

Rapport



Réaliser par: Chabchoub Salim et Elabridi Mustapha

Manuelle d'utilisation:

La première page de notre site c'est la page qui présente les produits les plus vendus pour accéder à cette page il suffit de saisir ce lien : <http://localhost/IsiShop/Controller/top5.php>



Bienvenue sur le meilleur site de E-Commerce

Voici nos articles les plus vendus :



Après pour naviguer sur le site vous pouvez utiliser la barre de navigation. N'oubliez pas d'exécuter le dump sql web4shop.sql qui a été envoyé avec le code car la base de données a été modifiée.

Répartition du Travail:

Le Travail a été réparti d'une manière équitable (50/50).Voici les principales tâche de chacun :

- Salim : page d'accueil qui présente les produits les plus vendus, page de présentation des produits, la page de chaque produit, le panier et la page d'admin
- Mustapha : page de paiement,page d'ajout d'adresse, la page de connexion et la page de création de compte.

On s'est trop aidé l'un et l'autre surtout quand on rencontre un problème.

Difficultés Rencontrées:

L'une des difficultés les plus majeurs qu'on a rencontré c'était le partage de code sur github au début on a perdu notre code plusieurs fois mais après on a commencé à maîtriser le git et à comprendre comment ça marche.

En plus on a eu un peu du mal au début pour accéder à la base de données à partir de php. On a aussi trouvé quelques difficultés pour mettre les données de session dans la base de données au moment de la connexion donc il fallait alterner entre les deux.

Une autre difficulté réside dans la partie connexion : l'affichage des messages d'erreur. Sans accès à JavaScript (introuvable dans le cahier de charge donné), l'affichage devient plus difficile.

Solutions mises en oeuvre:

Pour nous faciliter un peu la tâche au niveau du travail on a préféré de faire quelques modification au niveau de la base de données : on a ajouté des triggers au niveau de la table orders et orderitems pour que dès que l'utilisateur ajoute un produit au panier le total va t'être mis à jour automatiquement :

```
CREATE TRIGGER update_order_total AFTER INSERT ON orderitems
FOR EACH ROW
BEGIN DECLARE order_total FLOAT;
  -- Calculer le nouveau total pour la commande
  SELECT SUM(oi.quantity * p.price) INTO order_total FROM orderitems oi JOIN
  products p ON oi.product_id = p.id WHERE oi.order_id = NEW.order_id;
  -- Mettre à jour le total dans la table orders
  UPDATE orders SET total = order_total WHERE id = NEW.order_id;
END;
La même chose si l'utilisateur supprime un produit du panier le montant de ce
produit va t être déduit du total :
CREATE TRIGGER
update_order_total_after_delete AFTER DELETE ON orderitems FOR EACH ROW
BEGIN
DECLARE order_total FLOAT;
  -- Calculer le nouveau total pour la commande
  SELECT SUM(oi.quantity * p.price) INTO order_total FROM orderitems oi JOIN
  products p ON oi.product_id = p.id WHERE oi.order_id = OLD.order_id;
  -- Mettre à jour le total dans la table orders
  UPDATE orders SET total = order_total WHERE id = OLD.order_id;
END;
```

En outre, on a mis la plupart des informations qui nous ont semblé utiles dans la session même si l'utilisateur est connecté ça nous a bien aidé au niveau de la récupération des données surtout au moment de la connexion.

Pour ce qui est de l'affichage de l'erreur, on a résolu cela simplement en mettant la page en twig, puis, envoyant un tableau des différentes erreurs pour qu'ils s'affichent ensemble.

Pour ce qui est des tests pour la validation de paiement, il faut vraiment entrer ses informations et réellement payer pour que ça s'active. Sinon, vous ne pourrez pas faire.

Concernant le paiement par chèque, on n'a pas réalisé la partie de validation de paiement puisque cela doit être validé par une vraie banque.