Rapport de projet de session

« Conception et réalisation d’un site web E-commerce sous nom : MySweater »

Réalisé par :

Khalid HAMDANI
Mariam ABALA

Encadré par :
M. Ahmed EL OUAFDI

Année universitaire 2019-2020

Contexte général

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 permettrai 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 « MySweater » pour la vente des chemises en ligne.

Fonctionnalités

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

-Contacter l'administrateur.

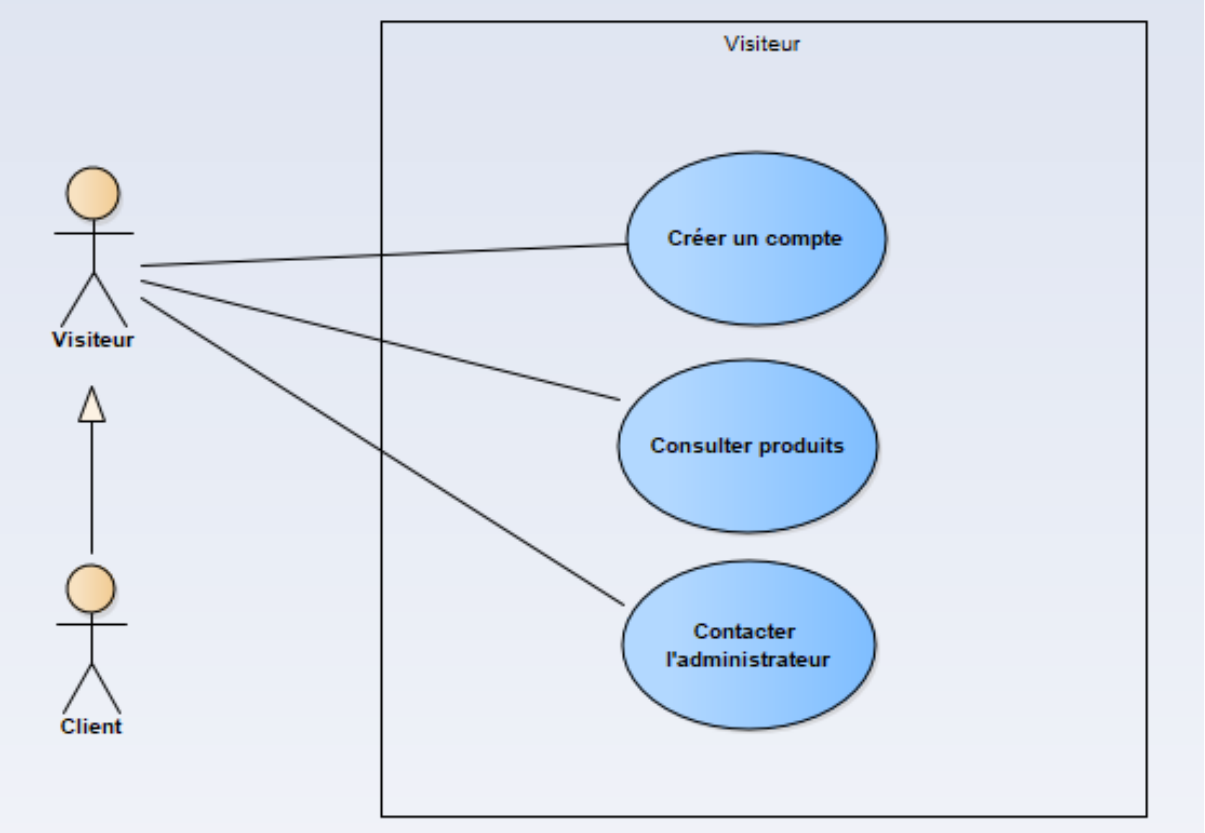
-Récupérer les informations de la connexion à la base de données depuis un fichier txt.

La conception

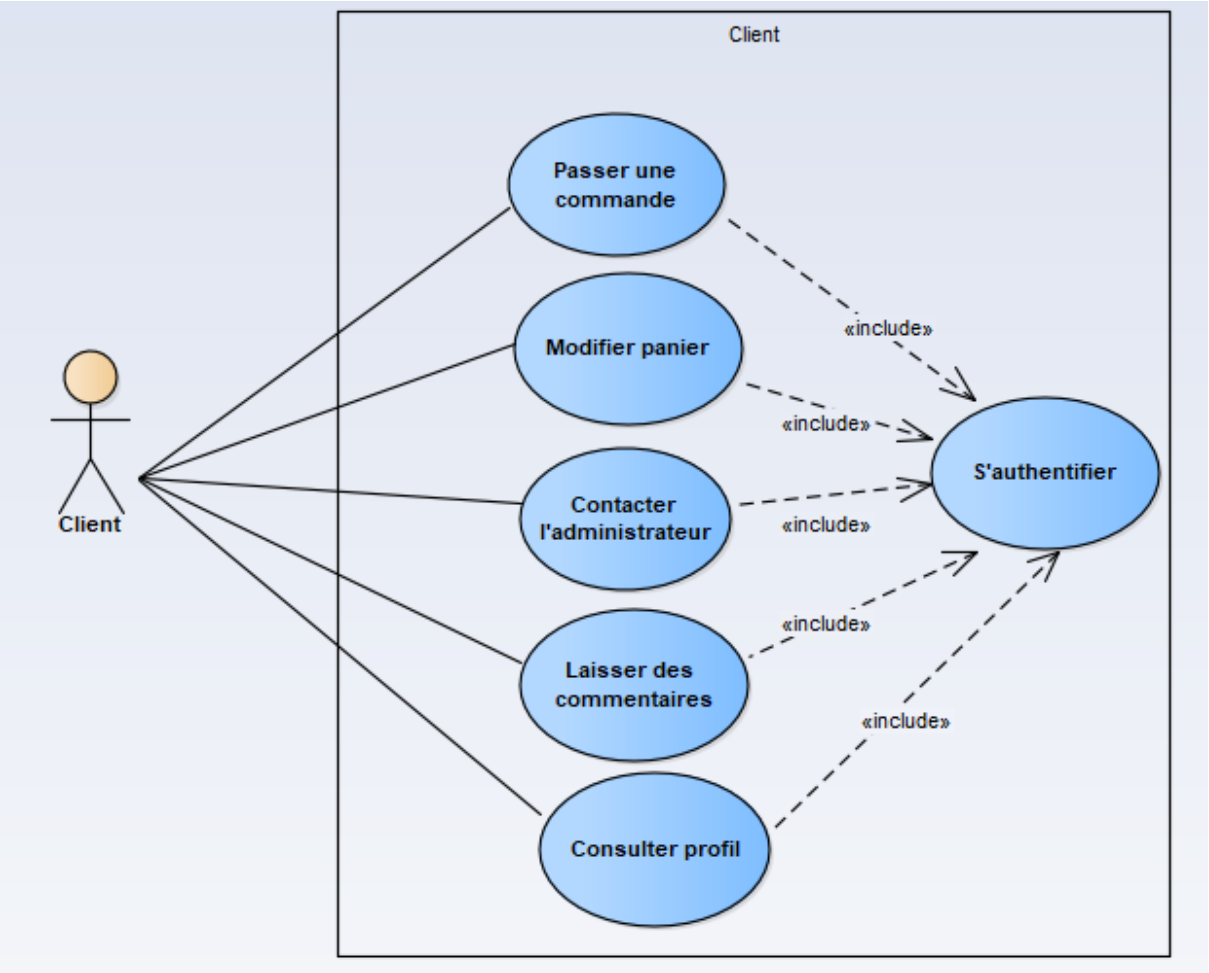
1- Diagramme de cas d'utilisation :

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.

Le visiteur : c'est un individu qui est en train de chercher un produit pour l'acheter ou pour avoir une idée sur les modèles et les prix. C'est un utilisateur inconnu donc il n'est pas encore un client.



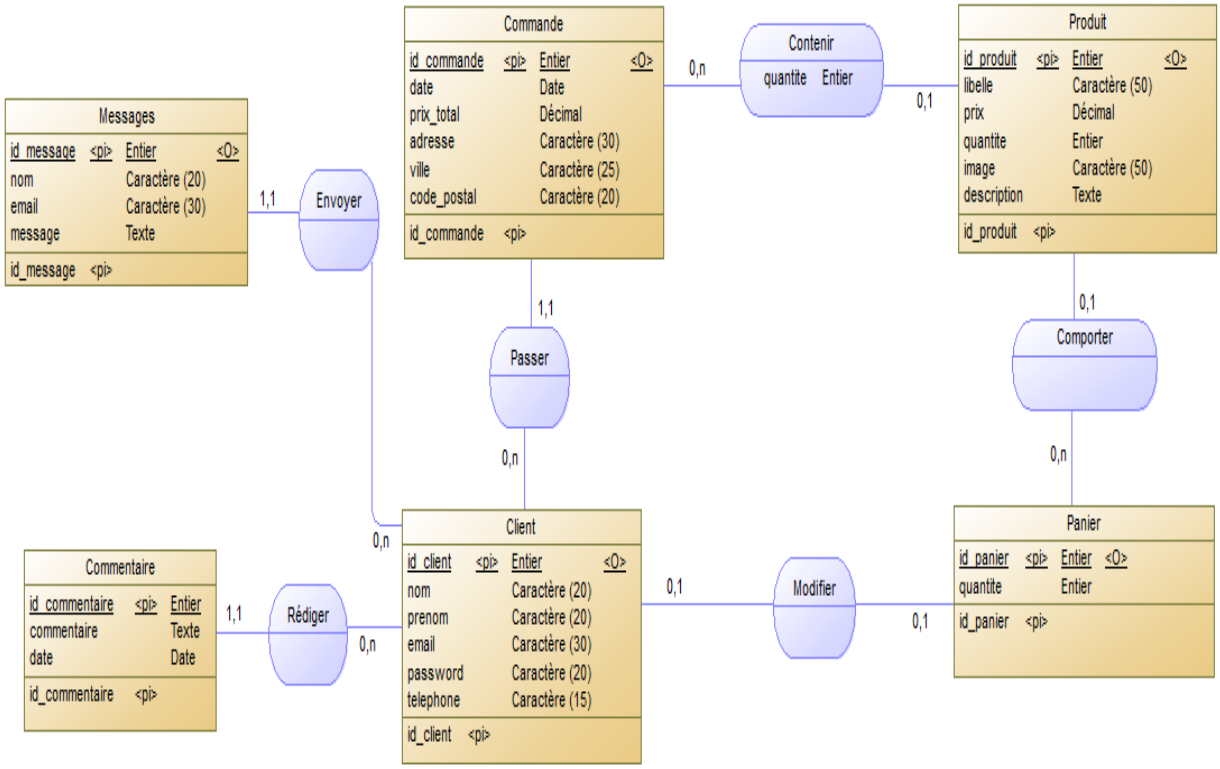
Le client : Cet acteur est un visiteur ayant déjà créer un compte sur notre site, il peut donc suivre le processus d'achat des produits en toute sécurité sachant que notre système doit être l'unique responsable de la confidentialité des données personnelles de ses clients.



2- Le Modèle Conceptuel de Données (MCD) :

Un Modèle Conceptuel de Données est la formalisation de la structure et de la signification des informations décrivant des objets et des associations perçus d'intérêt dans le domaine étudié, en faisant abstraction aux solutions et aux contraintes techniques et informatiques d'implantation en base de données.

Un MCD est exprimé en entité-relation Merise qui comporte les concepts basiques suivants : Entité, relation, cardinalités, propriétés, identifiant.


Commande
id_commande <pi> Entier <O>
date Date
prix_total Décimal
adresse Caractère (30)
ville Caractère (25)
code_postal Caractère (20)
id_commande <pi>
Contenir
quantite Entier
0,n
0,1
Produit
id_produit <pi> Entier <O>
libelle Caractère (50)
prix Décimal
quantite Entier
image Caractère (50)
description Texte
id_produit <pi>
Messages
id_message <pi> Entier <O>
nom Caractère (20)
email Caractère (30)
message Texte
id_message <pi>
1,1
Envoyer
1,1
Passer
0,n
0,1
Comporter
0,n
0,n
Client
id_client <pi> Entier <O>
nom Caractère (20)
prenom Caractère (20)
email Caractère (30)
password Caractère (20)
telephone Caractère (15)
id_client <pi>
Commentaire
id_commentaire <pi> Entier
commentaire Texte
date Date
id_commentaire <pi>
1,1
Rédiger
0,n
0,1
Modifier
0,1
Panier
id_panier <pi> Entier <O>
quantite Entier
id_panier <pi>

Réalisation

1- Les techniques et les fonctions utilisées :

1.1. 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.

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

```
s $_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
<?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
<?php
  include('config.php');
  session_unset();
  session_destroy();
  header("location: ../index");
?>
```

1.2. Téléchargement d'un fichier :

Le téléchargement d'un fichier (notamment ce rapport) ça fait à l'aide de la balise HTML suivante :

```
<a class="nav-link" href="Rapport-MySweater.pdf" download>Rapport</a>
```

1.3. 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);
```

1.4. 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_N
AME.';charset=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();
}
```

1.5. La manipulation des fichiers :

L'utilisation de fichiers en PHP va nous permettre de stocker de façon définitive des informations.

Ici, on utilise le fichier infos.txt pour récupérer les données associées à la connexion à notre base de données (le nom de la base données, le mot de passe, le login, le nom d'hôte).

```php
<?php
//récupérer les données depuis le fichier infos.txt
$fichier = __DIR__ . "/infos.txt";
//Vérifier si le fichier existe ou pas
if (!file_exists($fichier)) {
    die("Fichier infos.txt n'existe pas!");
}
//récupérer les infos sous forme des lignes
$infos = file($fichier);
//Supprimer les espaces
$infos[0] = preg_replace('/\s+/', '', $infos[0]);
$infos[1] = preg_replace('/\s+/', '', $infos[1]);
$infos[2] = preg_replace('/\s+/', '', $infos[2]);
$infos[3] = preg_replace('/\s+/', '', $infos[3]);

$DATABASE_NAME = $infos[0];
$DATABASE_PASSWORD = $infos[1];
$DATABASE_USER = $infos[2];
$DATABASE_HOST = $infos[3];

try {
    $db = new PDO('mysql:host='.$DATABASE_HOST.';dbname='.$DATABASE_NAM
E.';charset=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();
}
?>
```

1.6. 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['telephone'];

            header('Location: produits');
        } else{
            $erreur .= "Mot de passe est incorrect<br>";
        }
    }
```

1.7. 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 parcoure le tableau qui contient les commandes d'un client correspondant et les afficher.

1.8. Les expressions régulières et les fonctions PCRE :

preg_replace() : Recherche et remplace.

```
//Supprimer les espaces
$infos[0] = preg_replace('/\s+/', '', $infos[0]);
$infos[1] = preg_replace('/\s+/', '', $infos[1]);
$infos[2] = preg_replace('/\s+/', '', $infos[2]);
$infos[3] = preg_replace('/\s+/', '', $infos[3]);
```

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

```
        if($row['num'] > 0)
            $erreur .= "Email déjà ultilisé<br>";
        else if(!preg_match($pattern2, $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))
            $erreur .= "Mot de passe n'est pas très fort<br>";
```

1.9. Application Panier :

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

```
<?php
require 'config.php';
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;
}
```

```php
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;
   }
}
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'];
}
```

```php
function totalPrixPanier($idc, $db)
{
   $total = 0;
   $sql = "SELECT * FROM panier WHERE idClient = :idc";
   $stmt = $db->prepare($sql);
   $stmt->bindValue(':idc', $idc);
   $stmt->execute();
   $rows = $stmt->fetchAll(PDO::FETCH_ASSOC);

   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;
}
?>
```

1.10. Requête d'insertion d'un client :

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

1.11. Requête d'insertion des produits achetés par un client :

```
$sql = "INSERT INTO commandes
(idClient, date, prixTotal, adresse, ville, codePostal)
VALUES (:idClient, :date, :prixTotal, :adresse, :ville, :codePostal)";
```

1.12. Formulaire pour laisser des commentaires :

Chaque client peut laisser des commentaires sur un produit.

```
<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">S'identifer</a>
 </div>
 <?php } else { ?>
 <form action="" method="POST" class="p-5 bg-light">
 <div class="form-group">
 <label for="commentaire">Commentaire :</label>
 <textarea class="formcontrol" cols="30" id="commentaire" name="comment
aire" rows="5">
</textarea>
 </div>
 <div class="form-group">
 <input class="btn py-3 px-4 btn-primary" name="ajouter-
commentaire" type="submit" value="Poster le commentaire">
   </div>
  </form>
 <?php } ?>
```

Conclusion

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

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.