

Travail pratique : Vitrine commerciale (partie 2)

Énoncé :

Dans ce travail vous aurez à produire un site de commerce électronique simplifié. Précisément, vous devrez concevoir la partie catalogue et le panier d'achat d'un site de commerce électronique. Le site final comprendra un catalogue qui permet à des clients de faire des achats (sans paiement). Afin de faciliter le développement, la gestion des usagers est simplifiée.

Le travail est divisé en 3 parties.

Partie 2 : Affichage et chargement dynamique du catalogue

La seconde partie consiste faire le chargement des produits du catalogue de manière dynamique en utilisant des requêtes asynchrones (AJAX). Selon les paramètres d'affichage créés dans la partie 1, vous devez faire la requête AJAX pour charger les produits. Le code côté serveur est fourni, vous n'avez qu'à faire les requêtes pour recevoir les données. Les données sont retournées en format JSON.

De plus, la gestion des produits du catalogue se fera en utilisant le module Pattern en JavaScript. Vous aurez donc à créer des méthodes gérer l'affichage du catalogue et les options (structure au choix).

Le catalogue devra comprendre les fonctionnalités suivantes :

- Options d'affichage en vignette ou en liste
- Gestion de l'affichage en diverses pages (12, 24, 36 produits par page ou tous)

Critères d'évaluation

- Maîtrise de la technique de programmation AJAX (35 %)
- Maîtrise de la programmation orientée objet en JavaScript (35 %)
- Fonctionnement adéquat du site Web dynamique (15 %)
- Qualité du code et documentation adéquate (15 %)

Remise :

- Remise sur Léa, le **15 décembre 2013** avant la fin de la journée.

Cas d'utilisation :

1- Nom : CatalogueProduit
Pré-conditions : Aucune
Description : Au lancement du site web l'utilisateur visualise une page qui présente un catalogue de produits. La page comprend : un bandeau d'entête avec une navigation simple, le catalogue de produit (liste des produits) et un pied de page contenant les informations de droits d'auteur.
Exception: Aucune
Post-conditions : Le catalogue est affiché selon le cas CatalogueProduit_afficherPage .

2- Nom : CatalogueProduit_afficherPage
Pré-conditions : <ul style="list-style-type: none">• Une page ou un nombre d'article à afficher a été sélectionné
Description : Le catalogue envoie une requête au serveur afin de récupérer les produits demandés. Un indicateur visuel permet à l'utilisateur de voir que la requête progresse. Une fois les résultats reçus (format json), le catalogue affiche les produits selon : <ul style="list-style-type: none">• la page sélectionnée (CatalogueProduit_changePage)• le mode d'affichage (CatalogueProduit_changeModeAffichage)• le nombre de produit par page (CatalogueProduit_changeNombreProduit) Par défaut : <ul style="list-style-type: none">• Page 1• 12 produits/page• Affichage en mode vignette Le catalogue affiche le gestionnaire de page en tenant compte du nombre de page total et de la page active. Si le nombre de produit total est inconnu, le catalogue envoie une requête au serveur afin de récupérer le total. <ul style="list-style-type: none">• Le bouton précédent ne doit pas apparaître sur la première page• Le bouton suivant ne doit pas apparaître sur la dernière page• Un bouton permet d'aller à chaque page directement Le catalogue affiche les contrôles permettant de changer : <ul style="list-style-type: none">• le mode d'affichage (CatalogueProduit_changeModeAffichage)• le nombre de produit par page (CatalogueProduit_changeNombreProduit)
Exception: Si aucun résultat n'est retourné par le serveur après 5 secondes, un message d'erreur s'affiche au lieu du catalogue.
Post-conditions : Aucune

3- Nom : CatalogueProduit_changeModeAffichage
Pré-conditions : L'utilisateur change le mode d'affichage
Description : Deux modes d'affichage :
Vignette : <ul style="list-style-type: none">Affichage de 4 produits par ligne avec une grande image.
Liste : <ul style="list-style-type: none">Affichage d'un tableau de produits avec un petite image. <p>L'affichage de chaque produit comprend une image, une description, le nom de la catégorie, le prix et un bouton "Ajouter au panier" (non fonctionnel).</p>
Exception: Aucune
Post-conditions : Aucune

4- Nom : CatalogueProduit_changeNombreProduit
Pré-conditions : L'utilisateur change le nombre de produit par page
Description : L'utilisateur peut choisir entre : <ul style="list-style-type: none">12 produits/page24 produits/page36 produits/pagetous
Exception: Aucune
Post-conditions : Si la page active n'est pas la première page, la page active est changée pour la page 1 (CatalogueProduit_changePage)

5- Nom : AfficherCatalogueProduit_changePage
Pré-conditions : L'utilisateur choisi de changer la page qui est affiché.
Description : <ul style="list-style-type: none">• Si l'utilisateur appuie sur le bouton suivant la page est incrémentée• Si l'utilisateur appuie sur le bouton précédent, la page est décrémentée• L'utilisateur peut choisir directement une page en cliquant sur le bouton correspondant
Exception: <ul style="list-style-type: none">• Si la page choisie est inexistante, ne pas effectuer de changement et afficher un message d'erreur à l'utilisateur.
Post-conditions : Le catalogue est affiché selon le cas CatalogueProduit_afficherPage .

Requête serveur et structure des données JSON des produits

Les requêtes doivent se faire selon les paramètres suivants :

URL : <http://jmartel.webdev.cmaisonneuve.qc.ca/582-853/produit/ajaxProduits.php>

Requête du nombre total de produit (GET) :

requete = nombreProduits

Requête de produit (GET) :

requete = produits

index= index du premier élément

nombre = nombre de produit à retourner

La valeur de retour est structurée selon le format JSON suivant :

```
{
  "produits":
  [
    {
      "nom": "Nom du produit",
      "image": url,
      "description": "Une description assez courte",
      "prix": {
        "valeur": prix,
        "unite": $can
      },
      "categorie": Nom categorie
    },
    {
      "nom": "Nom du produit",
      "image": url,
      "description": "Une description assez courte",
      "prix": {
        "valeur": prix,
        "unite": $can
      },
      "categorie": Nom categorie
    }
  ]
}
```

Toutes les images utilisées auront la même dimension (300 x 300 pixels).