Module: OSI, INTERNET ET PROGRAMMATION WEB

TP 5 – Programmation Web avec PHP - Correction

Objectif : apprentissage PHP, Base de Données et PHP
Correction : la correction sera diffusée la semaine qui suit : refaire les exercices avec la correction
Note : vous créerez sur nyx un répertoire ~/web/TP5 dans lequel vous placerez vos fichiers html ainsi qu'une page index.html

Objectif : créer des scripts du côté serveur avec PHP, utiliser les variables de session
Note : vous placerez vos fichiers PHP sur linserv1 : 134.59.22.1 ou linserv2 : 134.59.22.24 port 80 et 443, dans un répertoire `~/web/TP5`
Pour plus d'informations sur ces serveurs :
pour plus d'infos sur ces serveurs:
https://134.59.22.1/infos.php
https://134.59.22.7/infos.php

1.Scripts du côté serveur avec PHP

Exemple d'introduction

1.1.Créez une page 1.1.php qui affiche simplement la chaîne de caractères "Hello PHP, nous sommes le " suivie de la date du jour sur le serveur.

```
<html><body>
   Hello PHP, nous sommes le <?php echo Date("j/m/Y"); ?>
</body></html>
```

1.2.Créez une deuxième version 1.2.php permettant d'afficher à la suite de la date, le message "Bon matin" ou "Bonne après midi" en fonction de l'heure sur le serveur.

```
<html>
<body>
    Nous sommes le <?php echo Date("j/m/Y"); ?><br>
    <?php
        echo "il est ".Date("H:i:s")."<br>";
        if ( Date("H") < 12 ) {
            echo "Bon matin ..";
        } else {
            echo "Bonne après midi ..";
        }
    ?>
</body>
</html>
```

Variables d'environnement

1.3.Afficher dans un tableau la signification et la valeur des variables d'environnement suivante

```
$SERVER_ADDR
$HTTP_HOST
$REMOTE_ADDR
gethostbyAddr($REMOTE_ADDR)
$HTTP_USER_AGENT
```

```
<html>
<body>
    <table border="1">
    <caption align="top">Quelques variables d'environnement</caption>
    <tr><th align="center">Variable</th><th align="center">Valeur</th></tr>
    <tr><td>Adresse IP du serveur web</td>
        <td><?php echo $_SERVER['SERVER_ADDR'] ?></td></tr>
    <tr><td>Nom du serveur web</td>
         <td><?php echo $_SERVER['HTTP_HOST'] ?></td></tr>
    <tr><td>Adresse IP du client</td>
        <td><?php echo $_SERVER['REMOTE_ADDR'] ?></td></tr>
    <tr><td>Nom de la machine cliente</td>
        <td><?php echo gethostbyAddr($_SERVER['REMOTE_ADDR']) ?></td></tr>
    <tr><td>Navigateur du client</td>
        <td><?php echo $_SERVER['HTTP_USER_AGENT'] ?></td></tr>
```

```
</table>
</body>
</html>
```

1.4.Afficher toutes les variables d'environnement en appelant la fonction `phpInfo()`

Créer simplement une page phpinfo.php, avec le contenu suivant :

```
<?php phpInfo(); ?>
```

Formulaires

1.5.Créez un formulaire 1.5.html comportant 2 champs de texte nom, prénom, un menu d'options sexe (M ou F), et un menu de choix multiples vins (bordeaux, beaujolais, loire). Vous ferez appel à un script 1.5.php pour traiter les données du formulaire. Elles seront envoyées par la méthode GET.

```
<!-- fichier 1.5.html -->
<html><body>
<form action="1.5.php" method="get">
    Nom: <input type="text" name="nom"><br>
    Prénom: <input type="text" name="prenom"><br>
    Sexe: <select name="sexe">
              <option>M
              <option>F
          </select><br>
    Vins: <br>
          <select multiple name="vin[]">
              <option value="bordeaux">bordeaux</option>
              <option value="beaujolais">beaujolais</option>
              <option value="côtes de provence">côte de provence</option>
          </select><br>
    <input type="submit" value="submit Me!">
</form>
</body></html>
```

```
<!-- fichier 1.5.php -->
<html><body>
<h1>Affichage des données saisies</h1>
<ul>
  <li>Nom:    <?php print $_REQUEST['nom']    ?></li>
  <li>Prenom: <?php print $_REQUEST['prenom'] ?></li>
  <li>Sexe:   <?php print $_REQUEST['sexe']   ?></li>
  <li>Vins:
      <ul>
      <?php
         if (isset($_REQUEST['vin']))
             foreach($_REQUEST['vin'] as $v) print "<li>$v</li>";
      ?>
      </ul>
</ul>
<a href="javascript:history.back()">Essayez à nouveau</a>
</body>
</html>
```

1.6.Créez une nouvelle version 1.6.php qui permettra à la fois de créer et de traiter le formulaire de l'exercice précédent :
- si le script est appelé sans paramètre il créera le formulaire
- s'il est appelé par l'envoi de données (bouton submit du formulaire), il traitera les données.

Vous utiliserez un champ caché permettant de tester si le script est appelé par l'envoi de données ou pas (ce champ caché est nécessaire dans le cas où le script serait appelé par envoi du formulaire avec tous les champs laissés vide)

```
<?php
if (array_key_exists('action', $_REQUEST) && $_REQUEST['action'] == 'submitted') {
    print "<h1>Affichage des données saisies 3</h1>";
    print "<ul>";
    print "<li>Nom: " . $_REQUEST['nom'] . "</li>";
    print "<li>Prenom: " . $_REQUEST['prenom'] . "</li>";
    print "<li>Sexe: " . $_REQUEST['sexe'] . "</li>";
    print "<li>Vins: ";
    print "<ul>";
    if (isset($_REQUEST['vin']))
        foreach($_REQUEST['vin'] as $v) print "<li>$v</li>";
    print "</ul>";
    print "</ul>";
    print "<a href='javascript:history.back()'>Essayez à nouveau</a>";
} else {
?>
```

```
<form action="1.6.php" method="get">
    Nom: <input type="text" name="nom"><br>
    Prénom: <input type="text" name="prenom"><br>
    Sexe: <select name="sexe">
              <option>M
              <option>F
          </select><br>
    Vins: <br>
          <select multiple name="vin[]">
              <option value="bordeaux">bordeaux</option>
              <option value="beaujolais">beaujolais</option>
              <option value="côte de provence">côte de provence</option>
          </select><br>
    <input type="hidden" name="action" value="submitted">
    <input type="submit" value="submit Me!">
</form>
<?php
}
?>
```

1.7.Créez un formulaire 1.7.html comportant une zone de texte, et ensuite créez le script 1.7.php qui renvoie le texte saisi en remplaçant les caractères nouvelle ligne par des balises
.
(utiliser la fonction **nl2br** de PHP).

```
<html><body>
   <form action="1.7.php" method="get">
   <textarea rows="20" cols="40" name="texte">vos commentaires</textarea>
   <br><input type="submit" value="Envoyer">
   </form>
</body></html>
```

```
<!-- Fichier 1.7.php -->
<html><body>
<h1>Affichage des données saisies</h1>
   <?php     echo nl2br($_REQUEST['texte'])  ?>
</body></html>
```

Fonctions et classes

1.8.Définir une classe Personne possédant les propriétés nom, prénom, et une méthode presenter() qui renvoie la chaîne de caractères "je m'appelle " suivie du nom et prénom.

```
// fichier Personne.php
<?php
class Personne {
   // proprietes
   var $nom       ;
   var $prenom    ;

   // constructeur
   function Personne($nom, $prenom) {
        $this->nom     = $nom         ;
        $this->prenom = $prenom  ;
   }

   // methodes
   function presenter() {
        return "je m'appelle $this->nom $this->prenom" ;
   }
}
?>
```

1.9.Créer un programme de test qui instancie 2 personnes, puis affiche leurs descriptions.

```
<html><body>
<?php
   include "./personne.php";

   // instanciation de Personne
   $p1 = new Personne("toto", "jules");
   $p2 = new Personne("tata", "julie");

   // affichage
   echo $p1->presenter()."<br>"  ;
   echo $p2->presenter()."<br>"  ;
?>
</body></html>
```

1.10.Ajoutez dans la classe Personne une propriété date de naissance et une méthode age() renvoyant l'age.

```
<?php
class Personne {
   // proprietes
   var $nom  ;
   var $prenom    ;
   var $date_naiss     ; //sous la forme jj/mm/aa

   // constructeur
   function Personne($nom, $prenom, $date_naiss) {
        $this->nom          = $nom          ;
        $this->prenom       = $prenom ;
        $this->date_naiss   = $date_naiss    ;
   }

   // methodes
   function presenter() {
        $ch = "je m'appelle ".$this->nom." ".$this->prenom.", "    ;
        $ch = $ch."j'ai ".$this->age()." ans"              ;
        return $ch     ;
   }

   function age() {
        $tabdate = split("/", $this->date_naiss);
        $t_naiss = mktime(0,0,0,$tabdate[1],$tabdate[0],$tabdate[2]);
        return (int)( ( time() - $t_naiss ) / (365*24*3600)) ;
   }
}
?>
```

1.11.Tester la classe Personne.

```
<html><body>
<?php
   include "./personne.php";
   $p1 = new Personne("toto", "jules", "2/15/1990"); //instanciation de Personne
   echo $p1->presenter()."<br>"  ;
?>
</body></html>
```

2.Sessions PHP

2.1.Afin de tester les variables de session, commencez par créez la page menu.html ci-dessous :

```
<html><body>
<h1>Test de variables de session</h1>
<a href="ouvrir_session.php">Ouvrir une session</a><br>
<a href="afficher_session.php">Afficher variables de session</a><br>
<a href="detruire_session.php">Détruire variables de session</a><br>
</body></html>
```

2.2.Créer le script ouvrir_session.php qui :

- crée une nouvelle session si aucune session n'existe (un SID est engendré et transmis dans un cookie)
- ou bien restaure la session en cours (connue par son identifiant de session SID)

Pour cela il faudra appeler en début de script la fonction PHP **`session_start()`**.

Afficher ensuite un formulaire permettant de saisir le nom et le prénom de l'utilisateur.
Ces deux informations seront conservées dans des variables de session lorsque le formulaire est envoyé.

```
<?php
    session_start();
?>

<html>
<body>
<h1>Ouvrir une session</h1>
<?php
```

```
    echo "<br>ID session: " . session_id() ;
    echo "<br>Nom session: " . session_name() ;
?>
<hr>
<form action="creer_session.php" method="get">
    Nom: <input type="text" name="nom"><br>
    Prénom: <input type="text" name="prenom"><br>
    <input type="submit" value="Envoyer">
</form>
</body>
</html>
```

2.3.Créer le script créer_session.php qui enregistre les données du formulaire en variables de session

Ce script renverra une page html affichant les variables de session, un lien en bas de page permettra de retourner au menu principal.

```
<?php
    session_start();

    // enregistrer les variables de session
    $_SESSION['nom'] = $_REQUEST['nom'] ;
    $_SESSION['prenom'] = $_REQUEST['prenom'] ;
?>

<html>
<body>
<h1>Variables de session créées</h1>
<?php
    echo "<br>ID session: " . session_id();
    echo "<br>Nom session: " . session_name();
    echo "<br>Nom: " . $_SESSION['nom'];
    echo "<br>Prénom: " . $_SESSION['prenom'];
?>
<br><a href="menu.html">Retour menu</a>
</body>
</html>
```

2.4.Créer le script afficher_session.php affichant les variables de session, vous ajouterez un compteur en variable de session afin d'afficher le nombre de fois que la page a été vue.

```
<?php
    session_start();

    // compte le nombre de fois que la page a été visitée
    if (!isset($_SESSION['compteur'])) {
        $_SESSION['compteur'] = 1;
    } else {
        $_SESSION['compteur']++;
    }

?>

<html>
<body>
<h1>Affichage des variables de session</h1>
<?php
    echo "Vous avez vu cette page " . $_SESSION['compteur'] . " fois<p>";
    echo "<br>ID session: " . session_id();
    echo "<br>Nom session: " . session_name();

    echo "<br>Nom: " . $_SESSION['nom'];
    echo "<br>Prénom: " . $_SESSION['prenom'];
    echo "<hr>";
    print_r($_SESSION);

?>
<br><a href="menu.html">Retour menu</a>
</body>
</html>
```

2.5.Créer le script `detruire_session.php` qui détruit les variables de session, utiliser la fonction PHP **`session_destroy()`**.

```
<?php
    session_start();
    $sid = session_id();
    session_destroy();
?>
```

```
<html>
<body>
<h1>Détruire les variables de session</h1>
<?php
   echo "<br>Variables de la session '$sid' détruites !\n" ;
?>
<br><a href="menu.html">Retour menu</a>
</body>
```

3.Créer une table sous ORACLE

3.1.Dans une fenêtre **telnet** sur nyx, ouvrir une session SQL*Plus sous Oracle avec la commande :

```
$ sqlplus
```

saisir votre nom d'utilisateur et votre mot de passe oracle

3.2.Sous SQL*Plus changer votre mot de passe oracle avec la commande :

```
SQL> GRANT CONNECT TO login IDENTIFIED BY passwd ;
```

3.3.Dans un bloc note rédiger la requête SQL permettant de créer une table ANNUAIRE donnant les numéros de poste téléphonique du personnel en fonction de leurs noms et prénoms.

- La table comportera trois champs : nom, prénom, numéro de poste, déclarés en VARCHAR2 (*taille*)
- Les numéros de poste sont de la forme 99.99
- La clé primaire sera composée des deux champs nom et prénom
- Les champs nom et prénom seront déclarés NOT NULL

```
CREATE TABLE annuaire (
   nom       VARCHAR2 (30)  NOT NULL  ,
   prenom    VARCHAR2 (30)  NOT NULL  ,
   no_poste  VARCHAR2 (5)             ,
   PRIMARY KEY (nom, prenom)
) ;
```

3.4.A l'aide d'un Copier/Coller, introduire la requête SQL sous SQL*Plus et l'exécuter.

3.5.Vérifier la structure de votre table ANNUAIRE sous SQL*Plus

```
SQL> describe annuaire
```

3.6.Insérer quelques tuples dans votre table ANNUAIRE, tester.

```
SQL> INSERT INTO annuaire VALUES ('X', 'x', '82.19') ;
SQL> INSERT INTO annuaire VALUES ('Y', 'y', '82.20') ;
SQL> INSERT INTO annuaire VALUES ('Z', 'z', '82.21') ;
SQL> SELECT * FROM annuaire ;
```

3.7.Donner le droit de consultation publique sur votre table ANNUAIRE

```
SQL> GRANT select ON annuaire TO PUBLIC ;
```

4.Créer les pages PHP pour consulter et mettre à jour la table annuaire

4.1.Créer un script php permettant d'effectuer à partir de pages web, les opérations suivantes :

1. Rechercher une entrée dans la table annuaire à partir des premières lettres du nom et/ou du prénom
2. Ajouter une nouvelle entrée dans l'annuaire
3. Modifier un numéro de poste
4. Supprimer une entrée de l'annuaire

Vous utiliserez une feuille de style pour la présentation des pages web.

Feuille de style annuaire.css

```
<style type="text/css">
   zz { color: red }

   hr { color: red }

   body {
        font-family: sans-serif ; background-color: yellow
   }

   a {
        text-decoration: underline;
        color: #000000
   }

   a:hover {
        text-decoration: none;
        color: Red
   }

   table.2 {
        background-color: #CCCCCC;
        border: solid 1pt #0000FF

   }

   table.2 td {
        border: solid 1pt #FFFFFF
   }

   td.1 {
        text-align: right
   }

   input {
        text-decoration: none;
        color: #000000;
        background-color : #FFFFFF;
        border : 1px solid #000000;
   }

   submit {
        text-decoration: none;
        color: #000000;
        background-color : #FFFFFF;
        border : 1px solid #000000;
   }
</style>
```

Script annuaire.php

```
<html>
<head>
   <title>Annuaire Téléphonique</title>
   <link rel="stylesheet" href="annuaire.css" type="text/css">
</head>
<body>

<?php

   $user="et";
   $pass="oraet";
   $db="iutinfo";

   $conn = OCILogon($user,$pass,$db) ;
   if (!$conn) {
        echo "Impossible de se connecter";
        exit;
   };


   echo "<table align='center'><tr><td>
        <h2>Annuaire Téléphonique</h2>
        <li><a href='annuaire.php?task=1'>ajouter une entrée dans la base</a></li><br>
        <li><a href='annuaire.php?task=2'>recherche dans la base</a></li>
        </td></tr></table>
        <hr>";
```

```
   $task=$_REQUEST['task'];
   switch($task) {


        case 1: // ajouter une entrée dans la base
             if (array_key_exists('op1', $_REQUEST)) {
                  $nom=$_REQUEST['nom'];
                  $prenom=$_REQUEST['prenom'];
                  $no_poste=$_REQUEST['no_poste'];
                  $query="INSERT INTO et.annuaire VALUES
('$nom','$prenom','$no_poste')";
                  $stmt = OCIParse($conn, $query) ;
                  $res = OCIExecute($stmt, OCI_COMMIT_ON_SUCCESS) ;
                  if ($res) {
                       echo "<center><h4>Entrée ajoutée</h4></center>";
                  } else {
                       echo "<center><h4>Erreur</h4></center>";
                  };
             };

             echo "<table class='2' cellpadding='5' align='center'>
                  <form name='ajouter' action='annuaire.php?task=1' method='post'>
                  <tr>
                       <td colspan='2'><h2>Ajouter une entrée</h2></td>
                  </tr>
                  <tr>
                       <td class='1'>Nom</td>
                       <td><input type='text' name='nom'></td>
                  </tr>
                  <tr>
                       <td class='1'>Prénom</td>
                       <td><input type='text' name='prenom'></td>
                  </tr>
                  <tr>
                       <td class='1'>Poste</td>
                       <td><input type='text' name='no_poste'></td>
                  </tr>
                  <tr>
                       <td> </td>
                       <td><input type='submit' name='op1' value='Ajouter'></td>
                  </tr>
                  </form>
                  </table>";
             break;

        case 2: // recherche dans la base
             if (array_key_exists('op2', $_REQUEST)) {
                  $nom=$_REQUEST['nom'];
                  $prenom=$_REQUEST['prenom'];
                  $query="SELECT * FROM et.annuaire WHERE nom IS NOT NULL";
                  if ($nom) {
                       $query.=" AND nom LIKE '$nom%'";
                  };
                  if ($prenom) {
                       $query.=" AND prenom LIKE '$prenom%'";
                  };
                  $query.=" ORDER BY nom, prenom ";
                  $stmt = OCIParse($conn, $query);
                  OCIExecute($stmt);

                  echo "<table class='2' cellpadding='5' cellspacing='1'
align='center'>";
                  echo "<tr>";
                  echo
"<th>Nom</th><th>Prénom</th><th>Poste</th><th> </th><th> </th>";
                  echo "</tr>\n";

                  $nbrows=0;
                  while ( OCIFetchInto($stmt,&$data,OCI_ASSOC) ) {
                       echo "<tr>";
                       $nom  = $data["NOM"];
                       $prenom  = $data["PRENOM"];
                       $no_poste = $data["NO_POSTE"];
                       echo "<td>$nom</td>";
                       echo "<td>$prenom</td>";
                       echo "<td>$no_poste</td>";
                       echo "<td><a
href='annuaire.php?task=4&nom=$nom&prenom=$prenom&no_poste=$no_poste'>modifier</a></td>
";
```

```
                    echo "<td><a
href='annuaire.php?task=3&nom=$nom&prenom=$prenom'>effacer</a></td>";
                    echo "</tr>";
                    $nbrows++;
               }
               echo "<tr><td colspan='5'>Nbre lignes trouvées: $nbrows</td></tr>";
               echo "</table>";
               OCIFreeStatement($stmt);
          };

          echo "<table class='2' cellpadding='5' align='center'>
               <form name='recherche' action='annuaire.php?task=2' method='post'>
               <tr>
                    <td colspan='2'><h2>Rechercher</h2></td>
               </tr>
               <tr>
                    <td class='1'>Nom</td>
                    <td><input type='text' name='nom'></td>
               </tr>
               <tr>
                    <td class='1'>Prénom</td>
                    <td><input type='text' name='prenom'></td>
               </tr>
               <tr>
                    <td> </td>
                    <td><input type='submit' name='op2' value='Rechercher'></td>
               </tr>
               </form>
               </table>";
          break;

     case 3: // confirmation suppression
          $nom=$_REQUEST['nom'];
          $prenom=$_REQUEST['prenom'];
               echo "<table align='center'><tr>
               <td>Vous êtes sûr ?
               <li><a
href='annuaire.php?task=5&nom=$nom&prenom=$prenom'>OUI</a></li>
               <li><a href='javascript:history.back();'>Annuler</a></li>
               </td></tr></table>";
          break;

     case 4: // modifier no de poste
          if (array_key_exists('op4', $_REQUEST)) {
               $nom=$_REQUEST['nom'];
               $prenom=$_REQUEST['prenom'];
               $no_poste=$_REQUEST['no_poste'];
               $query="UPDATE et.annuaire ";
               $query.="SET no_poste='$no_poste' ";
               $query.="WHERE nom='$nom' AND prenom='$prenom'";
               $stmt = OCIParse($conn, $query);
               $res = OCIExecute($stmt, OCI_COMMIT_ON_SUCCESS);
               if ($res) {
                    echo "<center><h4>Entrée modifiée</h4></center>";
               } else {
                    echo "<center><h4>Erreur</h4></center>";
               };
          };

          echo "<table class='2' cellpadding='5' align='center'>
               <form name='modifier' action='annuaire.php?task=4' method='post'>
               <tr>
                    <td class='m' colspan='2'><h2>Modifier</h2></td>
               </tr>
               <tr>
                    <td class='g'>Poste</td>
                    <td><input type='text' name='no_poste' value=$no_poste></td>
               </tr>
               <tr>
                    <td> </td>
                    <td><input type='hidden' name='nom' value='$nom'>
                    <input type='hidden' name='prenom' value='$prenom'>
                    <input type='submit' name='op4' value='Modifier'></td>
               </tr>
               </form>
               </table>";
          break;

     case 5:
          $query="DELETE FROM et.annuaire ";
```

```
            $query.="WHERE nom = '$nom' AND prenom='$prenom'";
            $stmt = OCIParse($conn, $query);
            $res = OCIExecute($stmt, OCI_COMMIT_ON_SUCCESS);
            if ($res) {
                 echo"<center><h4>Entrée supprimée</h4></center>";
            } else {
                 echo"<center><h4>Erreur</h4></center>";
            };
            break;

       default:
            echo "<center><h3>HELLO ;o)</h3></center>";
            break;
   }
   OCILogOff($conn);
?>

</body>
</html>
```

5.Créer des menus déroulants à partir de données puisées dans des tables

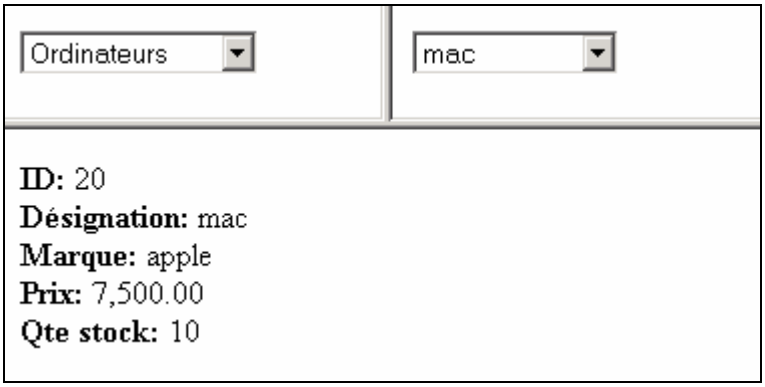
Nous considérons les deux relations suivantes :

- CATEGORIE (id_cat, designation, taux_tva)
une ligne représente une catégorie de produit (clé id_cat)

- PRODUIT (id_pro, id_cat, designation, marque, prix_uht, qstock)
une ligne représente un produit (clé id_pro)
un produit appartient à une seule catégorie (clé étrangère id_cat référence une catégorie)

Un jeu d'essai est créé sous oracle dans le compte ET (le script ayant permis de créer les tables et le jeu d'essai est donné en Annexe2). Les tables sont accessibles en lecture pour tous les utilisateurs d'oracle.

Nous voulons créer une page comportant trois cadres comme dans l'exemple ci-dessous :



- le cadre en haut à gauche contient un menu permettant de sélectionner une catégorie
- le cadre en haut à droite contient un deuxième menu permettant de sélectionner un produit de la catégorie qui a été sélectionnée à gauche
- le cadre en bas affiche le produit sélectionné

5.1.Créer le script menu.php qui découpe la fenêtre en trois cadres.

```
<! fichier menu.php -->
<frameset rows="60,*">
  <frameset cols="20%,*">
    <frame src="menu1.php" name="menu1">
    <frame src="menu2.php?id_cat" name="menu2">
  </frameset>
  <frame src="page_vierge.php" name="fiche">
</frame>
```

5.2.Créer le script menu1.php qui affiche dans le cadre en haut à gauche le menu permettant de sélectionner une catégorie.
Lorsque l'utilisateur sélectionne une catégorie, une fonction javascript permettra d'envoyer au serveur une requête menu2.php avec en paramètre l'identifiant de la catégorie sélectionnée. Ce script menu2.php renverra dans le cadre à droite le menu permettant de sélectionner un produit de la catégorie passée en paramètre.

```
<! fichier connexion.php -->
<?php
$user="scott";
$pass="tiger";
$bd="iutinfo.unice.fr";
if(!$conn=ociplogon($user,$pass,$bd)) {
    echo "Impossible de se connecter";
    exit;
}
?>

<! fichier menu1.php -->
<html>
<head>
<script language="javascript">
    function ouvrir_menu2(val){
        adresse = "menu2.php?id_cat="+val ;
        open(adresse,"menu2") ;
    }
</script>
</head>
<body>
    <select name="menu1" onchange="ouvrir_menu2(value)">
    <option value="" selected>Choix catégorie</option>
    <?php
    require "./connexion.php";
    $reqSQL="SELECT id_cat, designation FROM ET.categorie ORDER BY id_cat";
    $stmt=ociparse($conn, $reqSQL);

    ociexecute($stmt);
    while(ocifetchinto($stmt, &$ligne, OCI_ASSOC)){
      $id_cat = $ligne["ID_CAT"];
      $designation = $ligne["DESIGNATION"];
      echo "<option value=\"$id_cat\">$designation</option>\n";
    }
    ?>
    </select>
</body>
</html>
```

5.3.Créer le script menu2.php qui renvoie dans le cadre à droite le menu permettant de sélectionner un produit de la catégorie qui est passée en paramètre.
Lorsque l'utilisateur sélectionne un produit, une fonction javascript permettra d'envoyer au serveur une requête fiche.php avec en paramètre l'identifiant du produit sélectionné. Ce script fiche.php renverra dans le cadre en bas la fiche du produit sélectionné.

```
<! fichier menu2.php -->
<html>
<head>
<script language="javascript">
    function afficher(val){
        adresse = "fiche.php?id_pro="+val;
        open(adresse,"fiche") ;
    }
</script>
</head>
<body>
    <select name="menu1" onchange="afficher(value)">
    <option value="" selected>Choix produit</option>
    <?php
    require "./connexion.php";
    $reqSQL="SELECT id_pro, designation ";
    $reqSQL.="FROM et.produit ";
    $reqSQL.="WHERE id_cat = '" . $_REQUEST['id_cat'] . "' ";
    $reqSQL.="ORDER BY id_pro";
    $stmt=ociparse($conn, $reqSQL);
    ociexecute($stmt);
    while(ocifetchinto($stmt, &$ligne, OCI_ASSOC)){
      $id_pro = $ligne["ID_PRO"];
      $designation = $ligne["DESIGNATION"];
      echo "<option value=\"$id_pro\">$designation</option>\n";
```

```
    }
    ?>
    </select>
</body>
</html>
```

5.4.Créer le script fiche.php qui renvoie dans le cadre en bas la fiche du produit sélectionné.

```
<! fichier fiche.php -->
<html>
<body>
    <?php
    require "./connexion.php";
    $reqSQL="SELECT id_pro, designation, marque, prix_uht, qstock ";
    $reqSQL.="FROM et.produit ";
    $reqSQL.="WHERE id_pro = '" . $_REQUEST['id_pro'] . "' ";
    $reqSQL.="ORDER BY id_pro";
    $stmt=ociparse($conn, $reqSQL);
    ociexecute($stmt);
    while(ocifetchinto($stmt, &$ligne, OCI_ASSOC)){
        echo "<b>ID: </b>" . $ligne["ID_PRO"] . "<br>\n";
        echo "<b>Désignation: </b>" . $ligne["DESIGNATION"] . "<br>\n";
        echo "<b>Marque: </b>" . $ligne["MARQUE"] . "<br>\n";
        echo "<b>Prix: </b>" . number_format($ligne["PRIX_UHT"],2) . "<br>\n";
        echo "<b>Qte stock: </b>" . $ligne["QSTOCK"] . "<br>\n";
    }
    ?>
</body>
</html>
```

6.Annexe1 – Fonctions OCI (*Oracle Call Interface*) de PHP

Etablit une connexion à un serveur Oracle
resource ocilogon (string **username**, string **password**, string **db**)

ocilogon() retourne un identifiant de connexion, nécessaire à la plus part des fonctions OCI. Si l'option ORACLE_SID n'est pas précisée, PHP utilisera la variable d'environnement ORACLE_SID pour déterminer le serveur de connexion.

Les connexions sont partagées, à l'intérieur d'une même page avec ocilogon(). Cela signifie que COMMIT et ROLLBACK s'appliquent à toutes les transactions commencées à l'intérieur d'une même page, même si vous avez créé de multiples connexions.

Déconnexion d'un serveur Oracle
int ocilogoff (resource **connection**)

ocilogoff() ferme la connexion Oracle.

Analyse une requête
int ociparse (ocifreedesc **conn**, strint **query**)

ociparse() analyse la requête **query** sur la connexion **conn**, et retourne TRUE si la requête **query** est valide, et FALSE, si ce n'est pas le cas. **query** peut être n'importe quelle requête SQL.

Exécute une commande
int ociexecute (resource **statement**, int **mode**)

ociexecute() éxécute une commande déjà préparée (voir ociparse()). L'option **mode** vous permet de spécifier le mode d'exécution (par défaut, il est à OCI_COMMIT_ON_SUCCESS). Si vous ne voulez pas que la commande soit automatiquement validée, utilisez le mode OCI_DEFAULT.

Retourne le nombre de lignes affectées
int ocirowcount (resource **statement**)

ocirowcount() retourne le nombre de lignes affectées par une commande de modification. Cette fonction ne vous indiquera pas le nombre de lignes retournées par un SELECT : il faut que les lignes aient été modifiées.

Retourne la valeur d'une colonne dans une ligne lue
mixed ociresult (resource **statement**, mixed **column**)

ociresult() retourne les données de la colonne **column** dans la ligne courante (voir ocifetch()). ocifetch()retournera tout les types, sauf les types abstraits (ROWIDs, LOBs et FILEs).

Modifie la prochaine ligne dans le pointeur interne de résultat.
int ocifetch (resource **statement**)

ocifetch() place la prochaine ligne (d'une commande SELECT) dans le pointeur interne de résultat.

Retourne la ligne suivante dans un tableau
int ocifetchinto (resource **stmt**, array &**result**, int **mode**)

ocifetchinto() retourne la ligne suivante (pour une commande SELECT) dans le tableau **result**. ocifetchinto() écrasera le contenu de **result**. Par défaut, **result** sera un tableau à index numérique, commençant à 1, et qui contiendra toute les colonnes qui ne sont pas NULL.

L'option **mode** vous permet de modifier le comportement par défaut de la fonction. Vous pouvez passer plusieurs modes simplement en les additionnant (i.e. OCI_ASSOC+OCI_RETURN_NULLS). Les modes valides sont :

- OCI_ASSOC Retourne un tableau associatif.
- OCI_NUM Retourne un tableau à index numérique (DEFAULT, valeur par défaut)
- OCI_RETURN_NULLS Retourne les colonnes vides.
- OCI_RETURN_LOBS Retourne la valeur des objets LOB plutôt que leur descripteur.

Retourne toutes les lignes d'un résultat
int ocifetchstatement (resource **stmt**, array &**variable**)

ocifetchstatement() retourne toutes les lignes d'un résultat dans le tableau variable. ocifetchstatement() retourne le nombre de lignes retournées.

Teste si la valeur d'une colonne est NULL
int ocicolumnisnull (resource **stmt**, mixed **column**)

ocicolumnisnull() retourne TRUE si la colonne **col** du résultat **stmt** est NULL. Vous pouvez utiliser le numéro de colonne (l'indexation des colonnes commence à 1) ou le nom de la colonne, pour le paramètre **col**.

Valide les transactions en cours
int ocicommit (resource **connection**)

ocicommit() valide toutes les transactions en cours sur la connexion Oracle **connection**.

Annule les transactions en cours
int ocirollback (resource **connection**)

ocirollback() annule les transactions en cours sur la connexion Oracle **connection**.

Libère toutes les ressources occupées par une commande.
int ocifreestatement (resource **stmt**)

ocifreestatement() retourne TRUE en cas de succès, et FALSE en cas d'échec.

Retourne la dernière erreur de stmt|conn|global.
array ocierror (int **stmt)conn**|

ocierror() retourne la dernière erreur trouvée. Si l'option **stmt|conn** n'est pas fournie, la dernière erreur rencontrée est retournée. Si aucune erreur n'est trouvée, ocierror() retourne FALSE.

7.Annexe2 - jeu d’essai (compte ET sous oracle)

```
DROP TABLE produit;
DROP TABLE categorie;

CREATE TABLE categorie
(
   id_cat         VARCHAR2 (2)   PRIMARY KEY ,
   designation    VARCHAR2 (50)              ,
   taux_tva       NUMBER (4,2)
) ;

CREATE TABLE produit
(
   id_pro         VARCHAR2 (8)   PRIMARY KEY               ,
   id_cat         VARCHAR (2)    REFERENCES categorie      ,
   designation    VARCHAR2 (50)                            ,
   marque         VARCHAR2 (50)                            ,
   prix_uht       NUMBER (9,2)                             ,
   qstock         NUMBER (2)
) ;

/* jeu d'essai */
INSERT INTO categorie VALUES ('C1', 'Ordinateurs', 10) ;
INSERT INTO categorie VALUES ('C2', 'Logiciels', 20) ;
INSERT INTO categorie VALUES ('C3', 'Imprimantes', 30) ;

INSERT INTO produit VALUES ('10','C1','ps','ibm',500.00,10);
INSERT INTO produit VALUES ('20','C1','mac','apple',7500.00,10);
INSERT INTO produit VALUES ('30','C1','aptiva','ibm',12000.00,10);
INSERT INTO produit VALUES ('40','C1','power mac','apple',2000.00,10);
INSERT INTO produit VALUES ('50','C2','word','microsoft',100.00,10);
INSERT INTO produit VALUES ('60','C2','access','microsoft',100.00,10);
INSERT INTO produit VALUES ('70','C2','paradox','borland',80.00,30);
```

Tables sous MySql

```
CREATE TABLE IF NOT EXISTS `annuaire` (
  `id_ann` int(11) NOT NULL auto_increment,
  `nom` varchar(50) NOT NULL,
  `prenom` varchar(35) NOT NULL,
  `no_poste` varchar(5),
  PRIMARY KEY  (`id_ann`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=3 ;

--
-- Contenu de la table `annuaire`
--

INSERT INTO `dept` VALUES (1, 'Dupont', 'Jean', '82.19');
INSERT INTO `dept` VALUES (2, 'Martin', 'Paul', '81.12');
-- --------------------------------------------------------

--
-- Structure de la table `categorie`
--

CREATE TABLE IF NOT EXISTS 'categorie' (
```

```
  `id_cat` int(11) NOT NULL auto_increment,
  `designation` varchar(30) NOT NULL,
  `tva` float NOT NULL,
  PRIMARY KEY  (`id_cat`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=11 ;

--
-- Structure de la table `produit`
--
CREATE TABLE IF NOT EXISTS 'produit' (
  `id_pro` int(11) NOT NULL auto_increment,
  `id_cat` int(11) REFERENCES categorie,
  `designation` varchar(30),
  `marque` varchar(30),
  `prix_uht` float,
 `qstock` int,
  PRIMARY KEY  (`id_pro`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=11 ;
```