

« Conception et réalisation d'un site web E-commerce sous nom : luxforAll »



Réaliser Par :

Kamal Chokrane

Ayoub El Bouinany

Encadrer par :

Fatimaezzahra Sadraoui

Meryem Faouzi

1 Remerciements

En préambule à ce mémoire on remercie Dieu qui nous a aidés et nous a donné la patience et le courage durant ces longues durées d'étude.

Aussi nos remerciements au corps professoral et administratif de YouCode qui déploient de grands efforts pour nous assurer une très bonne formation. nous remercions sincèrement Mme Meryem Faouzi et Mme Fatimaezzahra Sadraoui notre encadreuses, qui sont toujours montrée disponible tout au long de la réalisation de ce projet, ainsi pour l'inspiration, l'aide et le temps qu'elles ont bien consacrer.

2 Sommaire

1	Remerciements	2
2	Sommaire.....	3
3	Chapitre 1: Contexte général du projet.....	5
3.1	Cahier de Charge :	5
3.1.1	présentation de la problématique:	5
3.1.2	Solution :	6
3.1.3	Backoffice (espace admin) :.....	7
4	Chapitre 2: Analyse et Conception	8
4.1	Diagramme de cas d'utilisation :	9
4.1.1	Définition :.....	9
4.1.2	Diagramme de cas d'utilisation de notre site web:	10
4.2	Diagramme de classe :	11
4.2.1	Définition :.....	11
4.2.2	Diagramme de classe de notre site web :	12
5	Chapitre 3: Réalisation de l'application	14
5.1	Les outils de développement :.....	14
5.1.1	Xampp :.....	14

5.1.2	HTML :	14
5.1.3	CSS :	15
5.1.4	BOOTSTRAP :	15
5.1.5	Sass :	16
5.1.6	PHP :	16
5.1.7	MYSQL :	17
5.1.8	JAVASCRIPT :	17
5.1.9	JQUERY :	18
5.2	Présentation de l'application :	18
5.2.1	L'en-tête du site :	18
5.2.2	La page d'accueil :	18
5.2.3	Page de produit :	21
5.2.4	Contacter nous :	22
5.2.5	Authentification :	24
5.2.6	produit détail:	26
5.3	Présentation de l'espace administrateur :	27
5.3.1	Authentification :	27
5.3.2	Page liste des commandes:	28
5.3.3	Page de gestion:	28
5.3.4	Page Messages :	29
5.4	Les techniques et les fonctions utilisées:	30
5.4.1	L'utilisation des sessions et des variables superglobales :	30
5.4.2	La connexion à la base de données MySQL :	33
5.4.3	Les structures conditionnelles :	34
5.4.4	Boucle de parcours (Foreach) :	35
5.4.5	Les expressions régulières :	36
5.5	Application Panier :	37

5.5.1	Requête d'insertion d'un client :	44
5.5.2	Requête d'insertion des produits achetés par un client :	44
5.5.3	Formulaire pour laisser des commentaires :	44
5.6	Backoffice:	46
6	Conclusion :	49

3 Chapitre 1: Contexte général du projet

3.1 Cahier de Charge :

3.1.1 présentation de la problématique:

Des ventes de mains en mains, vers des ventes virtuelles, passent les priorités des opérations de ventes des biens et des services, ce qui nous rend obligés de donner plus d'importance à la vente électronique. Les sites de vente en ligne permettent

aux clients de profiter d'une foire virtuelle disponible est quotidiennement mise à jour sans la moindre contrainte, ce qui leur permettra de ne jamais rater les coups de cœur, ainsi une foire sans problèmes de distance géographique, ni d'horaire de travail ni de disponibilité de transport. D'une autre part ces sites offrent à la société de profiter de cette espace pour exposer ses produits à une plus large base de clientèle. Grace à un site web e-commerce, on peut choisir et payer des articles comme dans un magasin réel. Pour acheter un produit d'une boutique virtuelle, il suffit le plus souvent de choisir les produits désirés puis de les mettre dans un panier d'achat. La commande sera livrée en fonction du choix de l'internaute et selon les modalités définies par le responsable de la boutique. Notre projet est réalisé dans le cadre d'améliorer nos compétences en langage de programmation web PHP ayant comme objectif principal :la conception et la création d'un site web e-commerce « luxforAll » pour la vente en ligne.

3.1.2 **Solution :**

Ce projet consiste à la mise en place d'un site Web dynamique qui gère la commercialisation des vêtements et accessoires utilisés. Ceci est possible à travers des catalogues en ligne proposant ces produits aux meilleurs prix par rapport aux prix

initiales. La société n'aura donc qu'à agencer ses produits et bien sûr de mettre sa base de données à jour. Le client est toujours anonyme mais pour pouvoir passer à un stade plus rigoureux, il faut qu'il s'inscrive, cela se fait uniquement pour la première commande mais après, notre client peut s'authentifier avec son E-mail et son mot de passe pour passer d'autres commandes. Le client a le droit de consulter tous les produits et ajouter ce qu'il veut à son panier. Et après le choix d'un produit le client doit mentionner la quantité qui s'ajoute automatiquement à son panier avec le prix unitaire et le prix total afin de finir sa commande. Alors ce site web offre beaucoup des services à savoir : - Consulter les produits. Laisser des commentaires sur les produits. -Modifier le panier en ajoutant ou supprimant des produits. -Consulter le profil. - Lancer une commande en ligne contient ces différents produits, qui sont par la suite livrés à domicile.

3.1.3 Backoffice (espace admin) :

Le backoffice regroupe un ensemble de tâches administratives, ce dernier doit également permettre une gestion quotidienne des activités e-commerce (suivi des commandes,

mise à jour des catalogues produits) sans avoir besoin de l'intervention de l'agence.

Parmi les tâches de gestion backoffice nous pouvons citer :

- *Ajout de produits*
- *Modification des produits*
- *Suppression des produits*
- *Gestion de stock*
- *Liste des commandes*
- *Gestion des messages*

4 Chapitre 2: Analyse et Conception

4.1 Diagramme de cas d'utilisation :

4.1.1 Définition :

Les rôles des diagrammes de cas d'utilisation sont de recueillir, d'analyser et d'organiser les besoins, ainsi que de recenser les grandes fonctionnalités d'un système. Il s'agit donc de la première étape UML pour la conception d'un système.

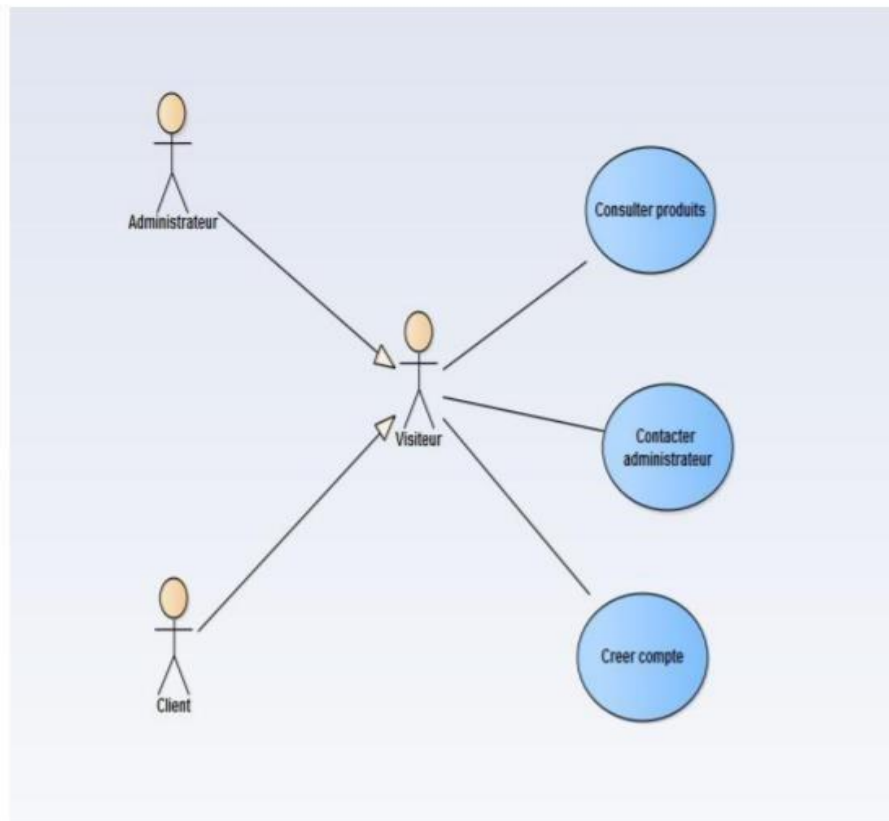
Le diagramme de cas se compose de trois éléments principaux:

Un Acteur : c'est l'idéalisation d'un rôle joué par une personne externe, un processus ou une chose qui interagit avec un système. Il se représente par un petit bonhomme avec son nom inscrit dessous.

Un cas d'utilisation : c'est une unité cohérente représentant une fonctionnalité visible de l'extérieur. Il réalise un service de bout en bout, avec un déclenchement, un déroulement et une fin, pour l'acteur qui l'initie.

Les relations : Trois types de relations sont pris en charge par la norme UML et sont graphiquement représentées par des types particuliers de ces relations. Les relations indiquent que le cas d'utilisation source présente les mêmes conditions d'exécution que le cas issu. Une relation simple entre un acteur et une utilisation est un trait simple.

4.1.2 Diagramme de cas d'utilisation de notre site web:



Le Client : cet acteur est un visiteur ayant déjà créé un compte sur notre site, il peut donc suivre le processus d'achat

des produits en toute sécurité sachant que notre système doit être l'unique responsable de la confidentialité des données personnelles de ses clients.

L'administrateur (admin) : pour les sites web on l'appelle généralement « le webmaster ». C'est celui qui assure le dynamisme du site et veille sur les mises à jour des produits, de leurs prix, de leurs disponibilités, de la gestion des paiements et la gestion des livraisons.

4.2 Diagramme de classe :

4.2.1 Définition :

Un diagramme de classes UML décrit les structures d'objets et d'informations utilisées sur notre site web, à la fois en interne et en communication avec ses utilisateurs. Il décrit les informations sans faire référence à une implémentation particulière. Ses classes et relations peuvent être implémentées de nombreuses manières, comme les tables de bases de données.

4.2.2 Diagramme de classe de notre site web :

En général un diagramme de classe peut contenir les éléments suivants :

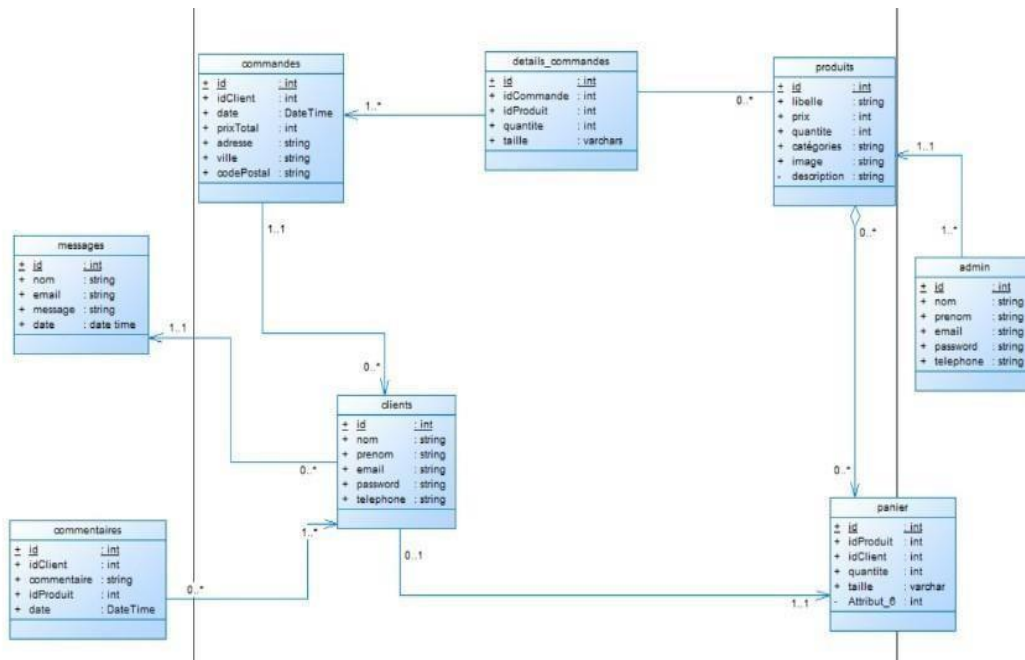
Les classes: une classe représente la description formelle d'un ensemble d'objets ayant une sémantique et des caractéristiques communes. Elle est représentée en utilisant un rectangle divisé en trois sections.

La section supérieure est le nom de la classe, la section centrale définit les propriétés de la classe alors que la section du bas énumère les méthodes de la classe.

Les associations : une association est une relation entre deux classes (association binaire) ou plus (association n-aire), qui décrit les connexions structurelles entre leurs instances. Une association indique donc que des liens peuvent exister entre des instances des classes associées.

Les attributs : les attributs représentent les données encapsulées dans les objets des classes. Chacune de ces informations est définie par un nom, un type de données, une visibilité et peut être initialisé.

Le nom de l'attribut doit être unique dans la classe.



5 Chapitre 3: Réalisation de l'application

5.1 Les outils de développement :

5.1.1 Xampp :

XAMPP est un ensemble de logiciels permettant de mettre en place facilement un serveur Web local, un serveur FTP et un serveur de messagerie électronique. Il s'agit d'une distribution de logiciels libres (X (cross) Apache MariaDB Perl PHP) offrant une bonne souplesse d'utilisation, réputée pour son installation simple et rapide.

5.1.2 HTML :

L'HypertextMarkupLanguage, généralement abrégé HTML, est le format de données conçu pour représenter les pages web. C'est un langage de balisage permettant d'écrire de l'hypertexte, d'où son nom. HTML permet également de structurer sémantiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images, des formulaires de saisie, et des programmes informatiques. Il est souvent utilisé conjointement avec des

langages de programmation (PHP, JavaScript...) et des formats de présentation (feuilles de style en cascade).

5.1.3 CSS :

l'anglais: Cascading Style Sheets, forment un Les feuilles de style en cascade, généralement appelées CSS de langage informatique qui décrit la présentation des documents HTML et XML Les standards définissant CSS sont publiés par le World Wide Web Consortium (W3C). Introduit au milieu des années 1990, CSS Devient couramment utilisé dans la conception des sites web et bien pris en charge par les navigateurs web dans les années 2000.

5.1.4 BOOTSTRAP :

Bootstrap est une collection d'outils utiles à la création du design (graphisme, animation et interactions avec la page dans le navigateur, etc.) de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option. C'est l'un des projets les plus populaires sur la plate-forme de gestion de développement GitHub.

5.1.5 Sass :

Sass est un langage dynamique de génération de feuilles de style en cascade initialement développé par Hampton Catlin et Nathalie Weizenbaum.

Sass est un préprocesseur CSS. C'est un langage de description compilé en CSS.

SassScript est un langage de script pouvant être utilisé à l'intérieur du code Sass.

5.1.6 PHP :

PHP: Hypertext Preprocessor, plus connu sous son sigle PHP(acronyme récursif), est un langage de programmation principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif orienté objet. PHP a permis de créer un grand nombre de sites web célèbres, comme Facebook, Wikipédia, etc. Il est considéré comme la base de la création des sites Internet dits Dynamiques.

5.1.7 MYSQL :

MySQL est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, Informix et Microsoft SQL Server.

5.1.8 JAVASCRIPT :

JavaScript est un langage de script orienté objet principalement utilisé dans les pages HTML. À l'opposé des langages serveur (qui s'exécutent sur le site), JavaScript est exécuté sur l'ordinateur de l'internaute par le navigateur lui-même. Ainsi, ce langage permet une interaction avec l'utilisateur en fonction de ses actions (lors du passage de la souris au dessus d'un élément, du redimensionnement de la page...).

5.1.9 JQUERY :

jQuery est une bibliothèque JavaScript open-source et crossbrowser qui permet de traverser et manipuler très facilement l'arbre DOM de vos pages JQuery est une bibliothèque JavaScript open-source et cross-browser qui permet de traverser et manipuler très facilement l'arbre DOM de vos pages.

5.2 Présentation de l'application :

5.2.1 L'en-tête du site :

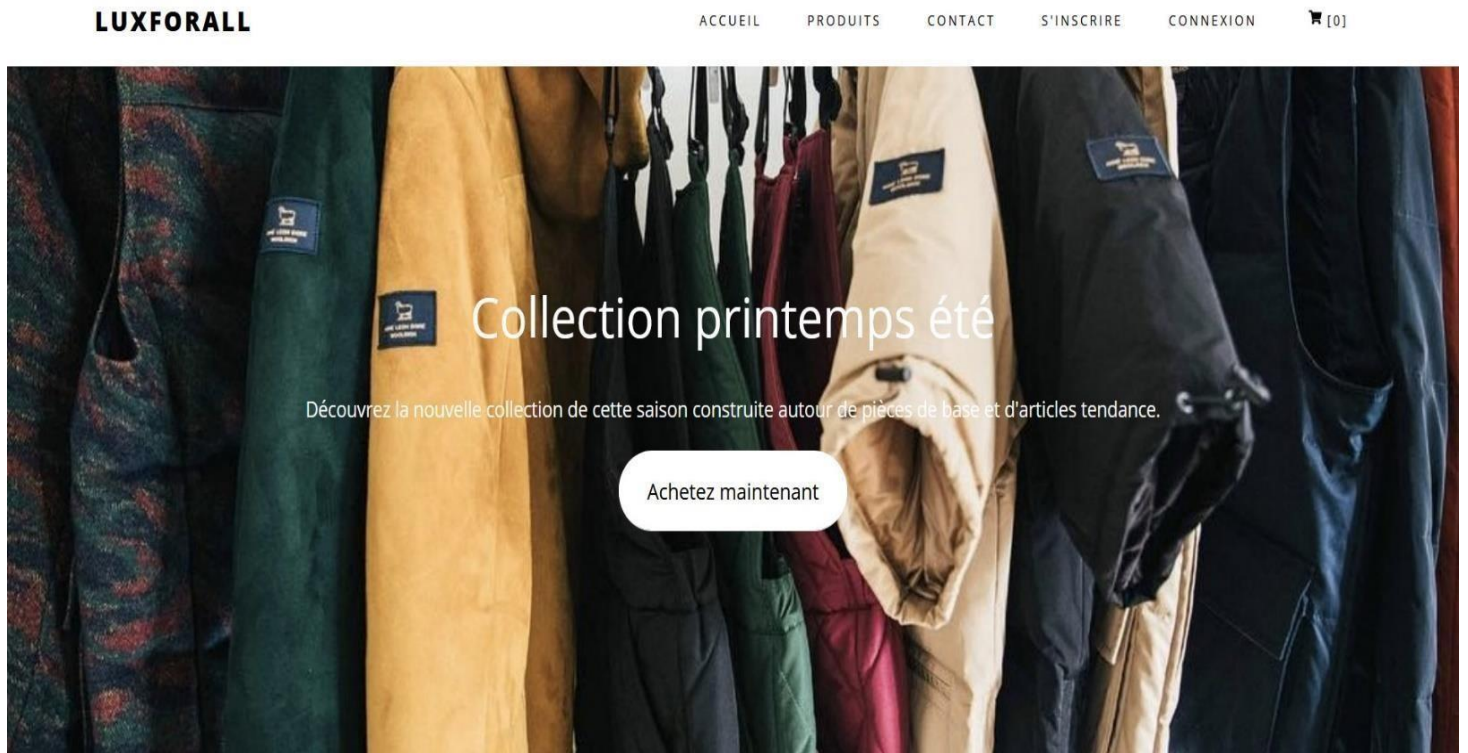
Le premier menu contient la case “connexion Or inscription” permettant soit de créer un compte soit se connecter en tant qu'un client déjà inscrit.

Et au dessous on trouve les plusieurs produits représentées par notre site comme les t-shirts , jackets, casquettes, pantalons, chaussures et contacter nous.

A droite on trouve le panier qui représente les produits réservés par le client.

5.2.2 La page d'accueil :

Cette partie représente une case pour effectuer une recherche avancée concerner pour un produit.



Meilleures ventes

Découvrez les dernières tendances ici, dès maintenant, avec nous.



Produits



T-SHIRT BLACK
~~120.00 DH~~ 80.00 DH

neuf/utilisé



VESTE PUMA
200.00 DH

neuf/utilisé



CHEMISE
140.00 DH

neuf/utilisé



T-SHIRT ZARA
~~120.00 DH~~ 80.00 DH

neuf/utilisé



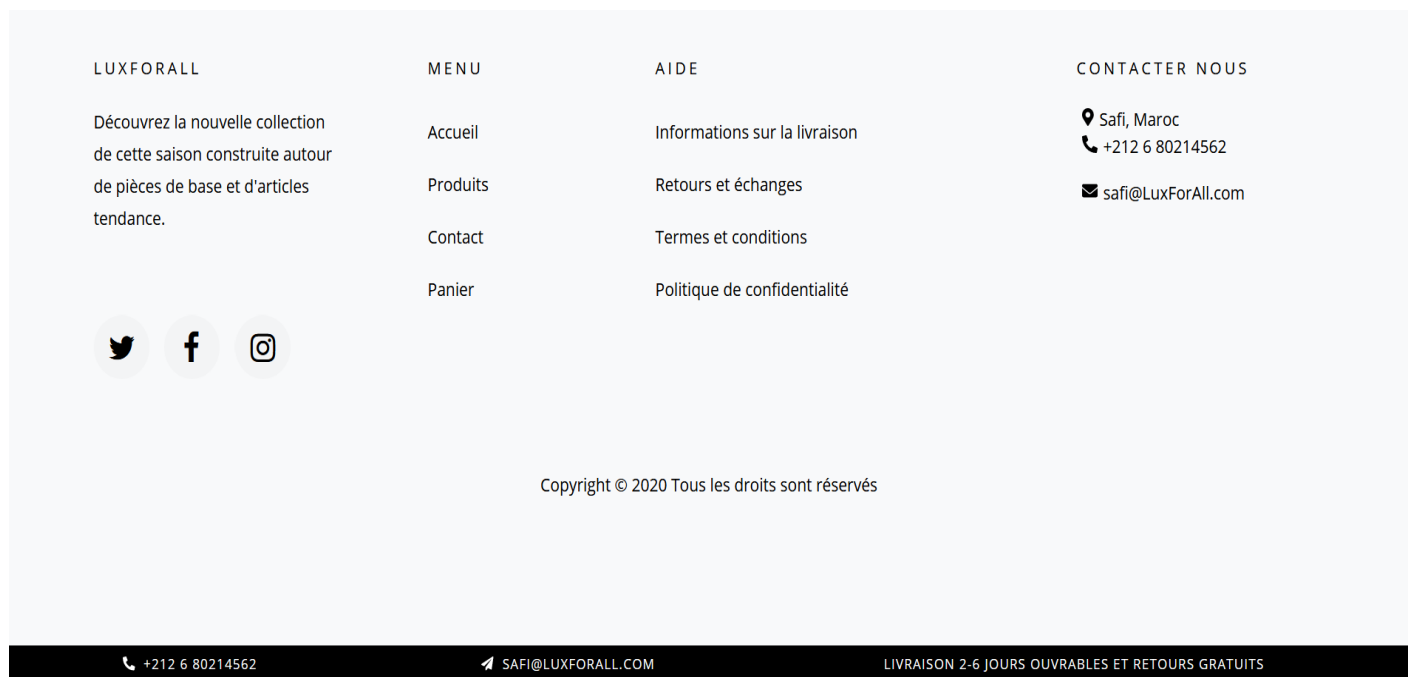
BASKET NIKE
300.00 DH

neuf/utilisé



BLOUSON MOTO FURYGAN
400.00 DH

neuf/utilisé



C'est la partie une de l'accueil de notre site qui fait représenter des produits proposés par la société, en plus une petit description pour les produits fournis par notre application.

5.2.3 Page de produit :

Lorsque tu as connecté sur notre application automatiquement rediriger au liste de nos offres.

Produits

toute les produits



neuf/utilisé



Recherche



SHORT EN JEANS TAILLE S

190.00 DHs

neuf



CASQUETTE ROUGE

289.00 DHs

neuf



CHEMISE RALPH LAUREN VERT OLIVE

199.00 DHs

utilisé

5.2.4 Contacter nous :

Cette page est réservée pour contacter l'admin de notre site web.

Contacter Nous

Notre nom

Votre Email

Message

Envoyer

5.2.5 Authentification :

Ces deux pages représente connexion et inscription pour le client.

S'identifier

Connexion

Créer un compte

Votre nom

Votre prénom

Votre Email

Mot de passe

Répéter le mot de passe

Téléphone

Enregistrer

5.2.6 produit détail:

Dans cette page on trouve le détail de chaque produit (taille, quantité en stock, avis des clients, type: neuf/utilisé...).



Short en jeans taille S

1 Vendue Type : [neuf](#)

190.00 DHs

short en jeans nouveau taille S

S ▼

- 1 +

20 pièce disponible

Ajouter au panier

5.3 Présentation de l'espace administrateur :

5.3.1 Authentification :

S'identifier Admin

Connexion

5.3.2 Page liste des commandes:

ayoub elbouinany

ayoub.elbouinany99@gmail.com

Téléphone : 0612890680


Commandes [Adresse](#)

Référence	Date	Prix total	Nombre des produits
COMM#1	2020-08-24	906.00 DHs	2
COMM#2	2020-08-24	418.00 DHs	1
COMM#3	2020-08-24	2910.00 DHs	1
COMM#4	2020-08-25	3199.00 DHs	1

5.3.3 Page de gestion:

ayoub kamal
ayoub.admin@gmail.com

Téléphone : 0612890680

Libelle	Prix	Quantite	Image	Type	Description	Traitement
Short en jeans taille S	190.00 DHs	20		neuf	short en jeans nouveau taille S	SUPPRIMER MODIFIER
AJOUTER						

5.3.4 Page Messages :

1 Messages



Ahmed

Bonjour admin J'ai un t-shirt Zara que je veux vendre

2020-09-10 07:25:37



5.4

Les techniques et les fonctions utilisées:

5.4.1 L'utilisation des sessions et des variables superglobales :

On a utilisé plusieurs variables superglobales (\$_GET, \$_POST, \$_SESSION ...).

-Les superglobales \$_GET et \$_POST : sont utilisées pour manipuler les informations envoyées via un formulaire HTML. \$_GET stockera les valeurs lorsque le formulaire sera envoyé via la méthode GET tandis que \$_POST stockera les valeurs lorsque le formulaire sera envoyé via la méthode POST.

htmlspecialchars : est pratique pour éviter que des données fournies par les utilisateurs contiennent des balises HTML, comme pour un forum ou un chat. Cette fonction prend un deuxième argument optionnel, qui indique comment doivent être traités les guillemets doubles et simples.

```
        if(isset($_POST['inscrire']) &&
!empty($_POST['inscrire']))
{
    $nom =htmlspecialchars($_POST['nom']);
    $prenom = htmlspecialchars($_POST['prenom']);
    $email = htmlspecialchars($_POST['email']);
    $telephone =htmlspecialchars($_POST['telephone']);
    $password
=htmlspecialchars($_POST['password']); }

```

Ici, on définit la méthode POST comme méthode d'envoi du formulaire et la page inscrire.php comme page de traitement des données (la page où les données du formulaire devront être envoyées).

Dans le cas présent, cette page est la même que la page dans laquelle se situe le formulaire, ce qui signifie que les données seront envoyées vers notre page courante, et ils seront insérées dans notre base de données.

-\$SESSION : Cette superglobale est un tableau associatif qui stocke les différentes variables de sessions avec leurs noms en index du tableau et leurs valeurs en valeurs du tableau. On a créé quelques variables de sessions dans notre fichier connexion.php :

```

$_SESSION['client_id'] = $client['id'];
    $_SESSION['client_nom'] = $client['nom'];
    $_SESSION['client_prenom'] =
$client['prenom'];
    $_SESSION['client_email'] = $client['email'];
    $_SESSION['client_telephone'] =
$client['telephone'];

```

Pour activer les sessions on va utiliser la fonction
session_start();

```

<?php require 'dbconn.php';
session_start();

```

Et pour terminer les sessions on va utiliser ces deux fonctions
session_destroy() , session_unset() .

session_destroy() : détruit toutes les données associées à la
session courante session_unset() : détruit toutes les variables
d'une session.

```

<?php

include('config.php');
session_unset
session_destroy();
    header("location: ../index.php"); ?>

```


5.4.2 La connexion à la base de données MySQL :

On peut utiliser l'extension PDO pour se connecter à la base de données MySQL.

```
$db = new  
    PDO('mysql:host='.$DATABASE_HOST.';dbname='.$DATABASE_NAME  
        .';charset=utf8mb4', $DATABASE_USER, $DATABASE_PASSWORD);
```

L'utilisation des exceptions :

Utiliser des exceptions va nous permettre de gérer les erreurs de manière plus fluide et de personnaliser la façon dont un script doit gérer certaines erreurs.

Ici on commence par établir la connexion vers la base de données MySQL en utilisant le PDO. Si la connexion est échouée une exception de la classe PDOException est également lancée. On capture cette erreur dans le bloc catch juste en dessous et on affiche les informations relatives à l'erreur avec la méthode getMessage() de la classe PDOException.

```
try {
```

```

$db = new
PDO('mysql:host='.$DATABASE_HOST.';dbname='.$DATABASE_NAME.';charse
t=utf8mb4', $DATABASE_USER, $DATABASE_PASSWORD);
$db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
$db->setAttribute(PDO::ATTR_EMULATE_PREPARES, false);
} catch (PDOException $e) {      echo
"Connection failed : ".
$e->getMessage();
}

```

5.4.3 Les structures conditionnelles :

La condition if est l'une des plus importantes et des plus utilisées

dans l'ensemble des langages de programmation, elle va nous permettre d'exécuter un bloc de code si et seulement si le résultat d'un test vaut true.

Ici, on utilise deux conditions if dans le fichier connexion.php, pour vérifier si le client saisie ses informations correctement, on utilisant ses données dans la table client dans notre base de données.

```

if($client === false){
    $erreur .= "Email insérée n'existe pas<br>";
} else{
    $validPassword = password_verify($password,
$client['password']);
    if($validPassword){

        $_SESSION['client_id'] = $client['id'];
        $_SESSION['client_nom'] = $client['nom'];
        $_SESSION['client_prenom'] =
$client['prenom'];
        $_SESSION['client_email'] = $client['email']
;
        $_SESSION['client_telephone'] = $client['tele
;
        header('Location: produits.php');
    } else{
        $erreur .= "Mot de passe est incorrect<br>";
    }
}
}

```

5.4.4 Boucle de parcours (Foreach) :

```

<?php foreach ($commandes as
$commande) { ?>

    <div class="row">
    <div class="col-md-3">
    <p>COMM#<?php echo
$commande['id'] ?></p>
    </div>
    <div class="col-md-3">
    <p><?php echo

```

```

$commande['adresse'] ?></p>
</div>
<div class="col-md-3">
<p><?php echo
$commande['ville'] ?></p>
</div>
<div class="col-md-3">
<p><?php echo
$commande['codePostal'] ?></p>
</div>
</div> <?php } ?>

```

La boucle foreach nous permettra de parcourir les tableaux. Ici, on parcourt le tableau qui contient les commandes d'un client correspondant et les afficher.

5.4.5 Les expressions régulières :

Preg_match() : Compare une regex à une chaîne de caractères

```

$pattern_password = '#^(?=.*[a-z])(?=.*[A-Z])(?=.*[0-9])(?=.*\W){8,}$#';
$pattern_email = "/^[_a-z0-9-]+(\.[_a-z0-9-]+)*@[a-z0-9-]+(\.[a-z0-9-]{2,})$/";

```

```
if($row['num'] > 0)
    $erreur .= "Email déjà
utilisé<br>";
else
if(!preg_match($pattern_email, $email))
$erreur .=
"Email n'est pas valide<br>";
    if($password !=
$_POST['password2'])
        $erreur .=
"Mot de passe n'est pas identique<br>";
else
if(!preg_match($pattern_password, $password))
$erreur .= "Mot de passe n'est pas très
fort<br>";
```

5.5 Application Panier :

Le fichier functions_panier.php contient toutes les fonctions effectuées sur le panier
(l'ajout, la suppression, la modification...)

```

require 'config.php'; //changerQuantite function
    changerQuantite($idp,
        $idc,$quantite, $db)
    {

        $sql = "UPDATE panier SET quantite = :quantite WHERE idProduit
= :idp AND idClient = : idc";
        $stmt = $db->prepare($sql);
        $stmt->bindValue(':quantite', $quantite);
        $stmt->bindValue(':idp', $idp);
        $stmt->bindValue(':idc', $idc);
        if($stmt->execute()) return
true;
    }
    //Ajouter Produit dans Panier
function ajouterProduit($idp, $idc,
$quantity, $taille, $db)
{
    if(produitExiste($idp,
$idc, $db)) {
        $quantite = quantiteProduit($idp, $idc,
$db);
        if(changerQuantite($idp, $idc,
$quantite+$quantity, $db)) return true;
    } else
{
        $data = [
            'idProduit' => $idp,
            'idClient' => $idc,
            'taille'=>$taille,
            'quantite' =>$quantity,
];

```

```
        $sql = "INSERT INTO panier (idProduit, idClient,taille,
quantite) VALUES (:idProduit, :i dClient,:taille, :quantite)";
$stat= $db->prepare($sql);        if($stat-
>execute($data)) {                return true;
    }
} }
```

```
// Produit Existe dans panier
function produitExiste($idp, $idc, $db)
{
    $sql = "SELECT COUNT(id) AS num FROM panier WHERE idProduit =
:idp AND idClient = :idc";
    $stmt = $db->prepare($sql);
    $stmt->bindValue(':idp', $idp);
    $stmt->bindValue(':idc', $idc);
    $stmt->execute();
    $row = $stmt-
>fetch(PDO::FETCH_ASSOC);
    if($row['num'] > 0)          return
true;      else          return
false;
}

function quantiteProduit($idp, $idc, $db)
{
    $sql = "SELECT quantite FROM panier WHERE idProduit = :idp AND
idClient = :idc";
    $stmt = $db->prepare($sql);
    $stmt->bindValue(':idp', $idp);
    $stmt->bindValue(':idc', $idc);
    $stmt->execute();
    $row = $stmt-
>fetch(PDO::FETCH_ASSOC);
    return $row['quantite'];}
function supprimerProduit($idp,
$idc,
$db) {
```



```

    $sql = "DELETE FROM panier WHERE idProduit = :idp AND idClient
= :idc";
    $stat= $db->prepare($sql);
    $stat-
>bindValue(':idp', $idp);
$stat-
>bindValue(':idc',
$idc);
if($stat->execute()) {
return true;
}
}
// Nombre Produits dans
Panier function
nombreProduits($idc, $db)
{
    $sql = "SELECT COUNT(id) AS num FROM panier WHERE idClient
= :idc";    $stmt = $db->prepare($sql);
    $stmt->bindValue(':idc', $idc);
    $stmt->execute();
    $row = $stmt-
>fetch(PDO::FETCH_ASSOC);
return $row['num'];
}

```

```

// function
totalPrixPanier($idc,
$db)
{
    $total = 0;

    $rows=lesProduitsExistent($idc,
$db);    foreach ($rows as $row) {
        $row2 = infosProduit($row['idProduit'],
$db);
        $total += ($row2['prix'] *
$row['quantite']);
    }    return
$total; } function
lesProduitsExistent($idc,
$db)
{
    $sql = "SELECT * FROM panier WHERE idClient = :idc";
    $stmt = $db->prepare($sql);
    $stmt->bindValue(':idc', $idc);
    $stmt->execute();
    $produits = $stmt->
>fetchAll(PDO::FETCH_ASSOC);    return
$produits;
} Function
infosProduit($idp,
$db)
{
    $sql = "SELECT * FROM produits WHERE id = :idp";
    $stmt = $db->prepare($sql);

```

```
    $stmt->bindValue(':idp', $idp);
    $stmt->execute();
    $produit = $stmt-
>fetch(PDO::FETCH_ASSOC);
return $produit; }
function
supprimerPanier($idc,
$db)
{
    $sql = "DELETE FROM panier WHERE idClient = :idc";
    $stat= $db-
>prepare($sql);    $stat-
>bindValue(':idc',
$idc);    if($stat->execute())
return true;
}
```

5.5.1 Requête d'insertion d'un client :

```
$sql = "INSERT INTO clients (nom, prenom, email, password, telephone) VALUES (:nom, :prenom, :email, :password, :telephone)";
```

5.5.2 Requête d'insertion des produits achetés par un client :

```
$sql = "INSERT INTO commandes (idClient, date, prixTotal, adresse, ville, codal) VALUES (:idClient, :date, :prixTotal, :adresse, :ville, :codePostal)"; //  
$sql2 = "INSERT INTO details_commandes (idCommande, idProduit, taille, quantite) VALUES (:idCommande, :idProduit, :taille, :quantite)";
```

5.5.3 Formulaire pour laisser des commentaires :

Chaque client peut laisser des commentaires sur un produit.

```

<div class="comment-form-wrap pt-5">
    <h3 class="mb-5">Laissez un commentaire</h3>
    <?php if(!isset($_SESSION['client_id'])) { ?>
        <div class="row justify-content-center">
            <a class="btn btn-primary py-3 px-4" href="connexion.php">S'identifier</a>
        </div>
    <?php } else { ?>
        <form action="" method="POST" class="p-5 bg-light">
            <div class="form-group">
                <label for="commentaire">Commentaire :</label>
                <textarea class="form-control" cols="30" id="commentaire" name="commentaire" rows="5"></textarea>
            </div>
            <div class="form-group">
                <input class="btn py-3 px-4 btn-primary" name="ajoutercommentaire" type="submit" value="Poster le commentaire">
            </div>
        </form>
    <?php } ?> </div>

```

5.6 Backoffice:

Le fichier admin_function.php contient toutes les fonctions effectuées sur la gestion des produits(l'ajout, la suppression, la modification, List des catégories, nombre des messages)

```
<?php
require '../inc/config.php';
//récupérer tous les categorie des produits depuis la base de données
function categories($db){
    $query1 = $db-
>prepare("SELECT DISTINCT categorie FROM produits where quantite>0");
    $query1->execute();
    $categories = $query1->fetchAll();
    return $categories;
}
//Ajouter Produit
function AjouterProduit($libelle,$prix,$quantite,$image,$categorie,$type
,$description, $db)
{
    $sql = "INSERT INTO `produits`(`libelle`, `prix`, `quantite`, `image`
,`categorie`,`type`, `description`) VALUES (:libelle,:prix,:quantite,:im
age,:categorie,:type,:description)";
    $stat= $db->prepare($sql);
    $data=[
        'libelle' => $libelle,
        'prix' => $prix,
        'quantite' => $quantite,
        'image'=>$image,
        'categorie'=>$categorie,
```

```

        'type'=>$type,
        'description' => $description
    ];
    if($stat->execute($data)) {
        return true;
    }
}

//Modifier Produit
function ModifierProduit($idp,$libelle,$prix,$quantite,$categorie,$type,
$description, $db)
{
    $sql = "UPDATE `produits` SET `libelle`=:libelle,`prix`=:prix,`quantite`=:quantite,`categorie`=:categorie,`type`=:type,`description`=:description WHERE id=:idp";
    $stat= $db->prepare($sql);
    $data=[
        'libelle' => $libelle,
        'prix' => $prix,
        'quantite' => $quantite,
        'categorie'=>$categorie,
        'description' => $description,
        'type'=>$type,
        'idp'=>$idp
    ];
    if($stat->execute($data)) {
        return true;
    }
}

// supprimer produit
function SupprimerProduit($idp, $db)
{
    $sql = "DELETE FROM produits WHERE id = :idp";
    $stat= $db->prepare($sql);

```

```
$stat->bindValue(':idp', $idp);
if($stat->execute()) {
    $sql2 = "DELETE FROM panier WHERE idProduit = :idp";
    $stat2= $db->prepare($sql2);
    $stat2->bindValue(':idp', $idp);
    if($stat2->execute()) {
        return true;
    }
}
}

//récupérer le nombre de message
function nombreMessages($db){
    $sql = "SELECT COUNT(id) AS num FROM messages";
    $stmt = $db->prepare($sql);
    $stmt->execute();
    $row = $stmt->fetch(PDO::FETCH_ASSOC);
    return $row['num'];
}

?>
```


6 Conclusion :

Ce projet consiste à concevoir un site web dynamique qui permet de réaliser le commerce électronique des produits déjà utilisés.

Nous avons présenté les différentes étapes de la réalisation de notre projet en commençant par le contexte général de projet, puis ses fonctionnalités, et après la conception, et enfin la réalisation et la description des techniques et des fonctions utilisées.

Des améliorations pourraient aussi être apportées à ce site par exemple la possibilité de modifier les informations personnelles de client et l'ajout d'une photo de profil et l'adresse de chaque commande, de vérifier l'email, l'envoi d'un SMS au client indique que sa commande a été confirmé et développer un espace pour l'administrateur pour gérer le site. Concernant les obstacles que nous avons affrontés, on peut citer : la possibilité de changer la quantité d'un produit dans le panier, c'est pour cela que nous avons créer une fonction en JavaScript permettent de actualiser la page de panier une fois que la valeur de quantité a été changé par le client. Cette fonction envoie un ordre au PHP à travers l'url de la page pour effectuer les

modifications nécessaires dans la base de données et changer la quantité ainsi le prix total du produit et par conséquence de la commande. consulte ce rapport qui résume deux mois de travail rigoureux. Pour le moment le site e-commerce est presque terminé nous souhaitons qu'il trouvera les conditions nécessaires pour entrer en vigueur.