

Transformations XSLT (2)

Le but de ce TP est d'écrire des transformations servant à se substituer à une implémentation impérative d'un site web. Les fichiers **XML** de ce TP décrivent des informations sur les plantes, les familles de plantes, et les commandes passées auprès d'un vendeur de plantes. Nous déploierons une solution essentiellement basée sur les technologies **XML/XSLT** pour gérer la consultation des plantes (par famille, avec ou sans affichage de la famille), la gestion d'un panier et de commandes.

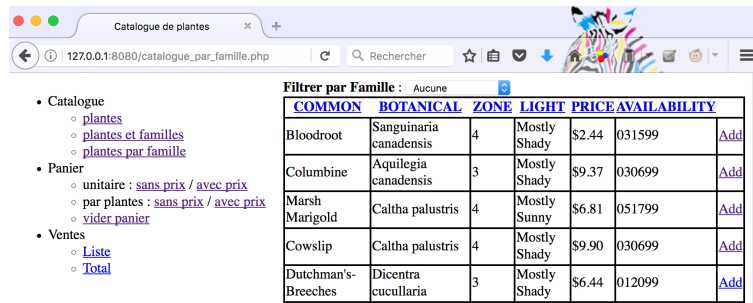


FIGURE 1 – Aperçu des fonctionnalités du site web

Nous commençons par un travail préparatoire afin de se familiariser avec la transformation des documents XML directement dans le navigateur en leur associant des feuilles de styles XSLT.

Nous attirons votre attention, sur le fait que les feuilles de styles XSLT ne sont supportées qu'en version "1.0" dans les divers navigateurs. Les moteurs XSLT ne respectent pas entièrement le standard XSLT. Certaines restrictions sont imposées par les navigateurs (par exemple, pour la famille Mozilla il faut se référer à <https://developer.mozilla.org/en-US/docs/Web/XSLT>).

Exercice 1 : Transformer les documents XML avec XSLT dans un navigateur

Pour réaliser la transformation directement dans le navigateur, vous devez ajouter une instruction de processing dans l'entête du fichier **plant_catalog.xml** comme indiqué ci-dessous :

```
<? xml version="1.0" encoding="ISO-8859-1" ?>
<? xml-stylesheet type="text/xsl" href="catalogue.xsl" ?>
<CATALOG>
  <PLANT>...</PLANT> ...
</CATALOG>
```

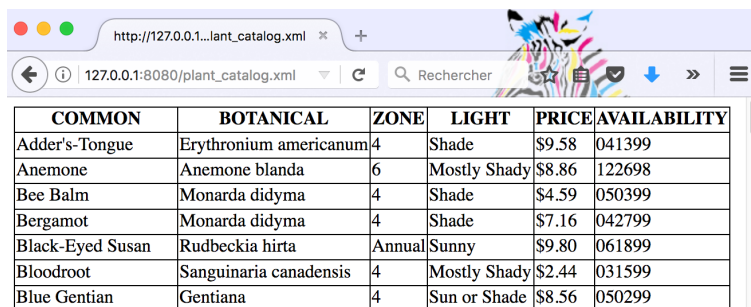
Afin que le résultat de la transformation soit affiché comme une page HTML dans le navigateur, il faut indiquer dans la feuille **catalogue.xsl** que le **output** est de type **html** et il faut associer aucun espace de noms sur les éléments créés par la feuille de style, comme illustré ci-dessous :

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="html"/>
  ...
</xsl:stylesheet>
```

Lorsque le document **XML** auquel nous avons associé la feuille de style est chargé dans la navigateur, il faut veiller à ce que le chargement ne se fasse pas directement depuis le système de fichiers **file** ://, mais en passant par un serveur web (local). Cette contrainte est imposée par des considérations liées à la sécurité (similaires aux contraintes liées au cross-domain).

La mise en place d’un serveur web à partir d’un répertoire local (contenant les fichiers .xml et .xsl), peut être réalisée avec la commande suivante : **php -S 127.0.0.1:8080**

Question 1 : Complétez le contenu de la feuille de style **catalogue.xsl** afin de générer une page de présentation du catalogue de plantes sous la forme d’un tableau comme indiqué dans la Figure ci-dessous.



COMMON	BOTANICAL	ZONE	LIGHT	PRICE	AVAILABILITY
Adder's-Tongue	Erythronium americanum	4	Shade	\$9.58	041399
Anemone	Anemone blanda	6	Mostly Shady	\$8.86	122698
Bee Balm	Monarda didyma	4	Shade	\$4.59	050399
Bergamot	Monarda didyma	4	Shade	\$7.16	042799
Black-Eyed Susan	Rudbeckia hirta	Annual	Sunny	\$9.80	061899
Bloodroot	Sanguinaria canadensis	4	Mostly Shady	\$2.44	031599
Blue Gentian	Gentiana	4	Sun or Shade	\$8.56	050299

FIGURE 2 – Catalogue de plantes

Exercice 2 : Création d’un site web de ventes de plantes

Lorsque l’on associe une feuille de style à un document XML chargé dans le navigateur, l’ensemble des transformations sont appliquées côté client dans le navigateur. Ceci est intéressant d’un point de vue de la répartition de la charge. En revanche, le mécanisme tel que présenté dans l’exercice précédent reste limité, car les navigateurs ne permettent pas :

- d’appliquer plus d’une feuille de style à un document
- de paramétrer convenablement la feuille de style (par exemple afficher les plantes par ordre alphabétique, ou bien par ordre de prix, etc.)

Les restrictions imposées par les transformations côté navigateur font que la personnalisation des feuilles de style ne peut être mise en place que côté serveur. Deux solutions peuvent se dégager :

- écrire des feuilles de style produisant à leur tour des feuilles de style - cette solution n’est pas supportée par les implémentations actuelles des navigateurs.
- préparer côté serveur des documents xml référençant les fichiers concernés par la transformation et les éventuels paramètres. La feuille de style associé peut adapter son comportement.

Dans la suite de ce TP nous mettrons en place des solutions s’appuyant sur la deuxième classe de solutions. Des fichiers PHP assureront la récupération des paramètres en entrée des transformations et l’orchestration de la mise en place des transformations. Ainsi, au lieu d’associer des feuilles de style, directement au fichier XML cible, nous associerons les feuilles de style à des documents XML *éphémères* générés par les fichiers PHP.

Les fichiers PHP auront pour mission de :

- récupérer des paramètres encodés dans les URL des pages (par exemple critère de recherche)
- construire un document temporaire indiquant les paramètres récupérés.

Le détail du processus mis en oeuvre est illustré dans la figure ci-dessous. Suite à la récupération d’une demande d’affichage d’une page Web, le **moteur PHP**, va générer un document *éphémère* XML associé à la feuille de style cible. Lors de la récupération de ce fichier XML par le navigateur, celui-ci va le déléguer à son propre **moteur XSLT** qui demandera le chargement des documents

référéncés et paramètrera le fonctionnement de la feuille de style. Une fois la transformation réalisée le contenu html ainsi généré sera affiché dans le navigateur. Cette architecture permet d'appliquer **une multitude de feuilles de styles paramétrées aux mêmes données XML**.

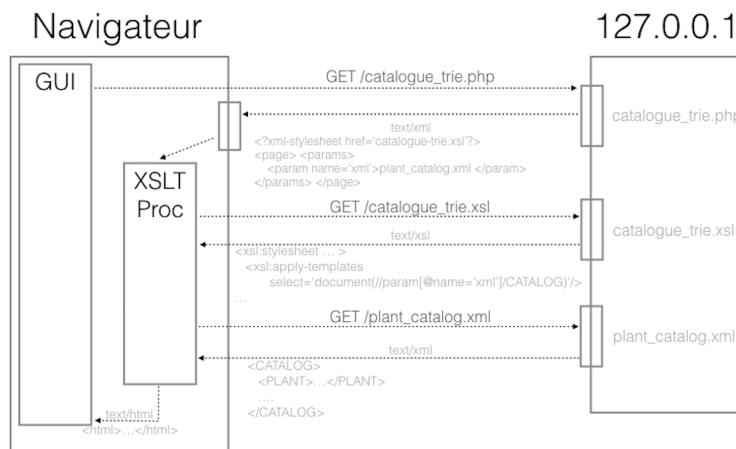


FIGURE 3 – Architecture hybride PHP XSLT pour personnaliser la transformation

Dans le code ci-dessous nous illustrons le contenu d'un fichier PHP **catalogue_trie.php** préparant l'application de la feuille de style **catalogue_trie.xsl** sur le document **plant_catalog.xml**

```
1: <?php header("Content-type: text/xml");?>
2: <?xml-stylesheet type="text/xsl" href="catalogue_trie.xsl"?>
3: <?php
4:   echo "<page> <params>
5:       <param name='cle_tri'>".$REQUEST['cle_tri'].</param>
6:       <param name='catalog_xml'>plant_catalog.xml</param>
7:       </params> </page>";
8:   ?>
```

Afin que le fichier PHP soit considéré comme produisant un document XML nous avons introduit (ligne 1) la commande **header** indiquant que le contenu généré sera bien du XML. La ligne 2 associe la feuille de transformation **catalogue_trie.xsl** au document XML *éphémère* qui sera généré. Les lignes 4 à 7 préparent le document XML *éphémère* qui sert à stocker les paramètres de la transformation. En effet, pour disposer d'un moyen de paramétrer le fonctionnement de la feuille, le document XML considéré par la transformation référence le véritable document XML contenant les données (ligne 6), ainsi que le paramètre qui servira à personnaliser le tri (ligne 5).

Lors du chargement de la page `http://127.0.0.1:8080/catalogue_trie.php?cle_tri=COMMON` le document XML suivant sera généré :

```
<?xml-stylesheet type="text/xsl" href="catalogue_trie.xsl"?>
<page><params>
  <param name='cle_tri'>COMMON</param>
  <param name='catalog_xml'>plant_catalog.xml</param>
</params></page>
```

La feuille de transformation **catalogue_trie.xsl** peut être définie ainsi :

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:include href="html_wrapper.xsl"/>
  <!-- html_wrapper.xsl contient les templates nommes header et menu -->
```

```

<xsl:template match="/">
  <xsl:variable name='catalog_xml' select="//param[@name='catalog_xml']"/>
  <xsl:variable name='cle_tri' select="//param[@name='cle_tri']"/>
  <html>
    <xsl:call-template name="header"/>
    <body>
      <xsl:call-template name="menu"/>
      <div id="content">
        <xsl:apply-templates select="document($catalog_xml)//CATALOG">
          <xsl:with-param name="cle_tri" select="$cle_tri" />
        </xsl:apply-templates>
      </div>
    </body>
  </html>
</xsl:template>

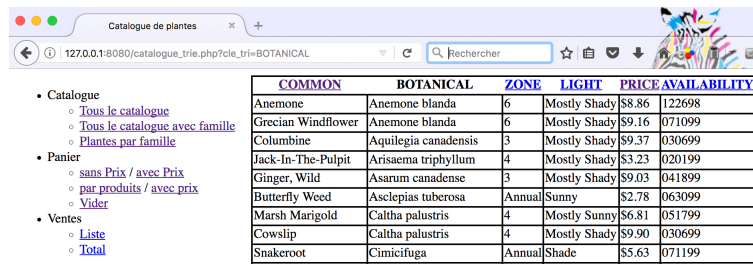
<xsl:template match="CATALOG">
  <xsl:param name="cle_tri"/>...
</xsl:template>

<xsl:template match="PLANT">
  <xsl:param name="cle_tri"/>...
<xsl:template>
</xsl:stylesheet>

```

Question 1 : Complétez les templates CATALOG et PLANT pour obtenir les rendus suivants en fonction des critères de tri indiqués dans l'URL :

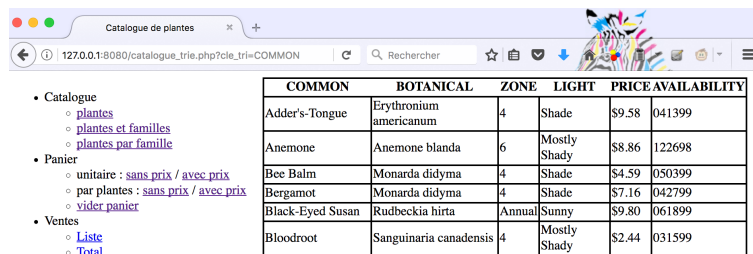
- http://127.0.0.1:8080/catalogue_trie.php?cle_tri=BOTANICAL



COMMON	BOTANICAL	ZONE	LIGHT	PRICE	AVAILABILITY
Anemone	Anemone blanda	6	Mostly Shady	\$8.86	122698
Grecian Windflower	Anemone blanda	6	Mostly Shady	\$9.16	071099
Columbine	Aquilegia canadensis	3	Mostly Shady	\$9.37	030699
Jack-In-The-Pulpit	Arisaema triphyllum	4	Mostly Shady	\$3.23	020199
Ginger, Wild	Asarum canadense	3	Mostly Shady	\$9.03	041899
Butterfly Weed	Asclepias tuberosa	Annual	Sunny	\$2.78	063099
Marsh Marigold	Caltha palustris	4	Mostly Sunny	\$6.81	051799
Cowslip	Caltha palustris	4	Mostly Shady	\$9.90	030699
Snakeroot	Cimicifuga	Annual	Shade	\$5.63	071199

FIGURE 4 – Catalogue trié par BOTANICAL

- http://127.0.0.1:8080/catalogue_trie.php?cle_tri=COMMON

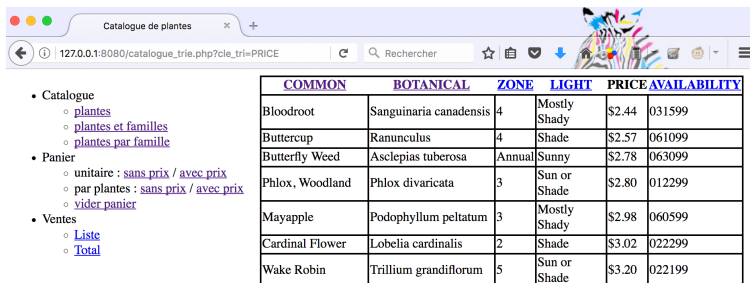


COMMON	BOTANICAL	ZONE	LIGHT	PRICE	AVAILABILITY
Adder's-Tongue	Erythronium americanum	4	Shade	\$9.58	041399
Anemone	Anemone blanda	6	Mostly Shady	\$8.86	122698
Bee Balm	Monarda didyma	4	Shade	\$4.59	050399
Bergamot	Monarda didyma	4	Shade	\$7.16	042799
Black-Eyed Susan	Rudbeckia hirta	Annual	Sunny	\$9.80	061899
Bloodroot	Sanguinaria canadensis	4	Mostly Shady	\$2.44	031599

FIGURE 5 – Catalogue trié par COMMON

Question 2 : Modifiez la feuille de transformation de telle sorte que les entêtes de la table soient remplacées par des liens permettant d'accéder au tableau trié par le critère correspondant. Dans

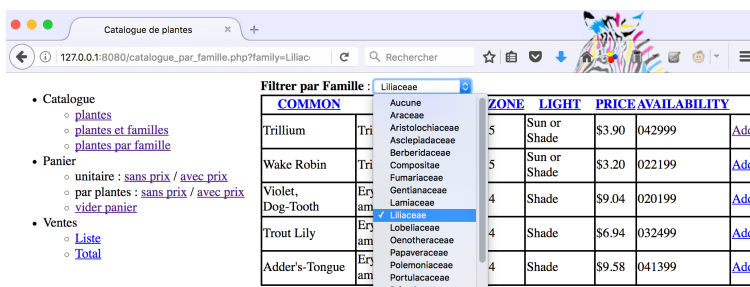
la figure ci-dessous où le catalogue est trié par PRICE, un click sur LIGHT va demander le rechargement de la page en triant le tableau par LIGHT. Pour cela chaque entête devra contenir des liens de la forme suivante : `COLONNE`



COMMON	BOTANICAL	ZONE	LIGHT	PRICE	AVAILABILITY
Bloodroot	Sanguinaria canadensis	4	Mostly Shady	\$2.44	031599
Buttercup	Ranunculus	4	Shade	\$2.57	061099
Butterfly Weed	Asclepias tuberosa	Annual	Sunny	\$2.78	063099
Phlox, Woodland	Phlox divaricata	3	Sun or Shade	\$2.80	012299
Mayapple	Podophyllum peltatum	3	Mostly Shady	\$2.98	060599
Cardinal Flower	Lobelia cardinalis	2	Shade	\$3.02	022299
Wake Robin	Trillium grandiflorum	5	Sun or Shade	\$3.20	022199

FIGURE 6 – Catalogue trié par PRICE

Question 3 : Les plantes sont organisées en familles. Nous désirons pouvoir de pouvoir explorer le catalogue en filtrant les plantes par famille. Ainsi, il vous est demandé de compléter la feuille de transformation `catalogue_par_famille.xsl` qui permet de générer le rendu suivant :

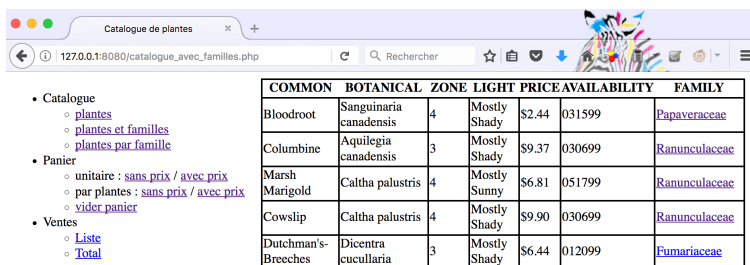


COMMON	BOTANICAL	ZONE	LIGHT	PRICE	AVAILABILITY	FAMILY
Trillium	Tri	5	Sun or Shade	\$3.90	042999	Add
Wake Robin	Tri	5	Sun or Shade	\$3.20	022199	Add
Violet, Dog-Tooth	Ery am	4	Shade	\$9.04	020199	Add
Trout Lily	Ery am	4	Shade	\$6.94	032499	Add
Adder's-Tongue	Ery am	4	Shade	\$9.58	041399	Add

FIGURE 7 – Catalogue filtré par famille

L'élément `select` construit par le template **CLASSIFICATION** est inclus dans un formulaire que déclenche le rechargement de la page en spécifiant le nouveau critère de filtrage à chaque changement de famille (`onChange="javascript:submit()"`). Lorsque aucune famille n'est indiquée, c'est l'ensemble des plantes qui est affiché.

Question 4 : Nous souhaitons maintenant enrichir les informations présentées dans le tableau avec la famille dont chaque plante fait partie comme indiqué ci-dessous. La colonne famille ne doit pas servir comme critère de tri, en revanche, en cliquant sur les noms des familles nous souhaitons arriver sur les pages listant l'ensemble des plantes d'une même famille (voir question précédente). Complétez la feuille `catalogue_avec_familles.xsl` afin d'obtenir un rendu similaire.



COMMON	BOTANICAL	ZONE	LIGHT	PRICE	AVAILABILITY	FAMILY
Bloodroot	Sanguinaria canadensis	4	Mostly Shady	\$2.44	031599	Papaveraceae
Columbine	Aquilegia canadensis	3	Mostly Shady	\$9.37	030699	Ranunculaceae
Marsh Marigold	Caltha palustris	4	Mostly Sunny	\$6.81	051799	Ranunculaceae
Cowslip	Caltha palustris	4	Mostly Shady	\$9.90	030699	Ranunculaceae
Dutchman's-Breeches	Dicentra cucullaria	3	Mostly Shady	\$6.44	012099	Fumariaceae

FIGURE 8 – Catalogue incluant information sur la famille des plantes

Exercice 3 : Pour aller plus loin ...

En s'appuyant sur la partie PHP de l'architecture exposée en Figure 3, il est également possible de mettre à jour les informations concernant le panier d'achat, d'enregistrer des ventes et mettre à jour, en conséquence, les stocks de plantes. La mise à jour des documents XML est rendue possible en appliquant côté serveur des transformations XSLT qui vont générer des nouvelles versions des documents XML.

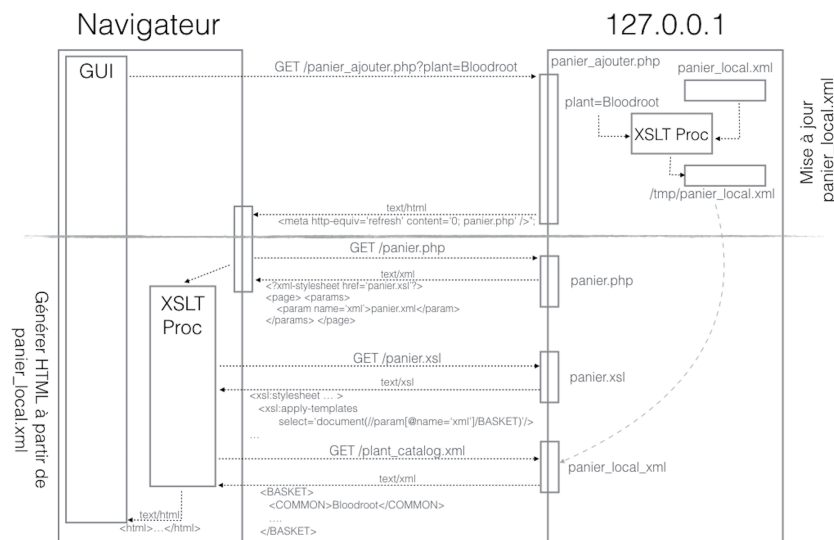


FIGURE 9 – Architecture hybride PHP XSLT pour personnaliser la transformation des données XML et mettre à jour les données XML

Dans la Figure 9 nous présentons dans le cadre de l'architecture introduite précédemment, les processus côté serveur et côté client qui permettront de réaliser la mise à jour du panier d'achats **panier_local.xml**. La transaction commence par une requête d'ajout d'une plante au panier **http://127.0.0.1:8080/panier_ajouter.php?plant=Bloodroot**. Lorsque cette demande est traitée par le serveur, l'exécution de **panier_ajouter.php** se traduit par le lancement d'une transformation **panier_ajouter.xsl** - paramétrée avec le nom de la plante afin d'ajouter une ligne dans le document **panier_local.xml**. La mise à jour du **panier_local.xml** se fait en deux temps. Premièrement, un fichier temporaire est obtenu suite à la transformation, et ensuite, le fichier temporaire remplace **panier_local.xml**. Ensuite, une demande de redirection est renvoyée au navigateur pour charger la page présentant le contenu du panier. Cette deuxième partie de la transaction est très similaire à ce que nous avons exposé dans l'exercice précédent.

Question 1 : Complétez la feuille de style **panier_ajouter.xsl** afin de rendre l'ajout d'un élément effectif. Le nom **COMMON** de la plante à ajouter au panier est envoyé en tant que paramètre de la feuille de style en utilisant l'option **-stringparam** de **xsltproc**. Le paramètre ainsi défini jouit d'une visibilité globale au niveau de la feuille de style.

Pour tester cette fonctionnalité vous pouvez taper directement dans la barre de navigation l'URL rendant effectif l'ajout : **http://127.0.0.1:8080/panier_ajouter.php?plant=Bloodroot**

Question 2 : Modifiez la ou les feuilles de style **catalogue_*.xsl** pour ajouter pour chaque plante un lien "Add" qui permet d'ajouter la plante en question dans le panier. Par exemple, pour la plante dont le nom **COMMON** est **Bloodroot**, le lien généré sera de la forme **href="panier_ajouter.php?plant=Bloodroot"**. Le rendu attendu est illustré dans la Figure ci-dessous pour la feuille **catalogue_par_famille.xsl**.

- Catalogue
 - plantes
 - plantes et familles
 - plantes par famille
- Panier
 - unitaire : sans prix / avec prix

Filtrer par Famille : Aucune						
COMMON	BOTANICAL	ZONE	LIGHT	PRICE	AVAILABILITY	
Bloodroot	Sanguinaria canadensis	4	Mostly Shady	\$2.44	031599	Add
Columbine	Aquilegia canadensis	3	Mostly Shady	\$9.37	030699	Add

FIGURE 10 – Visualisation du catalogue avec ajout au panier

Question 3 : Complétez la feuille de style **panier.xml** pour permettre d'afficher le contenu du panier en triant les éléments par ordre alphabétique. Prévoyez également la génération d'un lien "Retirer" pour chaque item dans le panier. Par exemple, pour supprimer le deuxième item du panier le lien généré sera de la forme **href="panier_supprimer.php?no_item=2"** . Le rendu attendu est illustré dans la Figure ci-dessous.

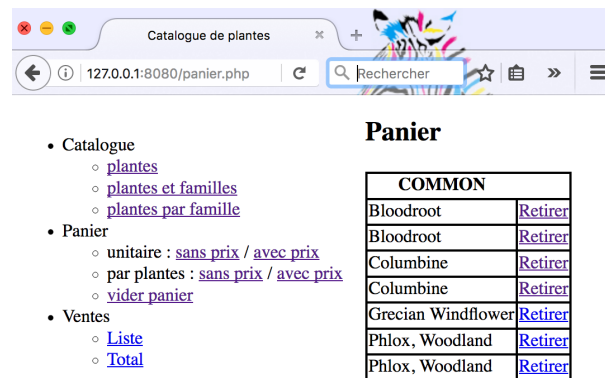


FIGURE 11 – Visualisation du panier produit par produit

Question 4 : Créez une nouvelle feuille de style **panier_avec_prix.xml** qui permettra d'afficher le contenu du panier en reportant également les prix des plantes. Cela nécessite de construire une feuille de style qui exploitera à la fois les informations contenues dans **panier_local.xml** et **plant_catalog.xml**. Prévoyez également la génération d'un lien "Retirer" pour chaque item dans le panier. Par exemple, pour supprimer le deuxième item du panier, le lien généré sera de la forme **href="panier_supprimer.php?no_item=2"** . Le rendu attendu est illustré dans la Figure ci-dessous.



FIGURE 12 – Visualisation du panier produit par produit en incluant le prix

Question 5 : Créez une nouvelle feuille de style **panier_regroupe.xml** qui permettra d'afficher le contenu du panier en regroupant la liste par plante et en indiquant la quantité ajoutée au panier.



COMMON	Qty
Bloodroot	2
Columbine	2
Grecian Windflower	1
Phlox, Woodland	2

FIGURE 13 – Visualisation du panier par groupe de produits

Question 6 : Créez une nouvelle feuille de style **panier_regroupe_avec_prix.xml** qui permettra d’afficher le contenu du panier en regroupant la liste par plante en indiquant à la fois la quantité, le prix unitaire et le prix total par ligne. De manière optionnelle, vous pouvez faire également apparaître le coût total du panier.



COMMON	qty	unit price	price
Bloodroot	2	\$2.44	\$4.88
Columbine	2	\$9.37	\$18.74
Grecian Windflower	1	\$9.16	\$9.16
Phlox, Woodland	2	\$2.80	\$5.6

FIGURE 14 – Visualisation du panier par groupe de produits en incluant le prix

Question 7 : Complétez la feuille de style **panier_supprimer.xml** afin de rendre la suppression d’un élément du panier effective. La position de l’élément à supprimer est envoyé en tant que paramètre de la feuille de style en utilisant l’option **–stringparam** de **xs:proc**. La déclaration du paramètre jouit d’une visibilité globale au niveau de la feuille de style..

Question 8 : Complétez la feuille de style **panier_vider.xml** qui permet de vider l’intégralité du panier.

Exercice 4 : Pour aller vraiment loin et de manière optionnelle...

Une fois le panier rempli, nous pouvons envisager la génération d’une commande. La possibilité de passer des commandes nécessite la mise en place des évolutions suivantes :

- ajouter un bouton **"Passer commande"** sur l’une des pages de consultation du panier
- générer un nouveau fichier **plant_order_NouveauID.xml** récapitulant les plantes, quantités et prix. Le NouveauID peut être obtenu en retrouvant le dernier ID commande dans **plant_orders.xml**
- référencer la nouvelle commande générée dans **plant_orders.xml**
- mettre à jour **plant_catalog.xml** en enlevant la quantité commandée de l’élément **AVAILABILITY** pour chacune des plantes de la commande.