

Projet Ventigisbi

I-Consigne à respecter :

-Crée une application pour permettre d'enregistrer les commandes de produits proposés par le visiteur, Elle doit être suivi et daté sur les opérations menées.

-Capacité du visiteur :

1-Le visiteur pourra commander plusieurs produits avec une quantité pour chaque.

2-L'ajout des produits sera géré grâce à un panier.

3- Lors de la validation, il devra avoir un récapitulatif du coût total, du délai et un mail de confirmation.

4-Il pourra lors d'une connexion ultérieure au site avoir un récapitulatif de ses commandes. On pourra éventuellement gérer le stock de chaque produit.

5-Le visiteur pourra voir les commandes des médecins dont il a la charge, les modifier ou les supprimer.

-Il aura une application web pour les médecins ainsi que pour les visiteurs.

-Les Récapitulatifs des commandes seront accessibles seulement par le visiteur en question.

-Un procédé d'authentification à mettre en place pour l'accès des contenus.

II-Contraintes :

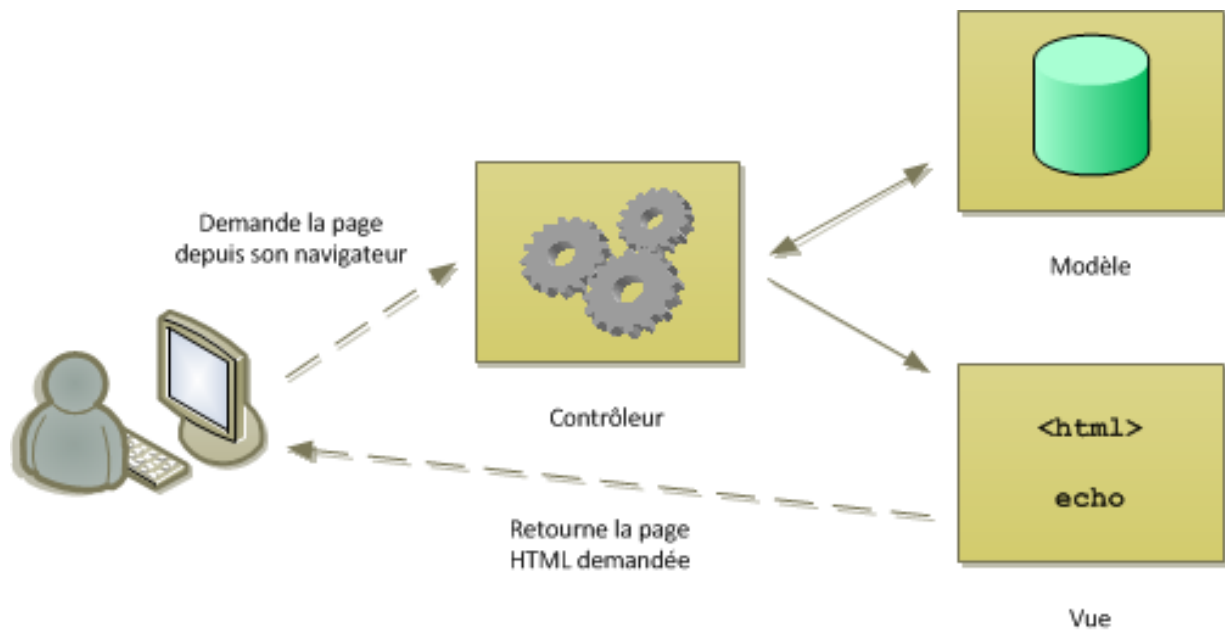
1)-L'architecture devra être en MVC.

MVC (Model view Controller) : Une architecture MVC, permet de bien organiser son code source. Le but de MVC est justement de séparer la logique du code en trois parties que l'on retrouve dans des fichiers distincts

- **Modèle** : cette partie gère les *données* de votre site. Son rôle est d'aller récupérer les informations « brutes » dans la base de données, de les organiser et de les assembler pour qu'elles puissent ensuite être traitées par le contrôleur. On y trouve donc les requêtes SQL.

Parfois, les données ne sont pas stockées dans une base de données. C'est plus rare, mais on peut être amené à aller chercher des données dans des fichiers. Dans ce cas, le rôle du modèle est de faire les opérations d'ouverture, de lecture et d'écriture de fichiers (fonctions fopen, fgets, etc.).

- **Vue** : cette partie se concentre sur l'*affichage*. Elle ne fait presque aucun calcul et se contente de récupérer des variables pour savoir ce qu'elle doit afficher. On y trouve essentiellement du code HTML mais aussi quelques boucles et conditions PHP très simples, pour afficher par exemple la liste des messages des forums.
- **Contrôleur** : cette partie gère la logique du code qui prend des *décisions*. C'est en quelque sorte l'intermédiaire entre le modèle et la vue : le contrôleur va demander au modèle les données, les analyser, prendre des décisions et renvoyer le texte à afficher à la vue. Le contrôleur contient exclusivement du PHP. C'est notamment lui qui détermine si le visiteur a le droit de voir la page ou non (gestion des droits d'accès).



-Codage :

2) Les connexions aux bases de données se feront en PHP avec l'objet PDO.

PDO est l'un des nombreux moyens pour se connecter à une base de données avec PHP, dans le but d'utiliser des transactions.

Fonctionnalité :

- La première chose à faire est de se connecter à la base de données
- Il faut comme avec le pilote MySQL habituel fournir les informations suivantes :
- L'adresse de la base de données
- Le nom de la base de données
- Le nom d'utilisateur
- Et bien sûr le mot de passe pour accéder à la base de données
- Il ne reste plus qu'à créer un objet PDO pour interagir avec MySQL.

Les exceptions :

Un des avantages de PDO est de pouvoir utiliser les exceptions. Cela va nous permettre de lancer une transaction et selon le résultat (échec ou réussite de la transaction) agir en conséquence :

En cas de réussite : continuer la suite du script (confirmation, validation, etc).

En cas d'échec : une erreur ayant été détectée, il est alors possible de la récupérer (dans un log par exemple) pour ensuite la traiter. Le plus souvent les erreurs détectées sont des problèmes de clefs étrangères ou tout simplement des erreurs de codage. En ce sens les exceptions sont aussi très utiles pour déboguer un système sans tout bousiller lors des tests.

Transactions complète :

- La première étape consiste à initialiser la transaction avec la méthode PDO
- Exécuter simplement les requêtes SQL que l'on désire
- Enfin on applique toutes ces requêtes avec la méthode commit

Sources :

<https://openclassrooms.com/courses/concevez-votre-site-web-avec-php-et-mysql/lire-des-donnees-2>

<https://openclassrooms.com/courses/les-transactions-avec-mysql-et-pdo>

3)-Les pages Web se feront en HTML 5, ainsi pour la présentation en CSS, et du javascript externe.

4)-Reprendre la base de données de **MEDIBASE**.

-Documentation :

-Utilisation de phpdocumentor pour réaliser

utilisation de phpdocumentor:

<http://blog.lazycloud.net/fr/phpdocumentor-commenter-pour-mieux-documenter/>

<http://chez-syl.fr/2012/11/documentation-php-automatique-avec-phpdocumentor-sous-windows/>