

Projet Web 2

L'application doit être réalisée en ayant une architecture Modèle Vue Contrôleur sur le framework PHP Silex avec comme moteur de Template « twig ».

Utiliser le fichier SQL joint et le modifier à votre bon vouloir. Indiquer les changements.

Le projet doit utiliser un maximum de briques logicielles de *Symfony* :

* Une version minimum (point en gras dans la feuille d'évaluation) doit utiliser le routeur Silex.

* Il est possible de réaliser une deuxième version avec le framework *Symfony*.

* Lors de l'évaluation les points suivants seront pris en compte : application REST, sécurité... , utilisation de requêtes paramétrées, voir de l'orm.

* Le « design » dans ce projet est un plus.

Objectif : Réaliser une application pour commander des produits (ne pas prendre les fruits et légumes dont le prix est au poids et/ou à l'unité, et pas forcément connu ce qui complique l'application).

La base de données doit comprendre au minimum 4 tables (produits, commandes, users, paniers (ou produits_commande)).

Pour chaque produit qui compose une commande (une ligne du panier), enregistrer :

- * Le nombre de produit associé (ainsi que le prix lors de l'achat car le vendeur peut faire évoluer le prix d'un article dans le temps).

- * L'identifiant du client (qui est dans la session)

- * La date de la commande

Par simplicité on n'utilise pas de variable de sessions pour sauvegarder le panier. (Mais il est possible de choisir cette solution).

Commandes (front Office) :

Un client peut remplir un panier en étant connecté. Il peut alors valider sa commande durant une petite période (à définir) . (on suppose que l'adresse de livraison est l'adresse du client).

Lorsque le client ajoute un article dans le panier, 2 cas peuvent se présenter :

- * Si l'article n'existe pas dans le panier, il faut alors créer une nouvelle ligne avec les références du produit dans le panier.

- * Si l'article est déjà présent dans le panier, il faudra mettre à jour la quantité de ce produit

- * Prévoir un moyen pour soit modifier la quantité (champ d'entrée), soit décrémenter la quantité (bouton + / -), soit supprimer l'article du panier.

La validation du panier provoque la création d'une commande, le panier n'est plus affiché, mais le client pourra consulter ses commandes ainsi que leur état.

Dans la théorie, le client doit s'acquitter du paiement avant de valider la commande (par paypal, banque ...), cette partie n'est pas à gérer.

Gestion des Commandes (back Office) :

Après connexion, le vendeur peut mettre à jour toutes les informations des produits d'une semaine disponibles à la vente (ajout/modification/suppression).

Préparation des commandes

Il peut naviguer facilement de commande en commande et passer l'état d'une commande de « À préparer » à « Expédier » (voir Préparer et éventuellement Livrer)

Il peut facilement signaler qu'un produit n'est plus disponible.

La structure minimum est évaluée chaque semaine avec des objectifs (voir feuille de suivi)

Lors de la dernière séance, plusieurs éléments qui améliorent l'application seront validés

Coté vendeur (front Office)

* lors d'une commande le stock est mis à jour (option)

- le vendeur peut voir à l'affichage : stock réel, stock après expédition des commandes en cours.

- le vendeur peut mettre à jour le stock (arrivage d'articles, articles défectueux ...), une modification du MR pourrait permettre de garder une trace de ces modifications

* comme sur certains sites de vente : le stock restant est affiché si il y a moins de 10 articles sinon l'article est indiqué comme « disponible », un client ne peut pas commander plus d'articles que dans le stock.

* Chargement d'image : le vendeur peut faire un « upload » des images de ces produits

Coté client (back Office)

Un client peut afficher les articles par type

Un client peut ajouter des commentaires : si il a commandé un produit, il peut laisser un commentaire sur ce produit

* utiliser de préférence un editeur wysiwyg pour les commentaires (<http://www.tinymce.com/> par exemple), lors de la création des commentaires

* Un utilisateur (client) peut changer de nom, email, mot de passe, prenom

* Mettre un système de captcha lors de la création des comptes

* crypter les mots de passe et attention à la sécurité (middleware)

Sécurité

* Middleware

*Token

Interface

* pour ce projet, l'ergonomie de l'interface est prise en compte ainsi que l'utilisabilité du site. Une bonne utilisation du « framework bootstrap » est un plus afin d'obtenir un site « responsive ».

Il est possible de faire une version avec « symfony » avec l'accord de l'enseignant car c'est plus compliqué et ça demande un investissement personnel plus important.

Modèle logique de données :

