Architecture et dev web
TEA – Site Web shop

But de ce TEA :

Ce TEA, vous permet de voir la construction d'un projet web de e-commerce dans son ensemble.

Brief de l'entreprise
L'entreprise Open shop souhaite présenter ses produits et si possible les vendre en ligne. Elle vous fait confiance pour la création d'un site simple et intuitif. Ses produits sont répertoriés en catégories pour faciliter les recherches du client

Le backlog suivant a été produit

N°	tâche	Priorité	Note
1	En tant qu'utilisateur, je souhaite voir 3 produits en page d'accueil	1	
2	En tant qu'utilisateur, je souhaite consulter tous les produits	1	
3	En tant qu'utilisateur, je souhaite consulter les caractéristiques d'un produit	1	
4	En tant qu'utilisateur, je souhaite consulter les produits d'une catégorie	2	
5	En tant qu'utilisateur, je souhaite commander un produit	2	
6	En tant qu'utilisateur, je souhaite créer un compte	1	
7	En tant qu'utilisateur, je souhaite voir mon panier d'achats	2	
8	En tant qu'utilisateur, je souhaite modifier mon panier d'achats	2	
9	En tant qu'administrateur, je souhaite pouvoir ajouter, supprimer les produits	1	
10	En tant qu'administrateur, je souhaite voir toutes les commandes	3	
11	En tant qu'administrateur, je souhaite avoir des informations sur les utilisateurs ayant commandé	3	

I. Création de la base de données

1. Récupérez sur moodle le dossier SiteWebShop et sauvegardez le sur votre compte.
2. Créez la base de données *SiteWebShop* avec un interclassement pour la connexion MySql : *utf8-general_ci*
   Voici les différentes tables

Article
id_article: int
id_categorie : int
designation : varchar(128)
prix : decimal(6,2)
tva : decimal(5,1) valeur par défaut :19.6
description : text
img_article : varchar(64)

categorie
id_categorie : int
nom : varchar(128)

3. Créez ces 2 tables sous phpmyadmin. N'oubliez pas la clé étrangère.

4. Ajoutez les enregistrements en utilisant le fichier contenuTables.sql. Dans *PhpMyAdmin*, faites « Importer » et choisissez le fichier.

5. Visualisez le contenu de ces tables et plus particulièrement le champs img_articles. Cela correspond il à votre arborescence. Modifiez celle-ci ci nécessaire

6. Pour éviter de perdre cette base de données, exportez son contenu et sauvegardez le fichier sur votre compte

II. Création des différentes pages du site

1. Visualisez l'index de ce site.

II.1. Fichiers d'include et tests

2. Visualisez les images *resultat_index.png*, *resultat_login.png*, *resultat_creer_compte.png* , *resultat_panier.png* , *resultat_admin.png*. Vous pouvez remarquer que l'entête et le pied de page sont identiques sur toutes les futures pages. Si une modification doit être apportée à l'entête et le pied de page, il est préférable d'avoir un fichier pour l'entête et un fichier pour le pied de page et d'inclure ces 2 fichiers dans d'autres fichiers
3. Créer un fichier header.php qui comprend le header de la page index.
4. Créer un fichier footer.php qui comprend le footer de la page index.
5. Supprimez dans le fichier index.php le header et le footer et remplacez ces éléments par l'inclusion de fichier (require). Testez le bon fonctionnement
6. Créez le fichier *panier.php de la même manière que l'index.php*.

II.2. Affichage de 3 articles au hasard sur la page index.

Le but de cette partie est de compléter le fichier index.php sous les commentaire <!--Affichage de 3 articles au hasard -->.

1. Créez le fichier utile.php qui contient le code de la fonction ci dessous. Cette fonction permet de tronquer du texte pour permettre de faire afficher qu'une petite partie de la description.
2.

```
function tronquer_texte($texte, $longeur_max = 100)
      {
          if (strlen($texte) > $longeur_max)
          {
                    $texte = substr($texte, 0, $longeur_max);
                    $texte .= "...";
          }
          return $texte;
      }
```

3. Dans le fichier index.php, incluez le fichier utile.php.

4. Réutilisez le fichier connexion.php permettant de vous connecter à la base de données créée et incluez ce fichier.

5. Faites afficher le contenu de 3 articles au hasard sous forme d'une liste non numérotée. La liste aura comme identifiant « product-list » et chaque <li> aura la classe « product ». Regardez le modèle resultat_index.png pour voir comment les éléments sont afficher.
   Remarques :
   - `$requete = "SELECT * FROM article ORDER BY RAND() LIMIT 3";`
   - le texte de la description sera tronqué en utilisant la fonction tronquer_texte
   - le texte « Voir les détails » est un lien vers la page vue_produit.php qui sera réalisée dans la partie suivante. Ce lien aura un passage de paramètre par url. Le paramètre passé correspond à l'identifiant de l'article.
     Exemple de lien :
     `<a href="vue_produit.php?article=22"> Voir les détails</a>`
     avec le numéro 22 issu de la base de données

III. Création de la page vue_Produit

Cette page permet de faire afficher les détails d'un article mais également de commander celui ci en choisissant le nombre d'article souhaité (Voir resultat_vue_Produit.jpg) Regardez également l'url, dans cette exemple c'est l'article 10 qui a été sélectionné.

1. Créez cette page comme la page d'index.
2. Au dessus du body, vous devez tester si un paramètre a été envoyé et récupérer cette valeur. Si un paramètre existe, il correspond à l'article devant être visualisé. Il suffit de faire une requête et de récupérer le renvoi de cette requête dans une variable $article par exemple.

```
if ( !empty($_GET['article']) )
{
        $numArticle = $_GET['article']);


        ……
        $article= ….
}
```

3. Dans la balise html <article>, affichez le contenu de la variable $article ainsi qu'un formulaire de commande permettant de choisir le nombre d'articles souhaité (de 1 à 5)
voir modèle ci-dessous

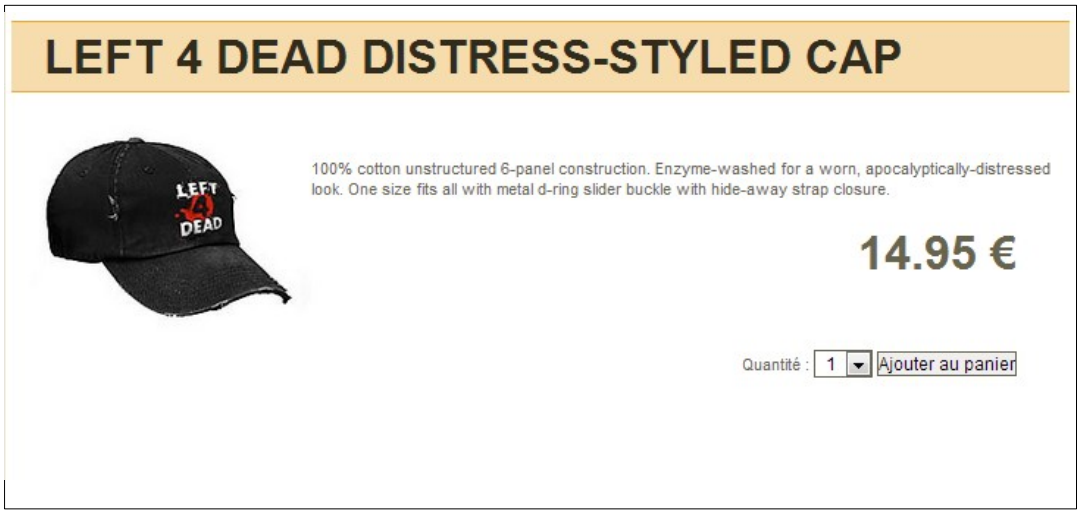
Schéma des balises (par rapport au code CSS)

Header h2

article

p

form



IV. *création de la page categorie.php.*

Chaque page a le sous-menu suivant : tous les produits | Vêtements | Accessoires | posters | DVD
L'utilisateur a la possibilité de visionner tous les articles concernant par exemple les DVD. Nous aurions pu faire 5 pages différentes mais pour des soucis de maintenance nous avons choisi de faire un passage de paramètre par URL.

- « tous les produits » aura un lien vers categorie.php?cat=all
- « Vêtements » aura un lien vers categorie.php?cat=1
- «Accessoires» aura un lien vers categorie.php?cat=2

1. Créez la page *categorie.php*. Cette page devra intégrer le header et le footer comme les pages précédentes. Dans la section, vous devez comme sur la page vue_produit.php, tester si un paramètre a été passé et si c'est le cas vous devez faire afficher tous les articles correspondants à la catégorie sélectionnée.

V. Enregistrement dans un cookies

1. Lorsque un utilisateur clique sur une catégorie, faites en sorte de stocker la valeur de la catégorie dans un cookie.
2. Si le cookie existe , faites en sorte que les trois articles affichés au hasard sur la page index soit de cette catégorie.

VI. Administration du site

Le but de ce manipulation est de créer le « back office » c'est à dire l'espace administrateur du site web de commerce en ligne pour permettre d'administrer la base de données des articles. Les pages seront accessibles par le lien Admin en bas à droite du site

VI.1. Création de l'espace Admin

- Créez dans le dossier "SiteWebShop" un dossier « admin » qui contiendra tout votre espace d'administrateur
- Créez dans ce répertoire un fichier index.php qui contient dans un div identifié « container » le titre <h1> Administration du site OpenShop</h1> . Liez ce fichier avec le fichier styleadmin.css que vous trouverez sous moodle.
- Modifiez le fichier « footer.php » du front office, pour permettre d'accéder au back office c'est à dire au fichier index.php du dossier admin.
- Créez le formulaire identique au fichier « indexAdmin.png ».
  Remarques
  - la validation du formulaire fera appel de nouveau au fichier index.php du dossier admin, l'envoi se fera part par la méthode POST
  - le contenu du select (c'est à dire vêtements accessoires ect…) est issu d'une requête permettant de récupérer les enregistrements de la table « Categorie »,
- Au dessus du formulaire, vous allez faire le traitement de celui-ci, vous devez :
  - Vérifiez si tous les champs ont été définis (fonction isset()) si c'est le cas vous devez récupérer la valeur de chaque champ dans une variable sinon mettre cette variable comme une chaine vide.
  - Vérifiez également si tous les champs ne sont pas vides. Si c'est le cas, les informations saisies seront stockées dans la base de données et l'image sera téléchargée sur le serveur. Un message devra être affiché si l'enregistrement a bien été effectué. Le téléchargement de l'image sera réalisé avec la fonction : move_uploaded_file(). Vous avez le droit d'insérer cet élément uniquement si le produit n'existe pas dans la base de données.

VI.2. Sécurisation de l'espace Admin

Sécurisez votre espace admin en utilisant une table user et les sessions

VII. Gestion des clients

VII.1. Base de données : table Client

Cette partie a pour but de créer des comptes « clients » pour qu'ils puissent acheter en ligne les produits de la société SiteWebShop. Il est nécessaire dans un premier temps d'avoir un lieu de stockage pour sauvegarder les informations du compte utilisateurs. Vous allez créer une table correspondant aux informations du compte client.

Client
id : int (Auto incrément)
email ; varchar(64)
motPasse : varchar(32)
civilite : varchar(32)
nom : varchar(64)
prenom : varchar(64)
adresse : varchar(255)
codePostal : int(10)
ville : varchar(64)
pays : varchar(32)
telephone : int(10)

Structure de la table client

- ✗ Avec PhpMyAdmin, exécutez le fichier client.sql dans votre base de données SiteWebShop. Si vous utilisez les machines universitaires il est nécessaire de réinstaller la base de données du site sitewebshop. Si vous n'avez pas sauvegarder le fichier vous pouvez utiliser le fichier siteWebShop.sql.
- ✗ Créez 2 clients. Le mot de passe doit être codé en sha256. Utilisez par exemple ce site http://www.xorbin.com/tools/sha256-hash-calculator pour encoder votre password.

VII.2. Création de la page de création de compte (creer_compte.png)

Le but de cette partie est de permettre la création des comptes clients.

Vous allez modifier dans cette partie le fichier creer_compte.php pour permettre la création d'un compte sur le site

- Lorsque l'utilisateur valide le formulaire, la page « creer_compte.php » est de nouveau appelée.

- Ecrivez le code PHP avant le doctype dans la partie traitement formulaire qui devra :
  - Vérifier que tous les champs ont été saisis. Si c'est le cas, récupérez le contenu de l'input dans une variable, sinon la variable est une chaine vide.
  - Corrigez le problème des ' en utilisant la fonction add*Slashes()*
  - Vérifier que l'email saisi n'existe pas dans la table client (client déjà enregistré) pour simplifier l'écriture, créez une fonction MailDansBase($mail) qui retourne vrai si le email est présent sinon retourne false
  - Encoder le mot de passe de votre client avant insertion dans la base de données avec la fonction hash() qui va crypté votre mot de passe :
    $motPasse=hash('sha256',$_POST['motPasse']);
  - Si tous les champs vérifiés ci-dessus sont corrects,
    - insérer le client dans la table,
    - envoyer un mail de confirmation d'inscription avec la fonction *mail() (cette fonction sera mise en commentaire par la suite par rapport au problème évoqué en TP)*,

- rediriger le client vers la page «login.php »

Remarques

- Si il y a des erreurs, il est nécessaire de ressaisir toutes les données …. ce n'est pas concluant …. Pour cela, il suffit de remplir l'attribut *value* des balises input avec le contenu de la variable associée au champ..

VII.3. Création de la connexion

✗ Vous allez modifier le fichier login.php

Remarque : La validation de ce formulaire fait appel à lui même

✗ Complétez ce fichier pour
- qu'il vérifie si le couple login/password saisi est identique au contenu de la base
- Si identité alors on stocke dans la SESSSION l'identifiant, le nom, le prénom et la civilité du client et retourne à la page index.php sinon on affiche un message d'erreur et on reste sur la page login.php

VII.4. Modification des pages

Une fois que le client est identifié, il est nécessaire d'ajouter dans le ul du nav id= "menu " un li contenant le texte : Bonjour Civilité de la personne et Nom de la personne et un second li contenant un lien vers un fichier de déconnexion.

Pour faire ceci vous devez :

- Démarrer une session dans la page index.php
- tester si une SESSION existe et n'est pas vide dans le fichier header.php et si c'est le cas créer les 2 li
- Modifier également les pages vue_produit.php et catalogue.php

VII.5. Déconnexion

✗ Créez le fichier deconnexion.php

VIII. Gestion de la commande

Vous allez dans cette partie construire les tables nécessaire à la gestion de la commande d'un client

VIII.1. Création des tables

Lorsqu'un client valide sa commande donc son panier, les informations devront être stockés dans la base de données dans ces 2 tables

- Créez ces tables sous phpmyadmin

Ligne_commande
id_commande : int
id_article: int
qte_article : int

Commande
id_commande : int (Auto-increment)
id_client : int
date : datetime
prix_total : decimal(9,2)

VIII.2. Commande.php

Une commande ne peut être effectué par un client si et seulement si :
- un client est identifié
- des articles sont dans le panier

Si le client n'est pas identifié alors il sera redirigé vers la page de connexion
Si il n'y a pas d'article alors le client sera redirigé vers la page d'index
Sinon, une commande sera créée et chaque élément du panier sera un enregistrement de la table ligne-commande. Lorsque tous les articles seront stockés dans la base il sera nécessaire de supprimer le panier(unset())

Lorsque la commande sera bien enregistrée, un mail sera envoyé au client et le client sera redirigé vers la page paiement.php

IX. Paiement de la commande

Il existe différent moyen de payer une commande. La solution simple est d'utiliser un compte stripe. Il existe une solution permettant de simuler ce paiement.