Principes et architectures

Table des matières

I. Livrables

1. Sources

Toutes les sources du site web se trouvent dans le dossier src.

2. Documents

Dans le dossier docs se trouve le sujet du projet.

II. Côté serveur

3. MVC

J'ai décidé de mettre en place côté serveur une architecture MVC.

Le fichier index.php permet de faire le routage vers la méthode du contrôleur désiré en se basant sur la variables serveur PATH_INFO et en fonction de la méthode http utilisée. Les différentes routes sont décrites dans la variable $routes.

Les modèles se trouvent dans le dossier models, les vues dans le dossier templates et les contrôleurs dans le dossier controllers.

A. Modèles

Les modèles servent uniquement à représenter et désérialiser les données. Toutes les actions de récupération et de sauvegarde se font depuis la couche data.

B. Contrôleurs

Les contrôleurs héritent tous de la classe BaseController qui fournit la méthode render, permettant d'afficher une vue. Ils permettront également de gérer les appels via ajax, en proposant des méthodes issues de l'architecture REST (excepté la méthode index, qui va afficher la vue liée à la page affichée. Dans un souci de simplification, les méthodes REST ont été directement rattachées au contrôleur de chaque page).

C. Vues

Les vues sont servies depuis les contrôleurs avec la fonction render et contiennent du php pour effectuer le templating, mais aucune action métier. La fonction render va en fait servir la vue de base : base.php, en y incluant la vue dont le nom est passé en paramètre à render.

4. Data

Les données de l'application sont extraites de fichiers json et le panier est stocké en session.

J'ai déclaré deux singletons pour gérer la base de données (Db.mock.php) et le panier (ShoppingCart.mock.php).

La couche data fournit aussi des méthodes de simulation d'un ORM (qui n'a pas été implémenté, car il dépassait le cadre du projet).

III. Côté client

1. Styles, scripts et images

Ils se trouvent dans le dossier assets. Les styles et scripts sont ajoutés directement via des balises html dans base.php.

A. Styles

Les styles sont divisés dans deux fichiers :

- styles.css : pour gérer les styles globaux de l'application.
- colors.css : pour définir et gérer les couleurs présentes en base. Elles sont séparées des styles car elles sont fortement liées aux données.

B. Scripts

Un script correspondant à chaque page est ajouté automatiquement à base.php.
Les scripts communs à plusieurs pages sont placés dans common.js.

Un découpage des fichiers javascript a également été mis en place. Les contrôleurs permettent de faire des appels ajax vers le serveur et les fichiers de vue vont modifier l'interface. Lors des appels ajax, des debounces ont été mis en place, afin de n'envoyer qu'une seule requête par action (par exemple, lorsque l'on appuie incrémente successivement la quantité d'un produit dans le panier, seule la dernière pression envoie la requête).

La vue va également faire transiter les évènements de la vue vers le contrôleur.

C. Images

Dans le dossier product, se trouve une image au format jpeg par produit, nommée par l'identifiant de ce dernier.

2. Gestion des bibliothèques

Côté client, un gestionnaire de paquets a été mis en place : Bower, pour gérer les dépendances du projet côté client. Dans un souci de simplicité, les bibliothèques sont intégrées directement au zip.