

Evolution n°1 : affichage informations détaillées d'un véhicule

- a) Afficher un n° de véhicule dans l'affichage des véhicules d'occasion : ce n° est un n° d'ordre, ce n'est pas la clé de la table vehicle

Liste de tous les véhicules				
N°	Marque	Modèle	Km	Prix
1	Peugeot	2008	5000	7 000,00 €
2	Renault	Captur	12000	11 000,00 €

- b) Créer un lien hypertexte sur ce n° de manière à afficher une page fournissant les informations détaillées du véhicule.

Remarques :

- il s'agit toujours du contrôleur accueil, mais l'action à passer en paramètre est **detail** et il faudra également passer en paramètre l'identifiant du véhicule. On utilisera le paramètre **id** pour passer cette valeur.

Lien pour afficher le détail du véhicule d'identifiant 1 :
<index.php?uc=accueil&action=detail&id=1>

- c) Ecrire dans **gestion_bdd.php** une nouvelle fonction **getLeVehicule(\$id)** recevant en paramètre d'entrée une variable **\$id** (identifiant du véhicule) dont l'objectif est d'afficher toutes les informations du véhicule.
- d) Modifier le contrôleur **c_accueil** de manière à ajouter la gestion de l'action **detail** qui prendra en charge les étapes suivantes :
- Récupérer l'identifiant du véhicule passé dans la barre d'adresse
 - Récupérer dans une variable **\$idVehicule** les informations du véhicule puis appeler la fonction **getLeVehicule()**
 - Appeler une vue nommée **v_vehicule_detail.php**
- e) Créer la vue **v_vehicule_detail.php** qui prend en charge l'affichage des informations détaillées du véhicule.

Informations détaillées d'un véhicule

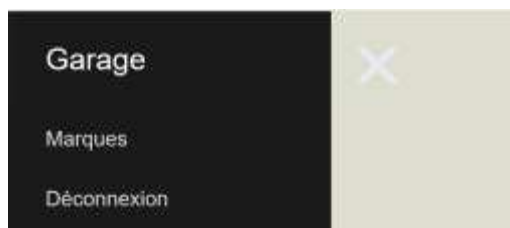
Marque	Modèle
Peugeot	2008
Energie	Kilométrage
Essence	5000
Couleur	Prix
Gris clair	7000.00
Options	
Peinture métallisée	

Evolution n°2 : formulaire d'authentification

- a) Ajouter un formulaire d'authentification sur la page d'accueil

Configurer ce formulaire pour qu'il envoie les informations saisies en POST à un nouveau contrôleur d'authentification. Aligner ce formulaire à droite.

- b) Créer le contrôleur `c_authentification.php` qui prendra en charge la réception des données Login et Mot de passe et l'appel de la fonction **verifierAuthentification()** déjà présente dans le fichier `gestion_bdd.php` dont il faudra revoir la requête SQL.
Attention, les mots de passe sont cryptés en MD5 dans la base de données. Les mots de passe en clair sont **pass**.
- c) Une authentification avec succès permettra d'afficher un menu comme suit (déjà configuré dans le fichier `header.php`)



- d) Lorsqu'une personne est authentifiée, le formulaire de connexion ne doit pas être affiché

Evolution n°3 : gérer la saisie, mise à jour et suppression des marques

- a) Créer un nouveau cas dans le contrôleur principal pour gérer le nouveau contrôleur **c_marque.php** qui sera à créer.
- b) Dans le contrôleur **c_marque.php**, le cas par défaut sera **afficher**. Ce cas permettra d'afficher dans l'ordre alphabétique les marques présentes dans la base de données.
 1. Ecrire la requête dans la fonction **getLesMarques()** de gestion_bdd.php
 2. Ecrire les lignes dans le contrôleur permettant l'appel de cette requête et l'appel de la vue associée :

v_marque.php

- c) Créer la vue **v_marque.php** permettant l'affichage des marques de la façon suivante :

Gestion des marques des véhicules		
Ajouter une nouvelle marque 		
Libellé		
BMW	<input type="button" value="Modifier"/>	<input type="button" value="Supprimer"/>
Fiat	<input type="button" value="Modifier"/>	<input type="button" value="Supprimer"/>
Mercedes	<input type="button" value="Modifier"/>	<input type="button" value="Supprimer"/>
Nissan	<input type="button" value="Modifier"/>	<input type="button" value="Supprimer"/>
Peugeot	<input type="button" value="Modifier"/>	<input type="button" value="Supprimer"/>
Renault	<input type="button" value="Modifier"/>	<input type="button" value="Supprimer"/>
Seat	<input type="button" value="Modifier"/>	<input type="button" value="Supprimer"/>
Toyota	<input type="button" value="Modifier"/>	<input type="button" value="Supprimer"/>
Volkswagen	<input type="button" value="Modifier"/>	<input type="button" value="Supprimer"/>

Lien hypertexte permettant d'appeler l'action **ajouter** de l'uc **marque**

Les boutons sont des formulaires permettant d'appeler l'action **modifier** ou **supprimer** de l'uc **marque**.

Ces formulaires contiennent en champ caché l'identifiant de la marque.

Ajout d'une marque

- d) Ajouter dans le contrôleur **c_marque.php** le cas **ajouter** qui permettra d'afficher la vue **v_marque_ajouter.php** contenant un formulaire renvoyant vers un nouveau cas **validajout** en utilisant la méthode POST :

Ajout d'une marque
<input type="text" value="Libellé de la marque"/>
<input type="button" value="Valider"/>

- e) Ajouter dans le contrôleur **c_marque.php** le cas **validajout** qui permettra de réaliser les actions suivantes :
 1. Tester si le libellé de la marque est saisi, si non afficher un message d'erreur en rouge :

Veillez saisir une marque

Retour permet de revenir au formulaire d'ajout ci-dessus.

2. Exécuter la requête d'insertion d'une marque (fonction **marqueInsertion** admettant un paramètre = le libellé de la marque : cette fonction est à écrire dans le fichier gestion_bdd.php), voici comment elle se présente :

```
/**
 *      insertion d'une marque
 *      Auteur : Corrigé
 *      @param $libelle : libellé de la marque
 *      @return un booléen indiquant si l'insertion a pu se faire
 */
function marqueInsertion($libelle) { require "connexion.php" ;
    $sql = "insert into marque (mq_libelle) values ('$libelle') " ;
    $exec=$bdd->prepare($sql) ; $resultat = $exec->execute() ; return $resultat;
}
```

3. Récupérer le résultat de l'exécution de cette fonction : si l'insertion a pu se faire, on affichera :

La marque Fiat a bien été ajoutée
Retour

Retour permet de revenir à la liste des marques.

4. si l'insertion n'a pu se faire, on affichera :

La marque Citroen n'a pas été ajoutée
Retour

Retour permet de revenir au formulaire d'ajout.

Remarque : vous réfléchirez à la façon d'afficher un message d'erreur au cas où la marque saisie existe déjà.

Modification d'une marque

f) Ajouter dans le contrôleur **c_marque** l'action **modifier** qui prendra en charge l'exécution de la requête **getLesMarques()** et l'affichage de la vue **v_marque.php**

g) Modifier la vue **v_marque.php** fin qu'elle puisse assurer la fonction de modification :

1. Dans la boucle d'affichage, repérer si la marque doit être mise à jour :

```
if (isset($_REQUEST['action']) && ($_REQUEST['action'] == 'modifier')
&& isset($_REQUEST['id']) && ($_REQUEST['id'] == $laMarque['mq_id']))
```

2. Si la marque doit être modifiée, afficher un formulaire permettant la saisie et validation de la marque, le bouton supprimer n'apparaît pas :

Libellé		
BMW	Valider	
Fiat	Modifier	Supprimer

BMW prêt à être modifié.
FIAT conforme page 1

3. Si la modification n'est pas demandée pour la marque, l'affichage reste conforme à celui de la page 1

```

foreach ($lesMarques as $laMarque) {
    echo '<tr>' ;

    if (isset($_REQUEST['action']) && ($_REQUEST['action'] == 'modifier')
        && isset($_REQUEST['id']) && ($_REQUEST['id'] == $laMarque['mq_id'] )) {
        echo ''
        . '<form method="POST" action="index.php?uc=marque&action=validmodif">'
        . '<td><input type="hidden" name="id" value="' . $_REQUEST['id'] . '">'
        . '<input type="text" name="mq_libelle" value="' . $laMarque['mq_libelle'] . '"></td>'
        . '<td><input type="submit" value="Valider"></td>'
        . '</form>'
        . '</td>'
        . '<td></td>' ;
    }
    else {
        echo '<td>'
        . $laMarque['mq_libelle']
        . '</td>'
        . '<td>'
        . '<form method="POST" action="index.php?uc=marque&action=modifier">'
        . '<input type="hidden" name="id" value="' . $laMarque['mq_id'] . '">'
        . '<input type="submit" value="Modifier">'
        . '</form>'
        . '</td>' ;
        echo
        . '<td>'
        . '<form method="POST" action="index.php?uc=marque&action=supprimer">'
        . '<input type="hidden" name="id" value="' . $laMarque['mq_id'] . '">'
        . '<input type="submit" value="Supprimer">'
        . '</form>'
        . '</td>' ;
    }
    echo '</tr>';
}

```

Cas de la modification du libellé :

Formulaire permettant la modification de la marque existante. Ce formulaire redirige vers l'action validmodif du cas marque en envoyant avec la méthode POST la valeur du libellé saisi.

Pas de modification du libellé : affichage du libellé et des bouton Modifier et Supprimer

h) Reprendre ce qui a été fait sur l'action **validajout** et réaliser la mise à jour du libellé d'une marque par écriture d'une nouvelle fonction **marqueModification** admettant deux paramètres : l'identifiant et le libellé de la marque.

Suppression d'une marque

i) Ajouter une nouvelle action **supprimer** dans le contrôleur **c_marque.php** de manière à prendre en compte la suppression d'une marque.

Evolution n°4 : introduction du javascript

Le javascript pour tester la validité des données saisies

Application : dans le formulaire d'ajout d'une marque, au lieu de tester si la zone est saisie dans le contrôleur (donc après envoi de la page au serveur), nous allons le faire au niveau du navigateur grâce à du javascript, cela évite l'envoi de données incorrectes au serveur.

- a) Ouvrir la page **v_marque_ajout.php** dans laquelle se trouve le formulaire de saisie d'une marque, et modifier la balise <form> de la manière suivante :

```
<form action="index.php?uc=marque&action=validajout" method="POST"
onsubmit="javascript:return estValideMarque()">
```

Le code en gras signifie que lorsqu'on cliquera sur le bouton submit (onsubmit) la fonction estValideMarque() sera exécutée et le submit ne se fera que si la fonction retourne la valeur true.

- b) Les fonctions écrites en javascript seront enregistrées dans le fichier **fonctions.js** du répertoire **includes/js**

- c) Ouvrir le fichier fonctions.js et écrire le code suivant :

```
function estValideMarque() {
    marque = document.forms[0].mq_libelle.value ;
    if (marque=="") { alert ("Le libelle de la marque est obligatoire") ;
    } else { return true ;
    }
}
```

Déclaration de la fonction estValideMarque()
Récupération dans le 1^{er} formulaire de la page (v_marque_ajout) de la valeur du champ nommé mq_libelle
Si ce champ est vide, une alerte sera affichée et la fonction retournera la valeur false.
Dans le cas contraire, la fonction retournera la valeur true.

- d) Enregistrer vos fichiers et rafraichir le navigateur. Tester votre formulaire. Sans donnée saisie, voici ce que vous devez voir :



- e) Et avec une donnée saisie, l'enregistrement en base de données se fait et on revient à la liste des marques.

Le javascript pour rafraichir une page

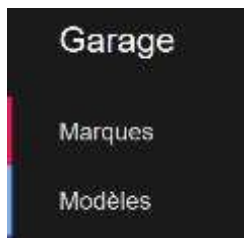
Application : nous allons ajouter une nouvelle option dans le menu : **Modèles** dont l'objectif est d'afficher la liste des modèles par marque.



Liste déroulante à balise <select>

1^{ère} ligne -- choisir une marque --

Lignes suivantes : affichage des marques dans l'ordre alphabétique.



Modifier le fichier header.php pour introduire cette nouvelle option, l'uc sera égale à la valeur suivante : modele à index.php?uc=modele

Modifier le contrôleur principal pour qu'il gère l'appel à un nouveau contrôleur nommé c_modele.php

Créer ce nouveau contrôleur.

a) Dans le contrôleur c_modele.php, l'action par défaut sera **afficher**. Cette action fera appel à la requête getLesMarques() et affichera une vue nommée **v_modele.php** qui dans un premier temps affichera une liste déroulante pour le choix d'une marque de la façon suivante :

b) Comment écrire une liste déroulante en HTML ?

```
<select name="nomListe">
<option value="1">Choix 1</option>
<option value="2">Choix 2</option>
<option value="3">Choix 3</option>
<option value="4">Choix 4</option>
</select>
```

La liste déroulante porte le nom **nomListe**

Après sélection d'une ligne, la valeur choisie sera connue dans \$_REQUEST['nomListe']

Cette valeur sera égale à l'information **value**

Exemple : si la ligne **Choix 3** est cliquée, la valeur contenue dans \$_REQUEST['nomListe'] sera **3**

c) Code de la vue **v_modele.php** permettant l'affichage de la liste déroulante des marques :

```
<select name="mq_id">
<option value="0">-- choisir une marque --</option>
<?php
foreach ($lesMarques as $laMarque) {
    echo '<option value="" . $laMarque['mq_id'] . "'>' . $laMarque['mq_libelle'] . '</option>'; }
?>
```

</select>

- d) Tester votre page afin de vérifier que la liste d'affiche correctement

- e) Introduction d'une fonction javascript permettant de récupérer la valeur choisie dans la liste déroulante et de recharger la page en passant en paramètre dans l'url l'identifiant de la marque. Cela permettra ensuite d'aller rechercher les modèles de la marque choisie.

```
function refreshPageModele(mq_id) {  
    document.location="index.php?uc=modele&action=affichmodele&id=" + mq_id ; }
```

Fonction nommée **refreshPageModele** recevant en paramètre l'identifiant (colonne value du select) de la marque sélectionnée.

Redirection vers la page index.php avec l'uc **modele**, l'action **affichmodele** et l'id de la marque

- f) Rafraichir la page et choisir une marque de manière à vérifier l'URL. Par exemple, pour la marque RENAULT, on doit voir : `index.php?uc=modele&action=affichmodele&id=1`

- g) Coder l'action `affichmodele` du contrôleur `c_modele.php`. Il devra prendre en charge :
 - ✓ la récupération de l'identifiant de la marque,
 - ✓ l'appel d'une nouvelle fonction intégrant une requête de recherche de la marque passée en paramètre (pour affichage dans le titre de la page)
 - ✓ l'appel d'une nouvelle fonction intégrant une requête de recherche des modèles de la marque passée en paramètre (pour affichage de la liste des modèles)
 - ✓ l'appel d'une nouvelle vue `v_modele_liste.php`

- h) Coder l'affichage des informations dans la vue **v_modele_liste.php**. Vous devez obtenir un affichage conforme à la liste suivante :