

Documentation technique

Objectifs de l'application :

Simplifier la recherche des magasiniers dans la base de données :

- Ajout/Suppression/Modification de produits/gammes/marques/fournisseurs
(A noter que l'utilisateur peut également renseigner un lien pour les marques et fournisseurs)
- Recherche par critères (pouvant être sélectionnés ou non) de : Type de produit / Gamme / Marque / Fournisseur puis affichage d'un tableau des retours de la BDD
- Rechercher par mot puis affichage d'un tableau des retours de la BDD
- Ajout d'une relation

Affichage du tableau :

- Affichage du nom du produit/nom de la marque/nom du fournisseur/gamme de prix
- Dans une dernière colonne afficher des possibilités de modification ou de suppression
- Lorsque l'on clique sur « Suppression », l'application doit envoyer un message de confirmation
- Lorsque l'on clique sur « Modifier » : redirection vers une page permettant de modifier la relation sélectionnée
- Si un lien est renseigné, cliquer sur un fournisseur ou une marque permet de rediriger vers le site en question
- Création d'un bouton de tri par ordre alphabétique pour le nom de la marque et du fournisseur et par ordre de prix pour la gamme de prix.

Outils utilisés :

- Visual Studio code
- XAMPP
- MySQL
- Apache
- HTML/CSS/JS/PHP

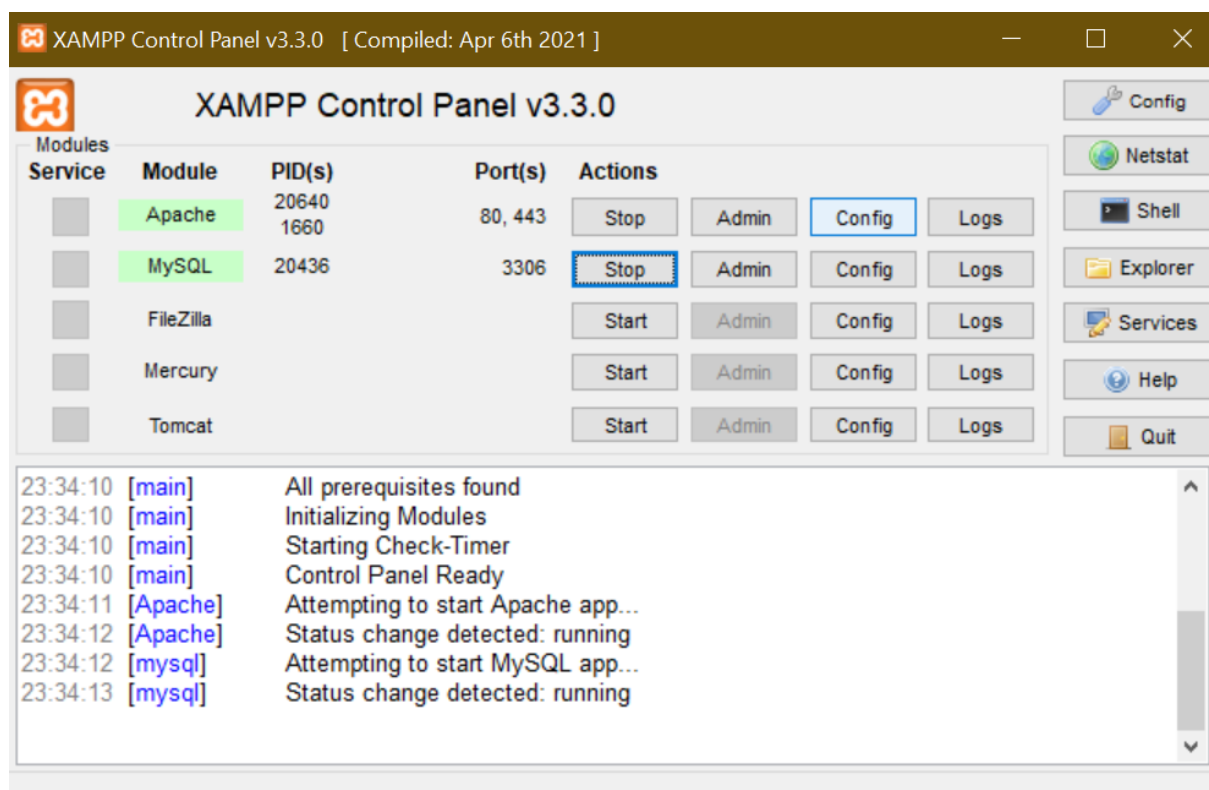
Mode opératoire :

Pour accéder au site, il faut installer xampp et placer dans le dossier xampp\htdocs votre dossier de site :

C:\xampp\htdocs\DTPL

Pour importer une BDD, tapez « localhost/phpmyadmin » dans votre navigateur, puis créez une BDD. Ensuite allez dans l'onglet importer et sélectionnez votre fichier SQL.

Pour lancer le site, lacer XAMPP Control Panel et démarrez les modules Apache et MySQL :



Tapez ensuite dans votre navigateur : localhost/votresite

Fonctionnement du modèle MVC :

Les différentes fonction PHP se situent dans le fichier modele/bd.votrefichier.php :

```
modele > bd.recherche.php > getLesGammeByNomGamme
1 <?php
2 include_once "$racine/modele/bd.pdo.php";
3 function getLesFournisseursByNomFournisseur($mot)
4 {
5     try {
6         $connex = connexionPDO();
7         $req = "SELECT idFournisseur FROM fournisseurs WHERE nomFournisseur LIKE ?";
8         $prep = $connex->prepare($req);
9         $prep->bindValue(1, "%" . $mot . "%");
10        $prep->execute();
11        $resultat = $prep->fetchAll(PDO::FETCH_OBJ); // Fetch all rows as objects
12        return $resultat;
13    } catch (PDOException $e) {
14        print "Erreur !: " . $e->getMessage();
15        die();
16    }
17 }
18 function getLesGammeByNomGamme($mot)
19 {
20     try {
21         $connex = connexionPDO();
22         $req = "SELECT idGamme FROM gammes WHERE nomGamme LIKE ?";
23         $prep = $connex->prepare($req);
24         $prep->bindValue(1, "%" . $mot . "%");
25         $prep->execute();
26         $resultat = $prep->fetchAll(PDO::FETCH_OBJ); // Fetch all rows as objects
27         return $resultat;
28     } catch (PDOException $e) {
29         print "Erreur !: " . $e->getMessage();
30         die();
31     }
32 }
```

Les appels aux fonctions et affichage des pages se fait dans le fichier controleur/votrefichier.php :

```
controleur > rechercheFournisseur.php > ...
1 <?php
2 $titre = "Recherche de fournisseurs";
3 include_once "$racine/vue/entete.html.php";
4 include_once "$racine/vue/vueRechercheFournisseur.php";
5 include_once "$racine/modele/bd.rechercheFournisseur.php";
6 include_once "$racine/modele/bd.recherche.php";
7 include_once "$racine/modele/bd.gestion.php";
8 $touslesproduits = getLesProduits();
9 $touteslesgammes = getLesGammes();
10 $touteslesmarques = getLesMarques();
11 $touslesfournisseurs = getLesFournisseurs();
12 $gestionProduits = getLesProduits();
13 $gestionGammes = getLesGammes();
14 $gestionMarques = getLesMarques();
15 $gestionFournisseurs = getLesFournisseurs();
```

Toute la partie visuelle se fait dans le fichier vue/vueVotreFichier.php :

```

1 <?php
2 |
3 session_start();
4
5 ?>
6 <div id="grosdivrecherche">
7   <fieldset id="divcriteres">
8     <h1 style="text-align: center;">Sélectionnez les critères :</h1>
9     <?php
10     $selectedTypeProduit = $_POST['typeProduit'] ?? 'NULL';
11     $selectedGame = $_POST['quelleGame'] ?? 'NULL';
12     $selectedMarque = $_POST['quelleMarque'] ?? 'NULL';
13     $selectedFournisseur = $_POST['quelFournisseur'] ?? 'NULL';
14     ?>
15     <form method="POST" action="" id="formaselec">
16
17       <div style="display: block;">
18         <label for="typeProduit">Type de produit : </label>
19         <select name="typeProduit" class="inputselect">
20           <option value="NULL">Peu importe</option>
21           <?php
22             foreach ($touslesproduits as $unproduit) {
23               $isSelected = ($unproduit->idProduit == $selectedTypeProduit) ? 'selected' : '';
24               ?>
25               <option value="<?> $unproduit->idProduit ?>" <?> $isSelected ?>> <?> $unproduit->nomPr
26             <?php
27             }
28           ?>
29         </select>
30         <label for="quelleGame">Game : </label>
31         <select name="quelleGame" class="inputselect">
32           <option value="NULL">Peu importe</option>
33           <?php

```

Pour récupérer les variables dans les documents, j'ai globalement utilisé des form method POST.

Pour effectuer une fonction lors d'un click sur un bouton, on appelle une fonction en récupérant les variables avec \$_POST['variable'] dans le contrôleur :

```

if (isset($_POST['Rechercher'])) {
    $produit = isset($_POST['typeProduit']) ? $_POST['typeProduit'] : '';
    $game = isset($_POST['quelleGame']) ? $_POST['quelleGame'] : '';
    $marque = isset($_POST['quelleMarque']) ? $_POST['quelleMarque'] : '';
    $fournisseur = isset($_POST['quelFournisseur']) ? $_POST['quelFournisseur'] : '';

```

Pour certains cas, notamment la modification d'une relation, il est nécessaire de passer une variable dans l'url avec un <a> :

```
<a href="./?action=modif&quellemodif=<?> $unique->idRelation ?>">Modifier</a>
```

```
localhost/dtpl/?action=modif&quellemodif=103|
```

Dans ce cas nous récupérerons la variable et l'utiliseront comme argument dans une fonction avec la méthode GET :

```
$info = getToutParID($_GET['quellemodif']);
```

Pour utiliser une variable retour d'une fonction, il faut préciser le champ de la BDD voulu :

```
<p>Type de produit actuel : <?> $info->nomProduit ?></p>
```

Pour affecter un style CSS, il faut utiliser une propriété link rel= « stylesheet » dans le <head> :

```
<link href="css/style.css" rel="stylesheet">
```