

Compte Rendu : Projet Web

Nous nous sommes donné beaucoup de mal au départ pour débiter ce projet car nous ne savions pas quel thème choisir : tantôt on voulait présenter des produits classiques, tantôt des produits originaux (trafic d'organes, vente de tank). Nous avons déjà commencé à mettre en place la présentation du site, mais en remplaçant les noms des produits par ("category 1", "category 2" etc...). Pendant ce temps, on discutait aussi sur le système de gestion de la base de donnée, et nous commençâmes par créer la table des clients (`customers`). Après de longues réflexions sur ce sujet, nous avons décidé de partir sur une initiative plus professionnelle. On a donc choisit de vendre des produits informatique tout en préservant le nom du lieu ou nous avons débuté ce projet « Laptop de Marne la Vallée © ».

Nous nous sommes servis du logiciel de versions décentralisés Git et de l'hébergement et des services en ligne gratuit de GitHub afin de gérer le projet en ligne afin de récupérer toute nouvelles modifications et de travailler en parallèle. Ce service est excellent puisqu'il nous permet aussi de commenter à chaque fois qu'on passe à une nouvelle modification, on peut aussi voir les graphes concernant nos heures de travail, organiser les tâches à effectuer, etc...

lien: <https://github.com/alisterwan/website>

Après avoir créer la présentation de l'entête et le pied de la page que nous avons inclus dans chaque nouvelle page créée, nous avons débuté ce projet par les formulaires d'inscription : quelques difficultés avec les requêtes SQL au début, mais avec de la pratique, le problème a vite été résolu. Le véritable problème se posait au niveau de l'incrémentation automatique des identifiants dans la table clients (`customers`).

Après avoir créé un formulaire simple, sans établir des conditions préalables (par exemple : la syntaxe des emails), nous avons procédé à la création de la procédure d'authentification pour bien vérifier que le client s'était bien enregistré dans la base de donnée. Nous avons créé de nombreux tests pour vérifier que les valeurs entrées dans le formulaire sont valides et en adéquations avec la base de données.

Une fois cette première étape terminée, nous sommes passés à l'espace back-office en créant un compte pour l'administrateur. Dans ce compte, nous pouvons faire la gestion de tout ce qui est client, produit et commande. Nous avons commencé par créer des formulaires d'inscription pour les produits. Nous avons commencé à remplir les produits dans la base de donnée en y associant une image correspondante, re-travaillé au niveau des contours sous le logiciel GIMP.

Nous avons aussi créer une barre de navigation où nous avons inclus les liens vers les produits ajoutés facilitant l'internaute à trouver son produit.

Ensuite, après avoir établit ces données, nous avons réfléchi sur l'affichage des produits : on a donc décider d'afficher par les catégories pour certains, et par marque pour d'autres en

y associant les images correspondantes renommées comme le nom du modèle (exemple : `macbookair.png`).

Nous avons commencé par créer la page d'affichage du produit correspondant en réfléchissant comment se référer au produit. Après un long moment de réflexion, nous avons trouvé une solution possible : récupérer les libellés des produits par la méthode GET qui part directement dans l'URL de la page. Cette méthode nous permet de partager le lien du produit concerné. Nous avons décidé de générer les pages d'affichage pour tous les autres type de produits.

Troisième étape : Création du panier.

Pour le chariot, nous récupérons les actions par méthode GET ou POST pour ajouter, supprimer et modifier la quantité de chaque produits.

Ajouter un produit met sa quantité 1 s'il n'existe pas dans le chariot. Décroître la quantité d'un produit jusqu'à zéro le supprimera du panier pour éviter de créer des valeurs négatives lors d'un rafraîchissement.

Ensuite, le panier est parcourue pour récupérer les valeurs et incrémenter le prix total. Si le prix est nul, par exemple si le produit n'existe plus ou la quantité est nul, le produit sera supprimé du panier afin d'éviter l'ajout de produit inexistant.

Enfin, si le prix total existe, c'est à dire si le panier contient des produits valides, le tableau sera affiché. Sinon, un message indiquant que le panier est vide y sera affiché.

Nous avons créer une table commande (`orders`) en créant un formulaire demandant le numéro de carte bancaire ou nous avons enregistré toutes les commandes passées dans le panier, avec une colonne en plus, `time`, qui stocké la date d'achat.

Ensuite nous sommes revenus sur les formulaires d'inscription, ou nous avons mis certaines contraintes sur les champs (e-mail, champs non vide, vérification des mots de passe, fonction qui envoie un email de confirmation), et nous avons utilisé une fonction de cryptage pour les mot de passe.

Nous avons aussi modifié l'espace du back-office en y ajoutant la gestion des stock des produits (modifier la quantité ainsi que les prix unitaire), gestion des clients (modifier, supprimer), ainsi que la liste des commandes passées.

Nous sommes passés aux triggers en PostgreSQL ainsi que leurs fonctions associées: création d'un trigger & fonction qui génère le numéro des clients lorsqu'on en rajoute un nouveau.