

Compte-rendu projet ISIWEB4SHOP :

I) Répartition des rôles dans l'équipe

- Aurélien :
 - Mise en place de l'architecture MVC (routeur, contrôleurs, modèles)
 - Footer
 - Gestion du panier : ajout des fonctionnalités, de la vue, ajout des commandes à la base de données, calcul automatique du total avec Javascript, style de la page de panier
 - Répartition des tâches au sein de l'équipe
- Luc :
 - Mise en place des accès base de données (modele.php)
 - Mise en place des modèles, vues et contrôleurs gérants la connexion/création de compte
 - Mise en place des modèles, vues et contrôleurs gérants la page admin (gérer stocks/ valider commande)
 - Page paiement
- Nathan :
 - Pages produits, catégories, accueil
 - Frontend
 - Gestion responsive avec Bootstrap
 - Page adresse
 - Gestion du paiement et de la facture

II) Difficultés rencontrées et solutions mises en œuvre

Au début du projet, nous avons du mal à comprendre et appliquer l'architecture MVC à notre projet. En effet, l'enchaînement des différents appels entre le routeur, le contrôleur et la vue n'était pas évident à assimiler. Cela nous a demandé un certain temps d'adaptation. Nous nous sommes également aidés du TP sur l'architecture MVC.

Pour la mise en place du panier, nous n'avons pas réalisé le nombre de fonctionnalités à implémenter. En effet, il fallait par exemple gérer la possibilité d'ajouter deux fois le même produit au panier, ne pas autoriser l'utilisateur à dépasser la quantité maximale du stock du produit, gérer les changements de quantités dans le panier à l'aide de Javascript.

Nous avons dû réfléchir à la manière de gérer les personnes qui commandaient sans être connectées.

Nous avons choisi pour ces utilisateurs de ne pas directement stocker leur commande sur la base de données. En effet, s'ils abandonnent leur commande, en quittant le site par exemple, il leur était impossible de la retrouver, et cette commande serait donc restée "en suspens" dans la base de données. Les commandes abandonnées auraient donc pollué la base de données.

Nous avons ainsi choisi de stocker dans un premier temps leur commande dans les variables de session, puis lorsque le paiement est effectué, nous stockons leur commande dans la base de données.