'origine), et on ajoute une ligne dans la table products commun.

- Lors de la modification de l'un des produits, on modifie le nom, marque, description, photo, attributs (sauf la contenance) de tous les produits identiques.

2-Pour la partie boutique:

- Ajout de la contenance dans les noms de produits affichés

- Affichage de la liste déroulante des différentes contenances dans les listes de produits:

- Modification des requêtes sur les produits pour obtenir, dans les cas de produits identiques et de contenance différente, un seul des produits (requête qui utilise une sous requête permettant le regroupement de ces produits). Lors de l'affichage, en cas de présence du champ commun_id, récupération grâce à une fonction tep_get_all_products_contenance des ids_products et des contenances, et affichage par une boucle de la liste déroulante. J'aurais souhaité pouvoir afficher les contenances dans l'ordre croissant, je ne l'ai pas fait par manque de temps (actuellement tri par l'id du produit), la requête qui renvoie l'id commun dans la liste des produits récupère le premier id produit qu'elle trouve.

- Création d'une fonction Ajax change_prix appelant le script contenance_ajax.php qui permet la récupération du prix grâce au passage en paramètre de l'idProduit lors de la sélection par l'internaute d'une autre contenance. Puis, dans cette fonction, grâce à l'utilisation du DOM, modification des liens de la ligne du produit, permettant l'accès à la page de détail du produit, ou l'achat.

- Modification de cette fonction pour l'intégrer à la page de détail du produit.

- L'utilisation du DOM en javascript est complexe à mettre en œuvre si l'on souhaite l'intéropérabilité des navigateurs. Dans mon cas j'ai souhaité le fonctionnement sur Internet Explorer et Mozilla, et l'arbre xml/dom n'est pas interprété de la même manière dans un cas ou dans l'autre. De même L'objet Ajax n'est pas le même suivant les navigateurs.

- Lors de l'utilisation de cette fonction, il persiste un problème d'interprétation des caractères spéciaux. C'est le cas de €, contenu dans le résultat de la requête Ajax. Ce problème n'a pas été résolu.

- Processus de la commande

- Tarif de la livraison ajouté dans le panier

- Fusion des pages livraison et paiement: (catalog/checkout_payment.php)

1- Affichage du montant total de la commande avec les frais de livraison,

2- Bloc d'affichage de l'adresse de livraison avec ajout d'une case à cocher pour recevoir la facture à l'adresse par défaut, si une autre adresse est sélectionnée. La livraison n'étant possible

qu'en France, vérification sur le pays de l'adresse indiqué lors de la validation du formulaire.

3- Bloc d'affichage du choix du mode de paiement (chèque ou CB)

- Page de confirmation

- Intégration de la contribution oscommerce Atos et installation d'une solution Atos Sips de test de la société générale:

1- Installation de la contribution oscommerce Atos: notamment

- Ajout du script catalog/atos_response.php (qui est le fichier de retour après le passage à la banque) et
- Ajout du script catalog/includes/modules/payment/atos.php qui contient les fonctions nécessaires à l'utilisation du module fourni par la banque,
- Création d'un dossier atos qui contiendra les certificats et binaires nécessaires. Ce dossier est protégé par un htaccess, les données contenues étant confidentielles.
- Modification du fichier checkout_confirmation.php pour permettre l'affichage des logos en cas de paiement par carte

2-Installation du module Atos Sips Sogenactif:

- Ajout des fichiers certif.fr.014213245611111, parmcom.014213245611111 , parmcom.sogenactif ainsi que les fichiers binaires request.exe et response.exe qui permettent la construction des requêtes d'envoi et de retour. Les certificats sont des certificats de test.
- Lors de l'activation du module via l'administration du site, le fichier atos/pathfile.parmcom.sogenactif doit se créer automatiquement et indiquer les chemins absolus vers les dossiers dont atos a besoin. Pour moi, cela n'a pas été le cas. J'ai dû le créer manuellement à partir du fichier d'exemple fourni par le module Sogenactif :
D_LOGO!/cheminRelatifVersLeDossier/logo!
F_CERTIFICATE!D:/sites/chemin.../atos/certif!
F_PARAM!D:/sites/chemin.../atos/parmcom!
F_DEFAULT!D:/sites/chemin/atos/parmcom.sogenactif!

- *Validation d'une commande* : on arrive sur le site de test de la Sogenactif, il suffit d'entrer un n° de carte (ex : 4974934125497800) avec un cryptogramme 000 et une date ultérieure à la date courante pour que le paiement soit validé, la banque nous renvoie ensuite sur la page includes/checkout_success.php. En cas de saisie du n° 4972187615205 , le paiement est refusé : retour sur la page de choix du paiement checkout_paiement.php.

- Divers:

- Lors de l'upload d'une image, modification du nom des images (lors des créations ou modifications de produit) en ajoutant l'idProduit au nom d'origine, ce qui évite les écrasements d'images différentes portant le même nom.