**Realisation D'un Site Web Complet**

Pré-Requis

flécheMaîtrise des langages HTML et CSS (recommandé).  
flécheBonnes connaissances du langage SQL (recommandé).

Objectifs

flécheCréer un site web « 2en1 » avec une partie cliente (front) et une interface de gestion (back).   
flécheDéveloppement de fonctionnalités. (CRUD orienté e-commerce)

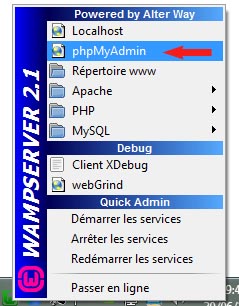
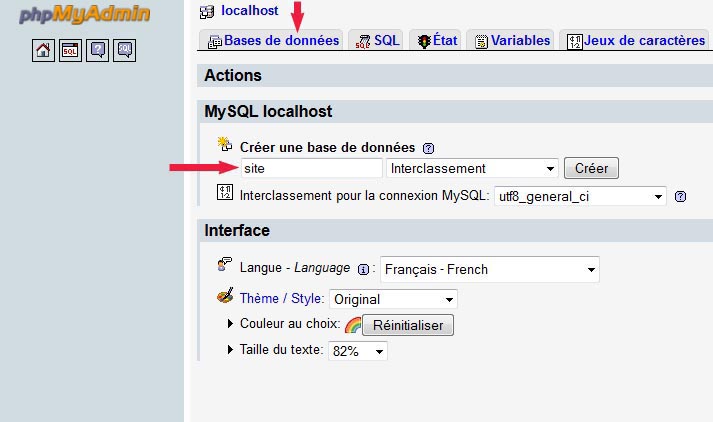
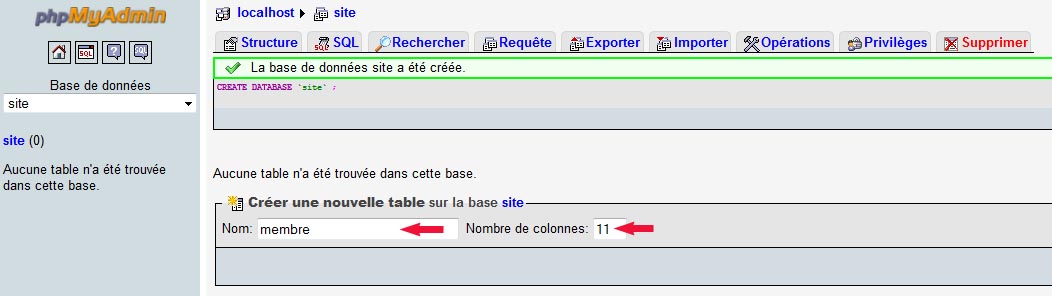
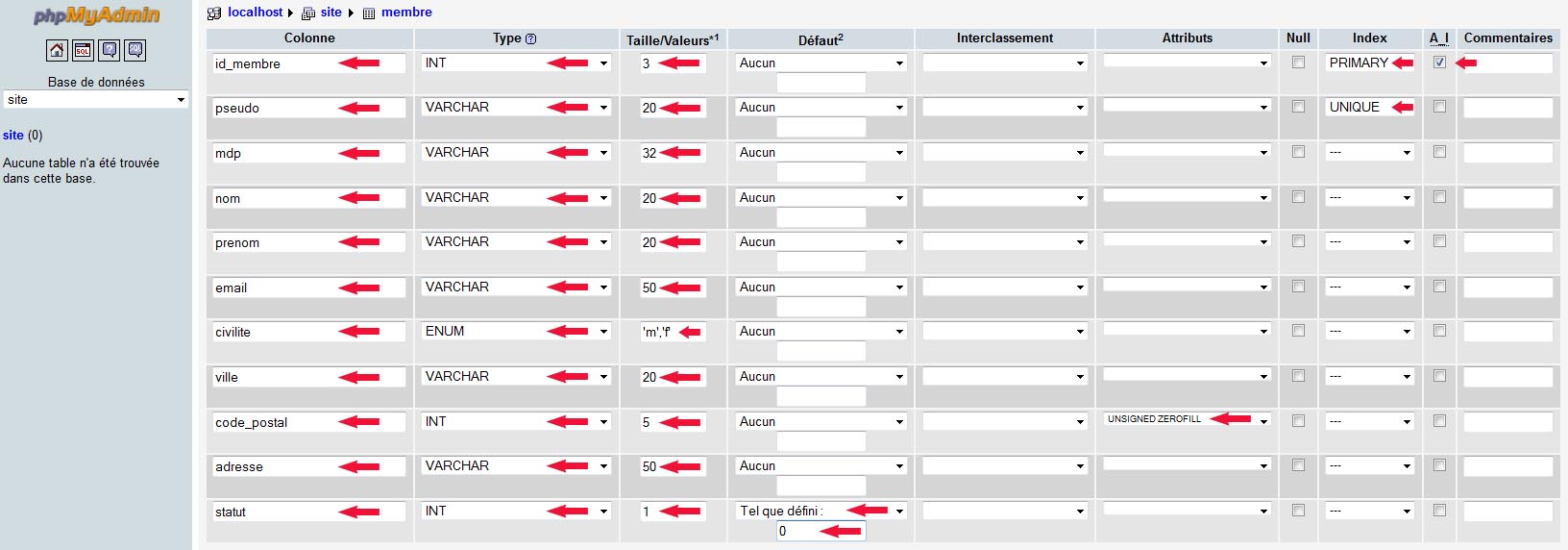
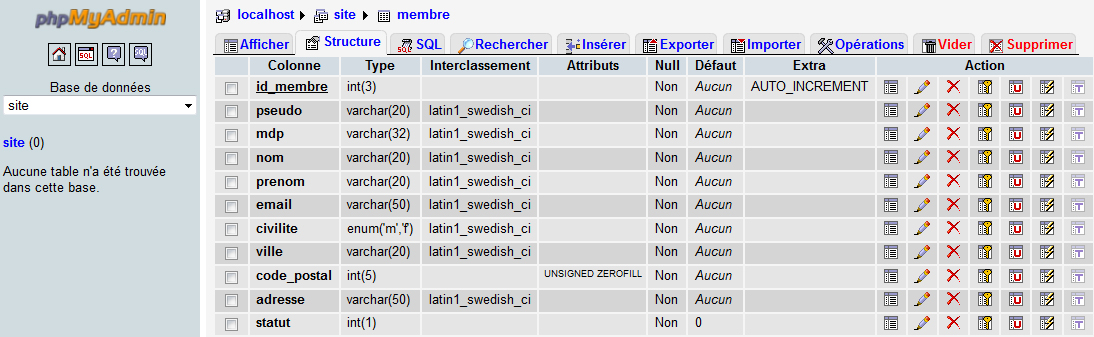
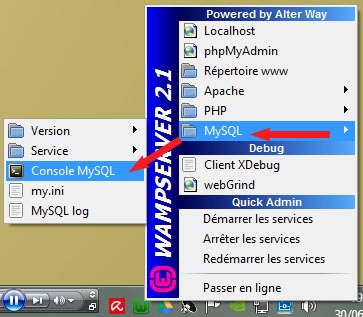
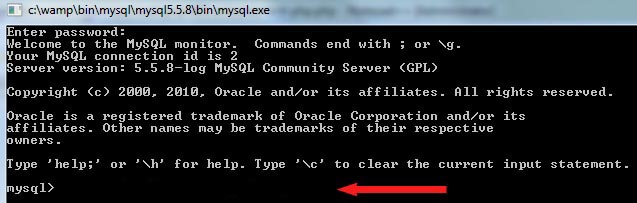
**Réalisation d'un site web (CRUD), tendance e-commerce**

Ce site aura le mérite de nous faire pratiquer le CRUD, **C**reate **R**ead **U**pdate **D**elete.  
  
Le CRUD représente la base en PHP. Il s'agit de savoir librement consulter, insérer, modifier ou supprimer des données.  
  
Sans ça, ne pensez pas mettre le mot "PHP" sur votre CV. *(ou alors il faudra éventuellement écrire « notions php »)*.

**Etape 1 : Modélisation et création de la base de données, table et champs.**

**La table membre**   
  
Base de données : **site** H:\IUSTE\PHP\Eprojet Php_fichiers\icone-bdd.png  
Table : **membre** H:\IUSTE\PHP\Eprojet Php_fichiers\icone-table.png

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Champ** | **Type** | **Taille** | **Spécificité** | **Description** |
| id\_membre | INT | 3 | Clé primaire (PK - Primary Key), AUTO\_INCREMENT (AI) | Ce champ correspond au numéro du membre qui sera auto-généré et incrémenté |
| pseudo | VARCHAR | 20 | UNIQUE | Ce champ correspond au pseudo du membre. Il sera unique et par conséquent 2 membres ne pourront pas avoir le même pseudo. |
| mdp | VARCHAR | 32 | - | Ce champ correspond au mot de passe du membre. La taille fait 32 caractères car il sera crypté par la suite. |
| nom | VARCHAR | 20 | - | Ce champ correspond au nom de famille du membre. |
| prenom | VARCHAR | 20 | - | Ce champ correspond au prénom du membre. |
| email | VARCHAR | 50 | - | Ce champ correspond à l'email du membre. |
| civilite | ENUM | 'm','f' | - | Ce champ correspond à la civilité du membre. Le membre sera soit Homme (M) soit Femme (F). Logiquement il n'y a que 2 choix possibles :p |
| ville | VARCHAR | 20 | - | Ce champ correspond à la ville du membre. |
| code\_postal | INT | 5 | UNSIGNED ZEROFILL | Ce champ correspond au code postal du membre. |
| adresse | VARCHAR | 50 | - | Ce champ correspond à l'adresse du membre. |
| statut | INT | 1 | DEFAULT 0 | Ce champ correspond au statut du membre. Par défaut il sera à zéro (ce qui correspondra à 1 membre). Nous pourrons mettre le chiffre 1 pour donner des droits d'administration à certains membres (1 admin est aussi 1 membre). |

**Pour créer cette base de données, vous pouvez également passer par le gestionnaire de base de données PhpMyAdmin :**  
Accès à PhpMyAdmin :  
  
  
Création d'une nouvelle base de données :   
  
  
Création d'une nouvelle table :  
  
  
Création de la structure de la table (champs/colonnes) :  
  
  
Structure de la table (relecture) :  
  
  
  
  
**Pour créer cette base de données, vous pouvez également passer par la console Mysql :**  
  
  
  
  
Voici le code à insérer :

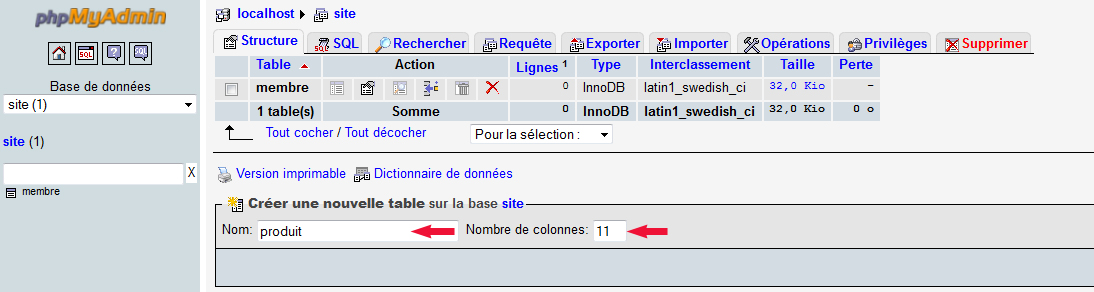
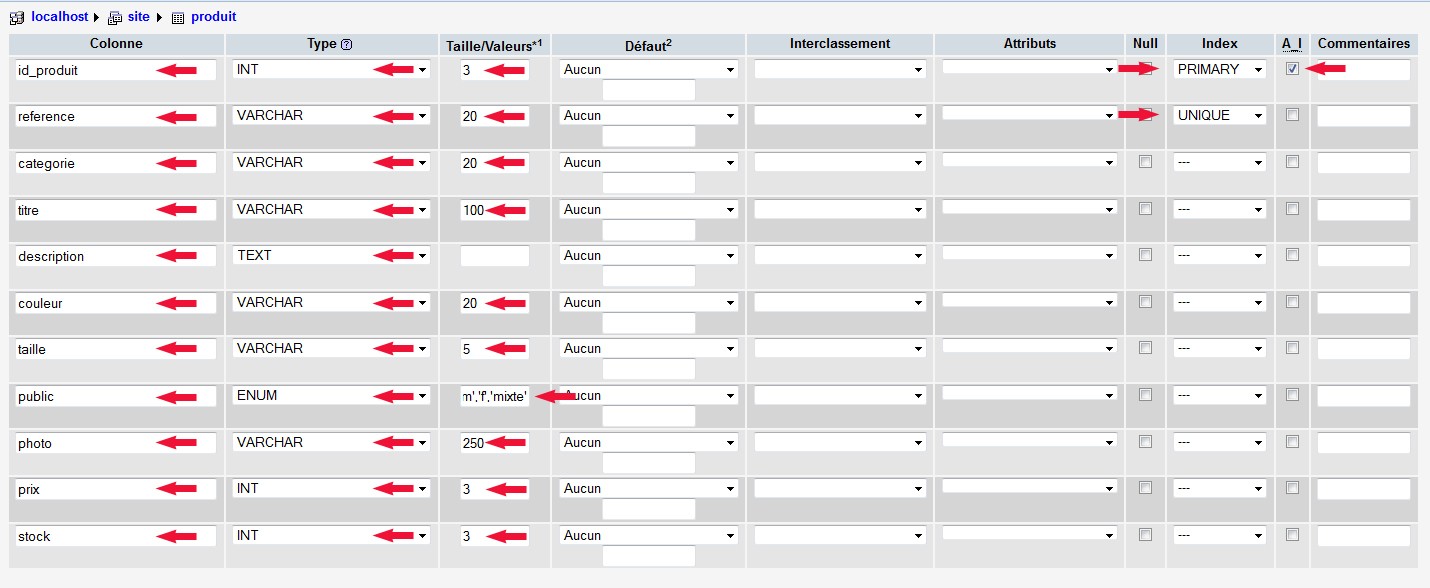
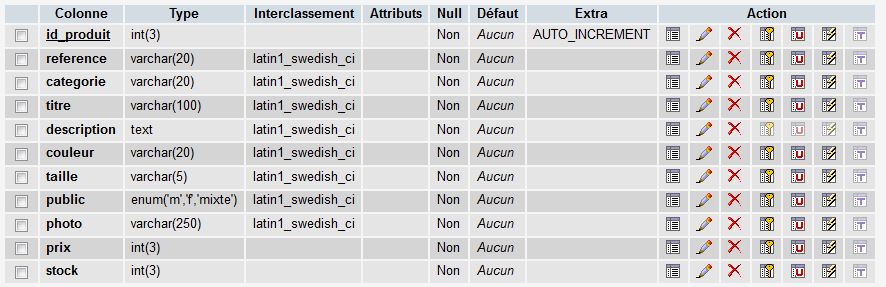
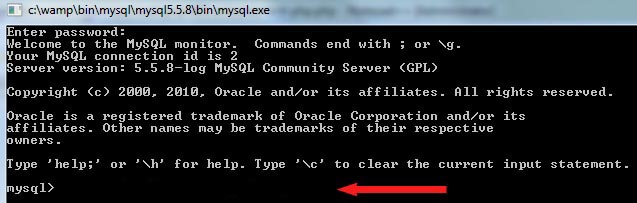
Base de données site - Table membre

[?](file:///H:\IUSTE\PHP\Eprojet%20Php.htm)

|  |
| --- |
| CREATE DATABASE site ;    USE site ;    CREATE TABLE membre (      id\_membre INT(3) NOT NULL AUTO\_INCREMENT PRIMARY KEY ,      pseudo VARCHAR(20) NOT NULL ,      mdp VARCHAR(32) NOT NULL ,      nom VARCHAR(20) NOT NULL ,      prenom VARCHAR(20) NOT NULL ,      email VARCHAR(50) NOT NULL ,      civilite ENUM('m', 'f') NOT NULL ,      ville VARCHAR(20) NOT NULL ,      code\_postal INT(5) UNSIGNED ZEROFILL NOT NULL ,      adresse VARCHAR(50) NOT NULL ,      statut INT(1) NOT NULL DEFAULT 0,      UNIQUE (pseudo)  ) ENGINE = InnoDB; |

**La table produit**   
  
Base de données : **site** H:\IUSTE\PHP\Eprojet Php_fichiers\icone-bdd.png  
Table : **produit** H:\IUSTE\PHP\Eprojet Php_fichiers\icone-table.png

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Champ** | **Type** | **Taille** | **Spécificité** | **Description** |
| id\_produit | INT | 3 | Clé primaire (PK - Primary Key), AUTO\_INCREMENT (AI) | Ce champ correspond au numéro du produit qui sera auto-généré et incrémenté |
| reference | VARCHAR | 20 | UNIQUE | Ce champ correspond à la référence du produit. Il sera unique et par conséquent 2 produits ne pourront pas avoir la même référence. |
| categorie | VARCHAR | 20 | - | Ce champ correspond a la catégorie du produit |
| titre | VARCHAR | 100 | - | Ce champ correspond au titre du produit. |
| description | TEXT | - | - | Ce champ correspond a la description du produit. |
| couleur | VARCHAR | 20 | - | Ce champ correspond à la couleur du produit. |
| taille | VARCHAR | 5 | - | Ce champ correspond à la taille du produit. |
| public | ENUM | 'm','f', 'mixte' | - | Ce champ permettra de determiner à quel public est destiné ce produit. Les choix possibles sont Homme (M), soit Femme (F) ou mixte (mixte). |
| photo | VARCHAR | 250 | - | Ce champ correspond au chemin de la photo qui sera enregistré pour représenté le produit. Ce ne sera pas le fichier image directement mais bien son chemin qui sera enregistré. |
| prix | INT | 3 | - | Ce champ correspond au prix du produit. |
| stock | INT | 3 | - | Ce champ correspond au stock restant du produit. |

**Pour créer cette base de données, vous pouvez également passer par le gestionnaire de base de données PhpMyAdmin :**  
Création d'une nouvelle table :  
  
  
Création de la structure de la table (champs/colonnes) :  
  
  
Structure de la table (relecture) :  
  
  
  
  
**Pour créer cette base de données, vous pouvez également passer par la console Mysql :**  
  
  
  
Voici le code à insérer :

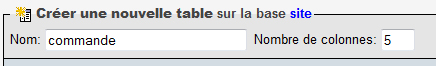
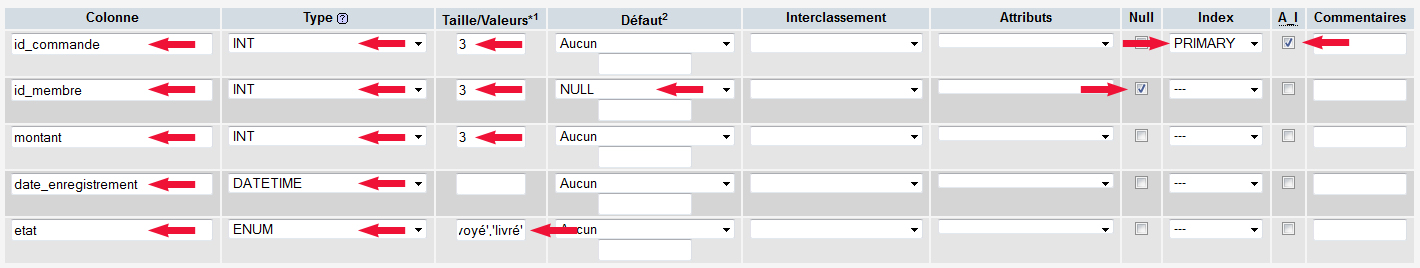
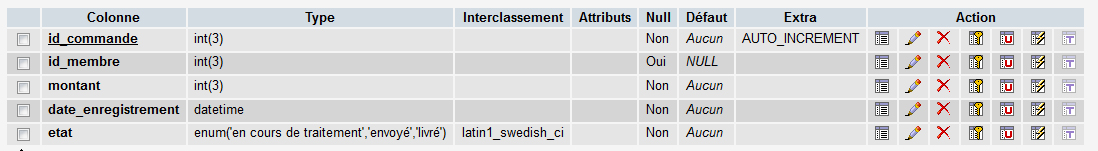
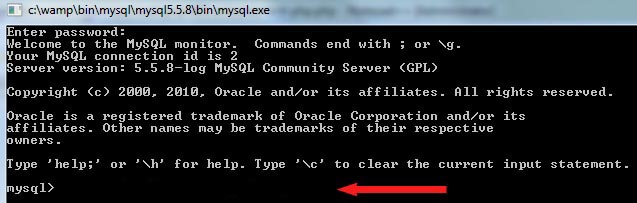
Base de données site - Table produit

[?](file:///H:\IUSTE\PHP\Eprojet%20Php.htm)

|  |
| --- |
| CREATE TABLE produit (      id\_produit INT(3) NOT NULL AUTO\_INCREMENT PRIMARY KEY ,      reference VARCHAR(20) NOT NULL ,      categorie VARCHAR(20) NOT NULL ,      titre VARCHAR(100) NOT NULL ,      description TEXT NOT NULL ,      couleur VARCHAR(20) NOT NULL ,      taille VARCHAR(5) NOT NULL ,      public ENUM('m', 'f', 'mixte') NOT NULL ,      photo VARCHAR(250) NOT NULL ,      prix INT(3) NOT NULL ,      stock INT(3) NOT NULL ,      UNIQUE (reference)  ) ENGINE = InnoDB; |

**La table commande**   
  
Base de données : **site** H:\IUSTE\PHP\Eprojet Php_fichiers\icone-bdd.png  
Table : **commande** H:\IUSTE\PHP\Eprojet Php_fichiers\icone-table.png

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Champ** | **Type** | **Taille** | **Spécificité** | **Description** |
| id\_commande | INT | 3 | Clé primaire (PK - Primary Key), AUTO\_INCREMENT (AI) | Ce champ correspond au numéro de commande qui sera auto-généré et incrémenté |
| id\_membre | INT | 3 | Clé étrangère (FK - Foreign Key), NULL, DEFAULT NULL | Ce champ correspond à l'id\_membre qui aura commandé. |
| montant | INT | 3 | - | Ce champ correspond au montant total (en euros) de la commande |
| date\_enregistrement | DATETIME | - | - | Ce champ correspond à la date et heure d'enregistrement de la commande. |
| etat | Enum | 'en cours de traitement','envoyé','livré' | DEFAULT 'en cours de traitement' | Ce champ correspond a l'état de commande. |

**Pour créer cette base de données, vous pouvez également passer par le gestionnaire de base de données PhpMyAdmin :**  
Création d'une nouvelle table :  
  
  
Création de la structure de la table (champs/colonnes) :  
  
  
Structure de la table (relecture) :  
  
  
  
  
**Pour créer cette base de données, vous pouvez également passer par la console Mysql :**  
  
  
  
Voici le code à insérer :

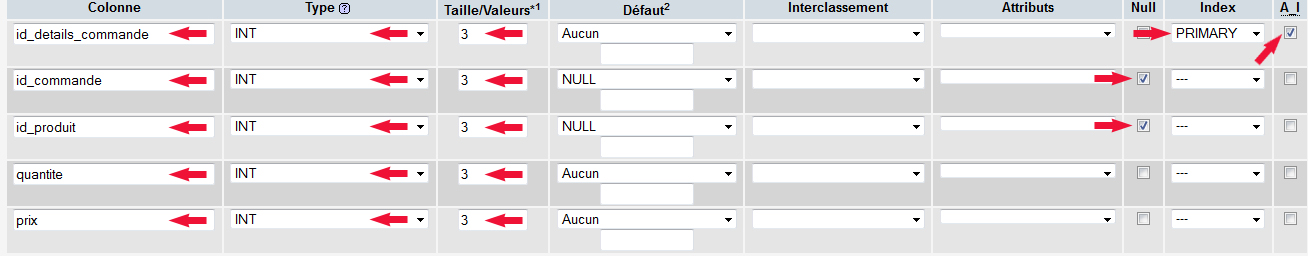
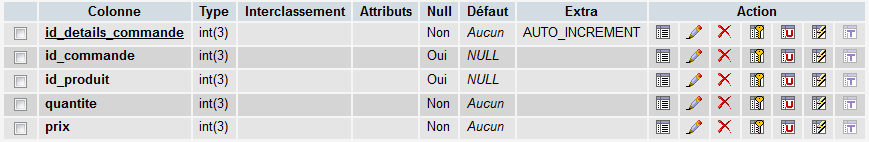
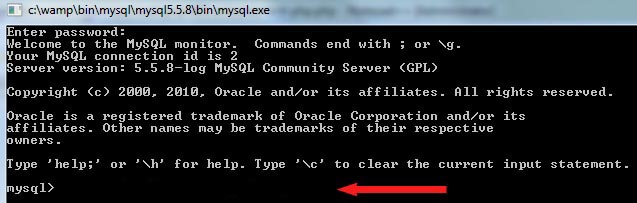
Base de données site - Table commande

[?](file:///H:\IUSTE\PHP\Eprojet%20Php.htm)

|  |
| --- |
| CREATE TABLE commande (      id\_commande INT(3) NOT NULL AUTO\_INCREMENT PRIMARY KEY,      id\_membre INT(3) NULL DEFAULT NULL,      montant INT(3) NOT NULL,      date\_enregistrement DATETIME NOT NULL,      etat ENUM('en cours de traitement', 'envoyé', 'livré') NOT NULL  ) ENGINE = InnoDB; |

**La table details\_commande**   
  
Base de données : **site** H:\IUSTE\PHP\Eprojet Php_fichiers\icone-bdd.png  
Table : **details\_commande** H:\IUSTE\PHP\Eprojet Php_fichiers\icone-table.png

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Champ** | **Type** | **Taille** | **Spécificité** | **Description** |
| id\_details\_commande | INT | 3 | Clé primaire (PK - Primary Key), AUTO\_INCREMENT (AI) | Ce champ correspond au numéro du détails de la commande qui sera auto-généré et incrémentée |
| id\_commande | INT | 3 | Clé étrangère (FK - Foreign Key), NULL, DEFAULT NULL | Ce champ correspond à l'id\_commande a laquelle le détail est rattaché. |
| id\_produit | INT | 3 | Clé étrangère (FK - Foreign Key), NULL, DEFAULT NULL | Ce champ correspond à l'id\_produit qui aura été commandé. |
| quantite | INT | 3 | - | Ce champ correspond à la quantité demandée par produit. |
| prix | INT | 3 | - | Ce champ correspond au prix du produit |

**Pour créer cette base de données, vous pouvez également passer par le gestionnaire de base de données PhpMyAdmin :**  
Création d'une nouvelle table :  
explication PHP  
  
Création de la structure de la table (champs/colonnes) :  
  
  
Structure de la table (relecture) :  
  
  
  
  
**Pour créer cette base de données, vous pouvez également passer par la console Mysql :**  
  
  
  
Voici le code à insérer :

Base de données site - Table details\_commande

[?](file:///H:\IUSTE\PHP\Eprojet%20Php.htm)

|  |
| --- |
| CREATE TABLE details\_commande (      id\_details\_commande INT(3) NOT NULL AUTO\_INCREMENT PRIMARY KEY,      id\_commande INT(3) NULL DEFAULT NULL,      id\_produit INT(3) NULL DEFAULT NULL,      quantite INT(3) NOT NULL,      prix INT(3) NOT NULL  ) ENGINE = InnoDB; |

Avec cette table nous pourrons avoir 1 commande (dans la table commande) comprenant par exemple 5 produits (dans la table details\_commande, pour cette même commande).  
  
Comme il s'agit d'un entrainement, nous reviendrons sur les clés étrangères et les contraintes d'intégrité un peu plus tard afin d'en parler en détail.  
  
Bien entendu, toutes les tables sont simplifiées, dans le cadre d'un vrai site ecommerce nous pourrions avoir des dizaines de tables.

**Etape 2. Création d'une arborescence**

Nous allons créer notre arborescence de site web :   
  
- H:\IUSTE\PHP\Eprojet Php_fichiers\icone-dossier.png/site/  
------- H:\IUSTE\PHP\Eprojet Php_fichiers\icone-dossier.png/photo/  
------- H:\IUSTE\PHP\Eprojet Php_fichiers\icone-dossier.png/admin/  
------- ------- H:\IUSTE\PHP\Eprojet Php_fichiers\icone-fichier.pnggestion\_boutique.php  
------- ------- H:\IUSTE\PHP\Eprojet Php_fichiers\icone-fichier.pnggestion\_commande.php  
------- ------- H:\IUSTE\PHP\Eprojet Php_fichiers\icone-fichier.pnggestion\_membre.php  
------- H:\IUSTE\PHP\Eprojet Php_fichiers\icone-dossier.png/inc/  
------- ------- H:\IUSTE\PHP\Eprojet Php_fichiers\icone-dossier.png/img/  
------- ------- H:\IUSTE\PHP\Eprojet Php_fichiers\icone-dossier.png/js/  
------- ------- H:\IUSTE\PHP\Eprojet Php_fichiers\icone-dossier.png/css/  
------- ------- ------- H:\IUSTE\PHP\Eprojet Php_fichiers\icone-fichier.pngstyle.php  
------- ------- H:\IUSTE\PHP\Eprojet Php_fichiers\icone-fichier.pnginit.inc.php  
------- ------- H:\IUSTE\PHP\Eprojet Php_fichiers\icone-fichier.pngfonction.inc.php  
------- ------- H:\IUSTE\PHP\Eprojet Php_fichiers\icone-fichier.pnghaut.inc.php  
------- ------- H:\IUSTE\PHP\Eprojet Php_fichiers\icone-fichier.pngbas.inc.php  
- H:\IUSTE\PHP\Eprojet Php_fichiers\icone-fichier.pnginscription.php  
- H:\IUSTE\PHP\Eprojet Php_fichiers\icone-fichier.pngconnexion.php  
- H:\IUSTE\PHP\Eprojet Php_fichiers\icone-fichier.pngprofil.php  
- H:\IUSTE\PHP\Eprojet Php_fichiers\icone-fichier.pngboutique.php  
- H:\IUSTE\PHP\Eprojet Php_fichiers\icone-fichier.pngfiche\_produit.php  
- H:\IUSTE\PHP\Eprojet Php_fichiers\icone-fichier.pngpanier.php  
- H:\IUSTE\PHP\Eprojet Php_fichiers\icone-fichier.png.htaccess  
  
  
Notre site web se trouvera à l'intérieur du dossier H:\IUSTE\PHP\Eprojet Php_fichiers\icone-dossier.png/site/.  
  
1 dossier H:\IUSTE\PHP\Eprojet Php_fichiers\icone-dossier.png/photo/ sera présent pour contenir les photos de nos produits.  
  
1 dossier H:\IUSTE\PHP\Eprojet Php_fichiers\icone-dossier.png/admin/ sera présent pour contenir les pages d'administration (BackOffice).  
  
1 dossier H:\IUSTE\PHP\Eprojet Php_fichiers\icone-dossier.png/inc/ sera présent pour contenir les fichiers n'étant pas des pages web a part entière (bien souvent il s'agit de fichiers inclus dans des pages web).  
  
Jusque là, comprennez vous notre arborescence ? elle est plutôt simple, non ? Et bien entendu, il y a aussi les pages web du site web côté FRONT :

* H:\IUSTE\PHP\Eprojet Php_fichiers\icone-fichier.pnginscription.php : page d'inscription pour les visiteurs
* H:\IUSTE\PHP\Eprojet Php_fichiers\icone-fichier.pngconnexion.php : page de connexion pour les membres
* H:\IUSTE\PHP\Eprojet Php_fichiers\icone-fichier.pngprofil.php : page de profil pour les membres connectés
* H:\IUSTE\PHP\Eprojet Php_fichiers\icone-fichier.pngboutique.php : catalogue des différents produits séparés par catégorie
* H:\IUSTE\PHP\Eprojet Php_fichiers\icone-fichier.pngfiche\_produit.php : fiche d'un produit en particulier
* H:\IUSTE\PHP\Eprojet Php_fichiers\icone-fichier.pngpanier.php : panier
* H:\IUSTE\PHP\Eprojet Php_fichiers\icone-fichier.png.htaccess : fichier permettant de faire des réglages, notamment sur les URLS pour améliorer le référencement/li>

**Etape 3. Ecriture des fichiers en inclusion**

**Etape 3. Ecriture du fichier inc/init.inc.php**

Le fichier init.inc.php (nommé .inc car destiné à l'inclusion et non pas à l'affichage), va nous permettre d'initialiser plusieurs choses sur notre site web.  
  
Le fichier init.inc.php sera donc inclut par toutes nos pages web afin de profiter de l'initialisation.  
  
Pourquoi initialiser ? qu'est-ce qu'on pourrait retrouver dedans ? la connexion à la base de données par exemple ! le réglage de l'encodage, et différentes variables...  
  
Et oui, sans ça, pas de site dynamique, nous en aurons donc besoin sur toutes les pages.

inc/init.inc.php

[?](file:///H:\IUSTE\PHP\Eprojet%20Php.htm)

|  |
| --- |
| <?php  //--------- BDD  $mysqli = new mysqli("localhost", "root", "", "site");  if ($mysqli->connect\_error) die('Un problème est survenu lors de la tentative de connexion à la BDD : ' . $mysqli->connect\_error);  // $mysqli->set\_charset("utf8");    //--------- SESSION  session\_start();    //--------- CHEMIN  define("RACINE\_SITE","/site/");    //--------- VARIABLES  $contenu = '';    //--------- AUTRES INCLUSIONS  require\_once("fonction.inc.php"); |

**Quelques explications**  
  
**Mysqli** est une classe prédéfinie en PHP me permettant de me connecter à la base de données.  
Pour cela il est nécessaire de lui annoncer le nom du serveur, le pseudo, le mot de passe, et la base de données à laquelle nous souhaitons nous connecter.  
  
1 condition est présente **$mysqli->connect\_error** pour afficher un message d'erreur en Français si jamais la connexion ne peut pas se faire (l'erreur est souvent due à une mauvaise information dans la chaine de connexion).  
  
**$mysqli->set\_charset("utf8");** permet de régler l'encodage de la base de données.  
  
  
**session\_start()** permet de créer (ou lire) 1 fichier de session sur le serveur. Sans cette ligne, nous ne pourrons pas connecter d'internautes à leurs espaces membres plus tard.  
Session\_start() permettra en effet de maintenir (et ne pas perdre) l'internaute connecté au site web même s'il navigue de page en page.  
  
  
**define("RACINE\_SITE","/site/");** permettra de gérer notre site web en chemin absolu et non pas relatif. Et pour éviter tout problème, nous pourrons modifier cette constante une fois en ligne pour que cela ait une repercution immédiate partout où elle sera appelée..  
  
  
**$contenu = '';** est une variable initialisée à vide pour éviter d'avoir des erreurs undefined si jamais nous tentons de l'afficher.  
Nous nous servirons de cette variable pour retenir des messages que nous devrions adresser à l'internaute, cela nous permettra de faire 1 affichage global de tous nos éventuels messages à un endroit précis (et non pas au dessus du doctype par exemple).  
  
  
**require\_once("fonction.inc.php");** nous allons inclure notre fichier de fonction avec nous. Du coup, lorsque nous appellerons le fichier init.inc.php, cela aura aussi pour effet d'inclure le fichier fonction.inc.php, (2 en 1) !

**Etape 3. Ecriture du fichier inc/haut.inc.php**

Nous allons écrire les balises indispensables de notre site web au format HTML.

inc/haut.inc.php

[?](file:///H:\IUSTE\PHP\Eprojet%20Php.htm)

|  |
| --- |
| <!Doctype html>  <html>      <head>          <title>Mon Site</title>          <link rel="stylesheet" href="<?php echo RACINE\_SITE; ?>inc/css/style.css">      </head>      <body>          <header>              <div class="conteneur">                  <div>                      <a href="" title="Mon Site">MonSite.com</a>                  </div>                  <nav>                      <a href="<?php echo RACINE\_SITE; ?>inscription.php">Inscription</a>                      <a href="<?php echo RACINE\_SITE; ?>connexion.php">Connexion</a>                      <a href="<?php echo RACINE\_SITE; ?>boutique.php">Accès à la boutique</a>                      <a href="<?php echo RACINE\_SITE; ?>panier.php">Voir votre panier</a>                  </nav>              </div>          </header>          <section>              <div class="conteneur"> |

Ce fichier nous permet de déclarer le haut du site avec le menu comprenant quelques liens (anticipés, puisque les pages ne sont pas encore créées).  
  
Les balises ne sont pas toutes fermées, c'est volontaire puisque nous allons mettre du contenu entre le fichier haut.inc.php et bas.inc.php.

**Etape 3. Ecriture du fichier inc/bas.inc.php**

Nous allons fermer les balises indispensables de notre site web au format HTML.

inc/bas.inc.php

[?](file:///H:\IUSTE\PHP\Eprojet%20Php.htm)

|  |
| --- |
| </div>          </section>          <footer>              <div class="conteneur">                  <?php echo date('Y'); ?> - Tous droits reservés - MonNom MonPrenom.              </div>          </footer>      </body>  </html> |

Ce fichier à l'avantage d'être simple !

**Etape 3. Ecriture du fichier inc/css/style.css**

Nous allons écrire un peu de code permettant de faire une mise en forme minimale.

inc/css/style.css

[?](file:///H:\IUSTE\PHP\Eprojet%20Php.htm)

|  |
| --- |
| \*{ margin: 0; }  a{ text-decoration: none; color: #000; }  .conteneur{ margin: 0 auto; max-: 1170px; }  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* HAUT \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  header { background: #000000; padding: 5px; text-align: center; }  header  span{ color: #fff; font-weight: bold; text-transform: uppercase; margin-right: 5%; }  header  nav{ display: inline; }  header  a{ color: #ffffff; text-decoration: none; padding: 5px; }  header  nav  a:hover{ background: #04baf6; }  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* MILIEU \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  section{ padding: 30px; min-height: 800px; }  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* BAS \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  footer{ background: #000; color: white; text-align: center; padding: 7px 0;}  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* GENERAL \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  .erreur{ background: #ff0000; padding: 5px; margin: 5px; }  .validation{ background: #669933; padding: 5px; margin: 5px; } |

Durant le projet, nous aurons l'occasion de revenir dans ce fichier pour l'alimenter d'avantages

**Etape 3. Ecriture de notre fichier inc/fonction.inc.php**

Nous allons créer 2 fichiers pour nous aider dans la conception du site web.

* Une fonction que nous nommerons **executeRequete** pour devinez quoi... exécuter des requêtes SQL !  
  En effet, échanger des informations avec la base est une action que nous devrons faire certainement plusieurs fois par page web, autant nous faciliter un peu la vie avec une fonction déjà prête.
* Une fonction que nous nommerons **debug** pour nous debugger !  
  En PHP, il est souvent nécessaire d'effectuer des var\_dump ou print\_r pour voir le contenu d'un tableau array, d'un objet ou de variables, nous allons donc prévoir un code en conséquence afin de gagner du temps.

inc/fonction.inc.php

[?](file:///H:\IUSTE\PHP\Eprojet%20Php.htm)

|  |
| --- |
| function executeRequete($req)  {      global $mysqli;      $resultat = $mysqli->query($req);      if(!$resultat) //      {          die("Erreur sur la requete sql.<br>Message : " . $mysqli->error . "<br>Code: " . $req);      }      return $resultat; //  }    function debug($var, $mode = 1)  {      echo '<div style="background: orange; padding: 5px; float: right; clear: both; ">';      $trace = debug\_backtrace();      $trace = array\_shift($trace);      echo 'Debug demandé dans le fichier : $trace[file] à la ligne $trace[line].';      if($mode === 1)      {          echo '<pre>'; print\_r($var); echo '</pre>';      }      else      {          echo '<pre>'; var\_dump($var); echo '</pre>';      }      echo '</div>';  } |

**La fonction executeRequete**  
  
**function executeRequete($req)** La fonction sera destinée à recevoir 1 argument entrant (la requête SQL arrivera dans la variable de reception **$req** prévue à cet effet).  
  
**global $mysqli;** permet d'avoir accès à la variable $mysqli définie dans le fichier **init.inc.php** (espace global) à l'intérieur de notre fonction (espace local).  
  
**$resultat = $mysqli->query($req);** on exécute la requête reçue en argument et on gardera les résultats dans la variable **$resultat**.  
  
**if(!$resultat)** si la variable **$resultat** renvoie false, c'est qu'il y a une erreur de requête SQL.  
  
**die("Erreur sur la requete sql.<br>Message : " . $mysqli->error . "<br>Code: " . $req);** Dans le cas où la requête échoue, on lui demande d'adresser 1 message et d'arreter l'exécution du code avec l'utilisation de die.  
  
**return $resultat;** en cas d'une requête de SELECTion, on retournera un objet issu de la classe mysqli\_result. Sinon (pour INSERT/UPDATE/DELETE), nous retournerons un boolean TRUE (1).  
  
  
**La fonction debug**  
  
**function debug($var, $mode = 1)** La fonction sera destinée à recevoir 1 ou 2 argument(s) entrant(s). En premier ce sera la variable/array/object à explorer et en second ce sera 1 mode d'affichage.  
  
**$trace = debug\_backtrace();** Fonction prédéfinie retournant un tableau Array contenant des informations tel que la ligne et le fichier où est exécuté la fonction.  
  
**$trace = array\_shift($trace);** Extrait la première valeur d'un tableau et la retourne. Dans notre cas cela permet de retirer une dimension au tableau array $trace.  
  
**echo "Debug demandé dans le fichier : $trace[file] à la ligne $trace[line].<hr>";** Au moment de l'affichage, cela permettra de savoir de quel fichier vient la demande de debug  
  
**if($mode === 1)<hr>";** Si le mode 1 est précisé en argument (ou qu'il n'y a pas d'informations contraires), nous ferons un print\_r  
  
**else** Sinon, nous ferons un var\_dump

**Etape 4. Ecriture de la page inscription.php**

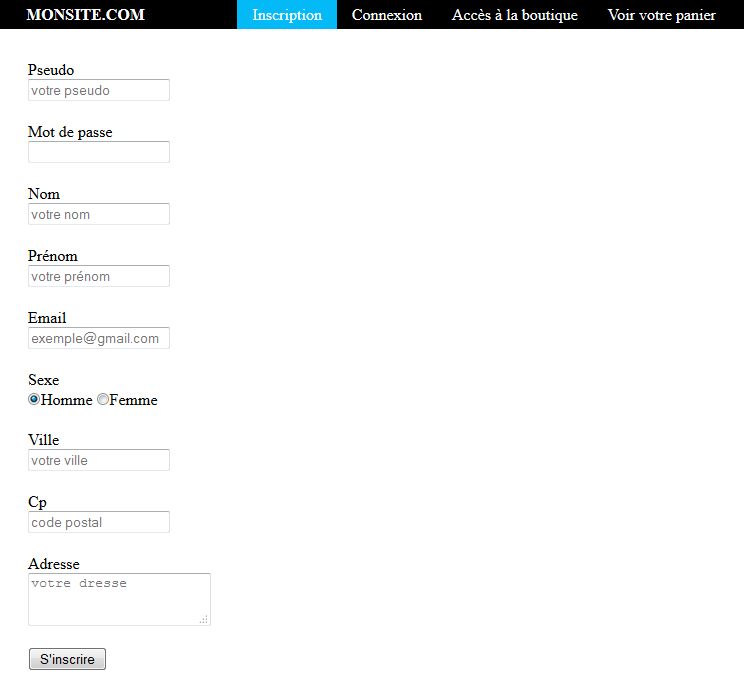
Nous allons commencer notre site web par l'espace membre et donc la page d'inscription !  
  
Pour cela, commençons par le formulaire HTML :

inscription.php

[?](file:///H:\IUSTE\PHP\Eprojet%20Php.htm)

|  |
| --- |
| <?php require\_once("inc/init.inc.php"); ?>  <?php require\_once("inc/haut.inc.php"); ?>    <form method="post" action="">      <label for="pseudo">Pseudo</label><br>      <input type="text" id="pseudo" name="pseudo" maxlength="20" placeholder="votre pseudo" pattern="[a-zA-Z0-9-\_.]{1,20}" title="caractères acceptés : a-zA-Z0-9-\_." required="required"><br><br>        <label for="mdp">Mot de passe</label><br>      <input type="password" id="mdp" name="mdp" required="required"><br><br>        <label for="nom">Nom</label><br>      <input type="text" id="nom" name="nom" placeholder="votre nom"><br><br>        <label for="prenom">Prénom</label><br>      <input type="text" id="prenom" name="prenom" placeholder="votre prénom"><br><br>        <label for="email">Email</label><br>      <input type="email" id="email" name="email" placeholder="exemple@gmail.com"><br><br>        <label for="civilite">Civilité</label><br>      <input name="civilite" value="m" checked="" type="radio">Homme      <input name="civilite" value="f" type="radio">Femme<br><br>        <label for="ville">Ville</label><br>      <input type="text" id="ville" name="ville" placeholder="votre ville" pattern="[a-zA-Z0-9-\_.]{5,15}" title="caractères acceptés : a-zA-Z0-9-\_."><br><br>        <label for="cp">Code Postal</label><br>      <input type="text" id="code\_postal" name="code\_postal" placeholder="code postal" pattern="[0-9]{5}" title="5 chiffres requis : 0-9"><br><br>        <label for="adresse">Adresse</label><br>      <textarea id="adresse" name="adresse" placeholder="votre dresse" pattern="[a-zA-Z0-9-\_.]{5,15}" title="caractères acceptés :  a-zA-Z0-9-\_."></textarea><br><br>        <input type="submit" name="inscription" value="S'inscrire">  </form>    <?php require\_once("inc/bas.inc.php"); ?> |

Nous incluons le fichier init.inc.php, le haut du site, le bas du site, et entre le haut et le bas nous mettons notre formulaire html afin que nos futurs internautes puissent s'inscrire.  
  
Il est très important que les attributs name du formulaire soient prévus afin de pouvoir récupérer et exploiter les saisies en PHP. De préférence, nous pouvons garder les mêmes name que le nom de nos champs dans notre base de données.

Résultat - inscription.php 

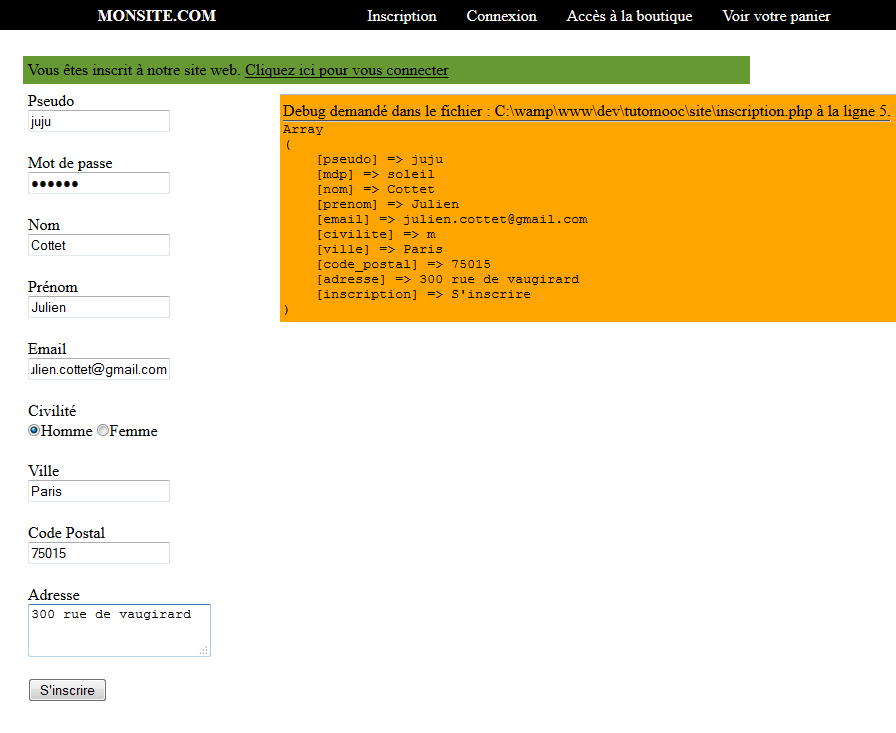
Il faut aussi prévoir la partie traitement en PHP, voici la suite du code :

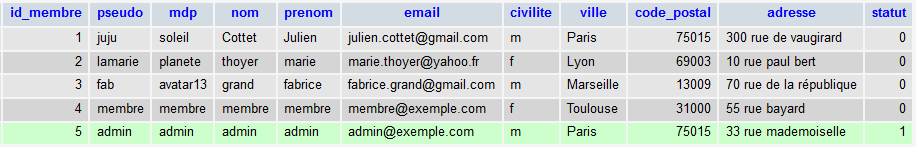
inscription.php

[?](file:///H:\IUSTE\PHP\Eprojet%20Php.htm)

|  |
| --- |
| <?php require\_once("inc/init.inc.php");  //--------------------------------- TRAITEMENTS PHP ---------------------------------//  if($\_POST)  {      debug($\_POST);      $verif\_caractere = preg\_match('#^[a-zA-Z0-9.\_-]+$#', $\_POST['pseudo']);      if(!$verif\_caractere && (strlen($\_POST['pseudo']) < 1 || strlen($\_POST['pseudo']) > 20) ) //      {          $contenu .= "<div class='erreur'>Le pseudo doit contenir entre 1 et 20 caractères. <br> Caractère accepté : Lettre de A à Z et chiffre de 0 à 9</div>";      }      else      {          $membre = executeRequete("SELECT \* FROM membre WHERE pseudo='$\_POST[pseudo]'");          if($membre->num\_rows > 0)          {              $contenu .= "<div class='erreur'>Pseudo indisponible. Veuillez en choisir un autre svp.</div>";          }          else          {              // $\_POST['mdp'] = md5($\_POST['mdp']);              foreach($\_POST as $indice => $valeur)              {                  $\_POST[$indice] = htmlEntities(addSlashes($valeur));              }              executeRequete("INSERT INTO membre (pseudo, mdp, nom, prenom, email, civilite, ville, code\_postal, adresse) VALUES ('$\_POST[pseudo]', '$\_POST[mdp]', '$\_POST[nom]', '$\_POST[prenom]', '$\_POST[email]', '$\_POST[civilite]', '$\_POST[ville]', '$\_POST[code\_postal]', '$\_POST[adresse]')");              $contenu .= "<div class='validation'>Vous êtes inscrit à notre site web. <a href=\"connexion.php\"><u>Cliquez ici pour vous connecter</u></a></div>";          }      }  }  //--------------------------------- AFFICHAGE HTML ---------------------------------//  ?>  <?php require\_once("inc/haut.inc.php"); ?>  <?php echo $contenu; ?>    <form method="post" action="">      <label for="pseudo">Pseudo</label><br>      <input type="text" id="pseudo" name="pseudo" maxlength="20" placeholder="votre pseudo" pattern="[a-zA-Z0-9-\_.]{1,20}" title="caractères acceptés : a-zA-Z0-9-\_." required="required"><br><br>        <label for="mdp">Mot de passe</label><br>      <input type="password" id="mdp" name="mdp" required="required"><br><br>        <label for="nom">Nom</label><br>      <input type="text" id="nom" name="nom" placeholder="votre nom"><br><br>        <label for="prenom">Prénom</label><br>      <input type="text" id="prenom" name="prenom" placeholder="votre prénom"><br><br>        <label for="email">Email</label><br>      <input type="email" id="email" name="email" placeholder="exemple@gmail.com"><br><br>        <label for="civilite">Civilité</label><br>      <input name="civilite" value="m" checked="" type="radio">Homme      <input name="civilite" value="f" type="radio">Femme<br><br>        <label for="ville">Ville</label><br>      <input type="text" id="ville" name="ville" placeholder="votre ville" pattern="[a-zA-Z0-9-\_.]{5,15}" title="caractères acceptés : a-zA-Z0-9-\_."><br><br>        <label for="cp">Code Postal</label><br>      <input type="text" id="code\_postal" name="code\_postal" placeholder="code postal" pattern="[0-9]{5}" title="5 chiffres requis : 0-9"><br><br>        <label for="adresse">Adresse</label><br>      <textarea id="adresse" name="adresse" placeholder="votre dresse" pattern="[a-zA-Z0-9-\_.]{5,15}" title="caractères acceptés :  a-zA-Z0-9-\_."></textarea><br><br>        <input type="submit" name="inscription" value="S'inscrire">  </form>    <?php require\_once("inc/bas.inc.php"); ?> |

**Quelques Explications**  
  
**if($\_POST)** Cette condition IF permet de detecter si l'internaute à cliquer sur le bouton submit pour s'inscrire.  
  
**debug($\_POST);** Si l'internaute sollicite une inscription, nous allons utiliser notre fonction debug afin de voir les saisies qu'il a postées (le temps de faire des tests). Cette fonction a été inclut par le fichier init.inc.php puisqu'il inclue lui même fonction.inc.php  
  
**$verif\_caractere = preg\_match('#^[a-zA-Z0-9.\_-]+$#', $\_POST['pseudo']);** Nous vérifions qu'il n'y ai pas de mauvais caractère dans le pseudo. (return 0 si mauvais caractère dans le pseudo, 1 sinon). vous pouvez écrire echo $verif\_caractere;  
*preg\_match() est une expression régulière (regex) toujours entourée du symbole # dieze afin de préciser des options choisies :****^*** *désigne le début de la chaine.****$*** *désigne la fin de la chaine.****+*** *est présent pour dire que les lettres autorisées peuvent apparaitre plusieurs fois.*   
  
**if(!$verif\_caractere && (strlen($\_POST['pseudo']) < 1 || strlen($\_POST['pseudo']) > 20) )** A travers cette condition, nous vérifions qu'il n'y ai pas un caractère interdit ou un problème de taille sur le pseudo. Cela reste faible, dans une version plus aboutie il faudrait penser à renforcer les contrôles (sur le pseudo mais aussi les autres champs).  
  
**$contenu .= "<div class='erreur'>Le pseudo doit contenir entre 1 et 20 caractères. <br> Caractère accepté : Lettre de A à Z et chiffre de 0 à 9</div>";** En cas d'erreur, nous allons en informer l'internaute mais pas tout de suite ! sinon nous serions au dessus du doctype niveau code-source, nous allons donc retenir l'affichage du message dans la variable $contenu que nous remplissons afin de l'afficher plus tard.  
  
**else** Sinon, la variable $contenu est vide c'est qu'il n'y a pas eu d'erreur précédemment.  
  
**$membre = executeRequete("SELECT \* FROM membre WHERE pseudo='$\_POST[pseudo]'");** Nous allons utiliser notre fonction executeRequete pour allez voir si le pseudo que l'internaute tente de prendre n'est pas déjà attribué à un autre membre.  
  
**if($membre->num\_rows > 0)** Si la requête renvoie plus de 0 résultat (donc au moins 1), c'est que le pseudo est déjà attribué à quelqu'un d'autre.  
  
**$contenu .= "<div class='erreur'>Pseudo indisponible. Veuillez en choisir un autre svp.</div>";** Nous invitons le membre à choisir un autre pseudo si celui qu'il convoite est déjà attribué.  
  
**else** Sinon, on lance l'inscription.  
  
**// $\_POST['mdp'] = md5($\_POST['mdp']);** ce code est en commentaire, nous pouvons crypter le mot de passe afin qu'il ne soit pas affiché en clair dans la base de données. /!\ Attention, si vous activez cette ligne, il faudra penser à ajouter 1 ligne de cryptage au moment de la connexion (le membre envoie son mot de passe, on le recryptera pour le comparer avec la chaine cryptée en base). Pour la suite du cours, nous ne garderons pas cette ligne et travaillerons avec des mots de passe en clair (le temps de l'entrainement).  
  
**foreach($\_POST as $indice => $valeur){ $\_POST[$indice] = htmlEntities(addSlashes($valeur));}** nous bouclons sur toutes les saisies afin de les passer dans les fonctions prédéfinies PHP htmlEntities et addSlashes. /!\ Cela permet d'effectuer 1 premier traitement mais ce n'est pas pour autant complétement sécurisé.  
  
**executeRequete("INSERT INTO membre (pseudo, mdp, nom, prenom, email, civilite, ville, code\_postal, adresse) VALUES ('$\_POST[pseudo]', '$\_POST[mdp]', '$\_POST[nom]', '$\_POST[prenom]', '$\_POST[email]', '$\_POST[civilite]', '$\_POST[ville]', '$\_POST[code\_postal]', '$\_POST[adresse]')");** Cette requête permet d'insérer le membre dans la base ! C'est à ce moment-là que l'enregistrement se fait.  
  
**$contenu .= "<div class='validation'>Vous êtes inscrit à notre site web. <a href=\"connexion.php\"><u>Cliquez ici pour vous connecter</u></a></div>";** Nous l'informons de son inscription et lui proposons de se connecter. (nous pourrions aussi prévoir de connecter l'internaute automatiquement).

Résultat - inscription.php 

Vous devriez retrouver la trace de votre 1er membre dans la base de données :  
  
explication PHP  
  
Maintenant que votre formulaire fonctionne, inscrivez plusieurs membres différents histoire que nous ayons plusieurs enregistrements pour faire des tests plus tard :  
  
  
  
Rendez-vous dans PhpMyAdmin pour modifier le statut de l'un des membres afin de mettre son statut sur "1". Cela nous permettra d'avoir 1 administrateur (indispensable pour la suite des tests).

**Etape 5. Ecriture de la page connexion.php**

Maintenant que nous avons une page d'inscription fonctionnelle et plusieurs inscrits, nous allons attaquer la page de connexion !  
  
Pour cela, commençons par le formulaire HTML :

inscription.php

[?](file:///H:\IUSTE\PHP\Eprojet%20Php.htm)

|  |
| --- |
| <?php require\_once("inc/init.inc.php"); ?>  <?php require\_once("inc/haut.inc.php"); ?>    <form method="post" action="">      <label for="pseudo">Pseudo</label><br>      <input type="text" id="pseudo" name="pseudo"><br> <br>        <label for="mdp">Mot de passe</label><br>      <input type="text" id="mdp" name="mdp"><br><br>         <input type="submit" value="Se connecter">  </form>    <?php require\_once("inc/bas.inc.php"); ?> |

Nous incluons le fichier init.inc.php, le haut du site, le bas du site, et entre le haut et le bas nous mettons notre formulaire html afin que nos futurs internautes puissent se connecter.  
  
Il est très important que les attributs name du formulaire soit prévue afin de pouvoir récupérer et exploiter les saisies en PHP. De préférence, pour une meilleure cohérence, nous pouvons garder les mêmes name que le nom de nos champs dans notre base de données.

Résultat - connexion.php 

Il faut aussi prévoir la partie traitement en PHP, voici la suite du code :

connexion.php

[?](file:///H:\IUSTE\PHP\Eprojet%20Php.htm)

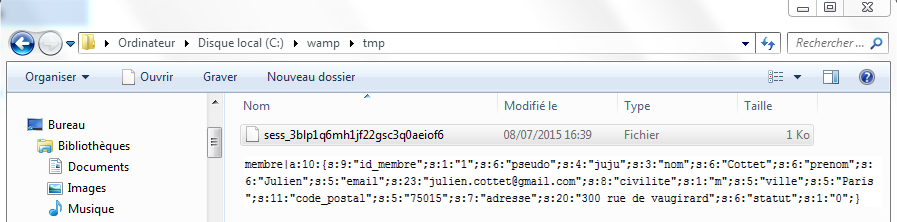
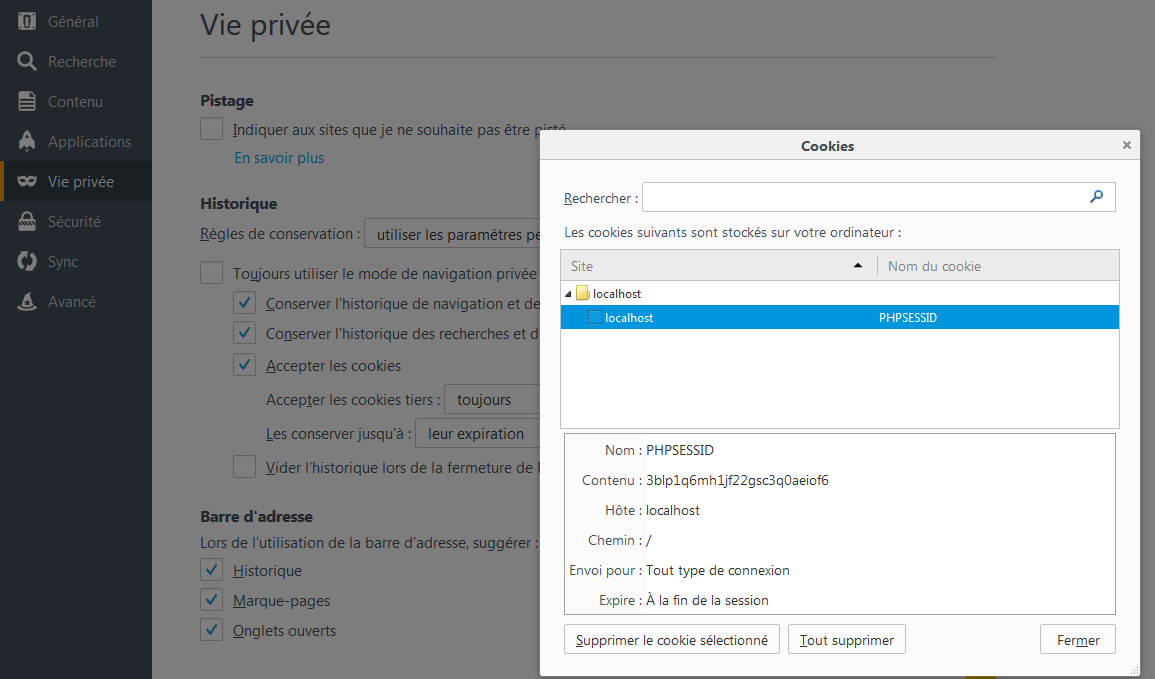
|  |
| --- |
| <?php require\_once("inc/init.inc.php");  //--------------------------------- TRAITEMENTS PHP ---------------------------------//  if($\_POST)  {      // $contenu .=  "pseudo : " . $\_POST['pseudo'] . "<br>mdp : " .  $\_POST['mdp'] . "";      $resultat = executeRequete("SELECT \* FROM membre WHERE pseudo='$\_POST[pseudo]'");      if($resultat->num\_rows != 0)      {          // $contenu .=  '<div style="background:green">pseudo connu!</div>';          $membre = $resultat->fetch\_assoc();          if($membre['mdp'] == $\_POST['mdp'])          {              //$contenu .= '<div class="validation">mdp connu!</div>';              foreach($membre as $indice => $element)              {                  if($indice != 'mdp')                  {                      $\_SESSION['membre'][$indice] = $element;                  }              }              header("location:profil.php");          }          else          {              $contenu .= '<div class="erreur">Erreur de MDP</div>';          }      }      else      {          $contenu .= '<div class="erreur">Erreur de pseudo</div>';      }  }  //--------------------------------- AFFICHAGE HTML ---------------------------------//  ?>  <?php require\_once("inc/haut.inc.php"); ?>  <?php echo $contenu; ?>    <form method="post" action="">      <label for="pseudo">Pseudo</label><br>      <input type="text" id="pseudo" name="pseudo"><br> <br>        <label for="mdp">Mot de passe</label><br>      <input type="text" id="mdp" name="mdp"><br><br>         <input type="submit" value="Se connecter">  </form>    <?php require\_once("inc/bas.inc.php"); ?> |

**Quelques Explications**  
  
**if($\_POST)** Cette condition IF permet de detecter si l'internaute à cliqué sur le bouton submit pour se connecter.  
  
**$resultat = executeRequete("SELECT \* FROM membre WHERE pseudo='$\_POST[pseudo]'");** Nous allons utiliser notre fonction executeRequete pour allez consulter la base afin de savoir si le pseudo avec lequel l'internaute tente de se connecter correspond bien à 1 compte réel sur notre site web. Y'a t'il un enregistrement correspondant dans notre base ?  
  
**if($resultat->num\_rows != 0)** Si le nombre de retours est différent de 0 (donc 1 logiquement :p), c'est que le pseudo est connu et que le compte existe, on peut avancer...  
  
**else** Sinon, nous informons l'internaute qu'il y a une erreur sur son pseudo...  
  
**$membre = $resultat->fetch\_assoc();** Revenons sur le cas du pseudo valide, nous devons absolument traiter (fetch\_assoc) pour connaitre les informations récupérées en base. En effet, nous devons savoir si le membre a le bon pseudo mais aussi s'il possède le bon mot de passe associé.  
  
**if($membre['mdp'] == $\_POST['mdp'])** On compare le mdp posté (dans le formulaire de connexion) avec le mdp du membre (récupéré dans la base de données), s'il s'agit du même mdp dans les deux cas, on crée à l'internaute une session et on la remplit avec certaines informations (c'est ce qui permet réellement de connecter et de maintenir connecté quelqu'un sur 1 site web).  
  
**else** Sinon, le mdp est mauvais et nous informons l'internaute.  
  
**foreach($membre as $indice => $element){if($indice != 'mdp'){ $\_SESSION['membre'][$indice] = $element; } }** Nous créons une session avec les éléments de la base de données. La boucle foreach évite d'écrire les lignes suivantes : $\_SESSION['membre']['id\_membre'] = $membre['id\_membre']; $\_SESSION['membre']['pseudo'] = $membre['pseudo']; etc.. Par sécurité et comme nous n'en n'aurons pas besoin, nous ne conserverons pas le mdp dans la session (condition IF).  
Pour rappel, sans le session\_start() placée dans le fichier init.inc.php, nous n'aurions pas pu se servir du système de session en PHP  
  
**header("location:profil.php");** Si l'accouplement pseudo/mot de passe est bon, nous redirigeons l'internaute vers sa page de profil (puisqu'il est maintenant connecté !)

Résultat - connexion.php 

**Faites plusieurs tests !**

* Avec 1 mauvais pseudo,
* Avec 1 bon pseudo mais 1 mauvais mdp,
* Avec 1 bon pseudo et 1 mauvais mdp !

*ps : dans le cas d'un bon pseudo et mot de passe, si vous arrivez sur une page "non trouvée" c'est tout à fait normal car nous n'avons pas encore créé la page de profil.*  
  
Quoi qu'il en soit "être connecté à un site web" ne signifie pas seulement avoir le bon pseudo et le bon mot de passe.  
  
Pour maintenir une connexion il faut avoir un fichier de session (sur le serveur du site) et un fichier cookie sur votre ordinateur pour assurer la liaison.  
  
**Session dans le dossier /tmp/ du serveur :**   
  
  
*Ce fichier a été créé par session\_start() (placé dans le fichier inc/init.inc.php) automatiquement.  
Ce même fichier a été rempli par la superglobale $\_SESSION lors de la connexion.*  
  
**Cookie sur l'ordinateur de l'internaute :**   


**Etape 6. Ecriture de la page profil.php**

A présent, nos visiteurs peuvent s'inscrire et aussi se connecter !  
C'est un bon début mais si les internautes font la démarche de s'inscrire et de se connecter c'est pour arriver quelque part et ainsi profiter d'un espace reservé aux membres sur le site web.  
  
Nous allons donc prévoir une page de profil avec quelques informations !  
  
Pour récupérer les informations, nous pourrons nous servir du fichier de session *(par l'intermédiaire de la superglobale $\_SESSION)* que nous avons créé au moment de la connexion. (C'est pratique car ces informations de session seront accessibles sur tout le site web).  
  
Avant de rentrer dans l'espace de profil nous devons nous assurer que l'internaute qui tente d'y accèder à les droits nécessaires (c'est à dire, être passé par la page de connexion avec un bon pseudo et un bon mot de passe. En gros avoir un fichier de session).  
  
Par exemple, vous ne pouvez pas accèdez à votre boite de reception d'email sans vous être connecté au préalable. De la même manière, nous allons tester si 1 fichier de session existe avant d'accepter l'internaute sur l'espace de profil.  
  
Pour cela, nous allons écrire dans notre fichier inc/fonction.inc.php :

inc/fonction.inc.php

[?](file:///H:\IUSTE\PHP\Eprojet%20Php.htm)

|  |
| --- |
| function executeRequete($req)  {      global $mysqli;      $resultat = $mysqli->query($req);      if(!$resultat)      {          die("Erreur sur la requete sql.<br>Message : " . $mysqli->error . "<br>Code: " . $req);      }      return $resultat;  }  //------------------------------------  function debug($var, $mode = 1)  {      echo '<div style="background: orange; padding: 5px; float: right; clear: both; ">';      $trace = debug\_backtrace();      $trace = array\_shift($trace);      echo 'Debug demandé dans le fichier : $trace[file] à la ligne $trace[line].';      if($mode === 1)      {          print '<pre>'; print\_r($var); print '</pre>';      }      else      {          print '<pre>'; var\_dump($var); print '</pre>';      }      echo '</div>';  }  //------------------------------------  function internauteEstConnecte()  {      if(!isset($\_SESSION['membre'])) return false;      else return true;  }  //------------------------------------  function internauteEstConnecteEtEstAdmin()  {      if(internauteEstConnecte() && $\_SESSION['membre']['statut'] == 1) return true;      else return false;  } |

**Explications**  
La fonction **internauteEstConnecte()** va nous permettre de savoir si l'internaute est connecté par une simple condition :  
  
   **if(!isset($\_SESSION['membre'])) return false;** si la session membre n'existe pas, c'est que l'internaute n'est pas passé par la page de connexion (ou alors qu'il a été déconnecté depuis). on retournera la valeur "FALSE" pour dire "Faux l'internaute n'est pas connecté".  
  
   **else** sinon, c'est que la session membre existe et donc que l'internaute est connecté (avec 1 fichier de session + cookie). on retournera la valeur "TRUE" pour dire "Vrai l'internaute est connecté".  
  
L'avantage d'avoir mis ce code dans une fonction **internauteEstConnecte()** et non pas directement dans la page web c'est qu'il sera plus facile de s'en reservir sur les autres pages web (plutôt que re-trimbaler un même morceau code en copier/coller d'un fichier à l'autre). Cela sera pratique de savoir si l'internaute est connecté à divers endroits du site web.  
  
  
La fonction **internauteEstConnecteEtEstAdmin()** va nous permettre de savoir si l'internaute est connecté en tant qu'administrateur (statut a 1) ou en tant que membre (statut a 0) :  
  
   **if(internauteEstConnecte() && $\_SESSION['membre']['statut'] == 1** si la fonction internauteEstConnecte() renvoie "TRUE", l'internaute est connecté (avec 1 fichier de session + cookie), on vérifie donc si son statut est a 1. Si oui, nous renverrons TRUE pour dire "Vrai, cet internaute est connecté et est admin".  
  
   **else** sinon, c'est soit que l'internaute n'est pas connecté ou soit que l'internaute est connecté mais sans avoir les droits d'administration, nous renverrons donc "FALSE" pour dire "Faux cet internaute n'est pas administrateur".  
  
L'avantage d'avoir mis ce code dans une fonction **internauteEstConnecte()** et non pas directement dans la page web c'est qu'il sera plus facile de s'en reservir sur les autres pages web (plutôt que re-trimbaler un même morceau code en copier/coller d'un fichier à l'autre). Cela sera pratique de savoir si l'internaute est connecté à divers endroits du site web.  
  
  
Maintenant que nous avons 2 fonctions qui vont pouvoir nous aider, nous allons créer la page de profil :

profil.php

[?](file:///H:\IUSTE\PHP\Eprojet%20Php.htm)

|  |
| --- |
| <?php  require\_once("inc/init.inc.php");  //--------------------------------- TRAITEMENTS PHP ---------------------------------//  if(!internauteEstConnecte()) header("location:connexion.php");  // debug($\_SESSION);  $contenu .= '<p class="centre">Bonjour <strong>' . $\_SESSION['membre']['pseudo'] . '</strong></p>';  $contenu .= '<div class="cadre"><h2> Voici vos informations </h2>';  $contenu .= '<p> votre email est: ' . $\_SESSION['membre']['email'] . '<br>';  $contenu .= 'votre ville est: ' . $\_SESSION['membre']['ville'] . '<br>';  $contenu .= 'votre cp est: ' . $\_SESSION['membre']['code\_postal'] . '<br>';  $contenu .= 'votre adresse est: ' . $\_SESSION['membre']['adresse'] . '</p></div><br><br>';  //--------------------------------- AFFICHAGE HTML ---------------------------------//  require\_once("inc/haut.inc.php");  echo $contenu;  require\_once("inc/bas.inc.php"); |

**Explications**  
Nous verifions **if(!internauteEstConnecte())** si l'internaute (!) N'EST PAS connecté (le point d'exclamation demande si la fonction renvoie false, donc si l'internaute n'est pas connecté).  
  
Si l'internaute n'est pas connecté, il n'a rien à faire sur la page de profil, nous le renvoyons vers la page de connexion **header("location:connexion.php");**.  
  
Pour construire la page de profil, nous piochons dans le fichier session (dans le dossier /tmp/ sur le serveur, par l'intermédiaire de la superglobale $\_SESSION) afin d'afficher les informations de l'internaute connecté.  
  
Connectez-vous, vous verrez que notre profil est encore incomplet mais qu'il se construit peu à peu :

Résultat - profil.php 

Plus tard, nous pourrons ajouter le suivi des commandes dans l'espace de profil d'un membre.

**Etape 6. Un menu évolutif en fonction de notre statut**

Au sein de notre site web, il y a plusieurs statuts :

* **Visiteur** : correspond à un internaute non connecté
* **Membre** : correspond à un internaute connecté
* **Admin** : correspond à un internaute connecté avec des droits d'administration

Il est donc normal que selon son statut nous n'ayons pas accès aux mêmes droits sur le site web.  
  
Voici donc la mise à jour de notre fichier inc/haut.inc.php (au niveau de la navigation/menu) :

inc/haut.inc.php

[?](file:///H:\IUSTE\PHP\Eprojet%20Php.htm)

|  |
| --- |
| <!Doctype html>  <html>      <head>          <title>Mon Site</title>          <link rel="stylesheet" href="<?php echo RACINE\_SITE; ?>inc/css/style.css">      </head>      <body>          <header>              <div class="conteneur">                  <div>                      <a href="" title="Mon Site">MonSite.com</a>                  </div>                  <nav>                      <?php                      if(internauteEstConnecteEtEstAdmin())                      {                          echo '<a href="' . RACINE\_SITE . 'admin/gestion\_membre.php">Gestion des membres</a>';                          echo '<a href="' . RACINE\_SITE . 'admin/gestion\_commande.php">Gestion des commandes</a>';                          echo '<a href="' . RACINE\_SITE . 'admin/gestion\_boutique.php">Gestion de la boutique</a>';                      }                      if(internauteEstConnecte())                      {                          echo '<a href="' . RACINE\_SITE . 'profil.php">Voir votre profil</a>';                          echo '<a href="' . RACINE\_SITE . 'boutique.php">Accès à la boutique</a>';                          echo '<a href="' . RACINE\_SITE . 'panier.php">Voir votre panier</a>';                          echo '<a href="' . RACINE\_SITE . 'connexion.php?action=deconnexion">Se déconnecter</a>';                      }                      else                      {                          echo '<a href="' . RACINE\_SITE . 'inscription.php">Inscription</a>';                          echo '<a href="' . RACINE\_SITE . 'connexion.php">Connexion</a>';                          echo '<a href="' . RACINE\_SITE . 'boutique.php">Accès à la boutique</a>';                          echo '<a href="' . RACINE\_SITE . 'panier.php">Voir votre panier</a>';                      }                      ?>                  </nav>              </div>          </header>          <section>              <div class="conteneur"> |

**Explications**  
Si l'internaute est **Administrateur** **if(internauteEstConnecteEtEstAdmin())** nous lui proposerons des liens de gestion (backOffice) pour gérer ses produits, ses commandes, ses membres, etc.  
  
Si l'internaute est **Membre** **if(internauteEstConnecte())** nous lui proposerons plusieurs liens dont son espace de profil  
  
Si l'internaute est **Visiteur** **else** nous lui proposerons d'autres liens notamment l'inscription et la connexion  
  
En conclusion :  
fléche1 visiteur aura donc accès à 4 liens  
fléche1 membre aura également 4 liens affichés  
fléche1 administrateur verra 7 liens sur la page web (1 administrateur est aussi 1 membre).  
  
Au passage, pour gérer la deconnexion, nous enverrons dans l'url une information ?action=deconnexion afin de pouvoir détecter que l'internaute ai bien cliqué sur le lien "se déconnecter".  
  
Sinon comment le savoir ? il nous faut une information dans l'url que nous pourrons aller chercher plus tard avec la superglobale $\_GET afin d'associer le traitement permettant de déconnecter l'internaute.  
  
Nous gérerons la deconnexion sur la page de connexion.

**Etape 6. La deconnexion et la fin de l'espace membre**

Que l'on puisse se connecter, c'est super, mais il faut aussi pouvoir se déconnecter :p.  
  
Pour terminer l'espace membre, nous allons mettre à jour la page de connexion :

connexion.php

[?](file:///H:\IUSTE\PHP\Eprojet%20Php.htm)

|  |
| --- |
| <?php require\_once("inc/init.inc.php");  //--------------------------------- TRAITEMENTS PHP ---------------------------------//  if(isset($\_GET['action']) && $\_GET['action'] == "deconnexion")  {      session\_destroy();  }  if(internauteEstConnecte())  {      header("location:profil.php");  }  if($\_POST)  {      // $contenu .=  "pseudo : " . $\_POST['pseudo'] . "<br>mdp : " .  $\_POST['mdp'] . "";      $resultat = executeRequete("SELECT \* FROM membre WHERE pseudo='$\_POST[pseudo]'");      if($resultat->num\_rows != 0)      {          // $contenu .=  '<div style="background:green">pseudo connu!</div>';          $membre = $resultat->fetch\_assoc();          if($membre['mdp'] == $\_POST['mdp'])          {              //$contenu .= '<div class="validation">mdp connu!</div>';              foreach($membre as $indice => $element)              {                  if($indice != 'mdp')                  {                      $\_SESSION['membre'][$indice] = $element;                  }              }              header("location:profil.php");          }          else          {              $contenu .= '<div class="erreur">Erreur de MDP</div>';          }      }      else      {          $contenu .= '<div class="erreur">Erreur de pseudo</div>';      }  }  //--------------------------------- AFFICHAGE HTML ---------------------------------//  ?>  <?php require\_once("inc/haut.inc.php"); ?>  <?php echo $contenu; ?>    <form method="post" action="">      <label for="pseudo">Pseudo</label><br>      <input type="text" id="pseudo" name="pseudo"><br> <br>        <label for="mdp">Mot de passe</label><br>      <input type="text" id="mdp" name="mdp"><br><br>         <input type="submit" value="Se connecter">  </form>    <?php require\_once("inc/bas.inc.php"); ?> |

**Quelques Explications**  
  
**if(isset($\_GET['action']) && $\_GET['action'] == "deconnexion")** Si l'internaute clic sur le lien deconnexion, nous arriverons sur la page connexion.php avec l'information suivante dans l'url ?action=deconnexion. C'est la raison pour laquelle nous utilisons la superglobale $\_GET afin de detecter cette action et de déconnecter l'internaute via session\_destroy();.  
  
Au passage, nous en profitons pour améliorer l'espace membre et dire que si l'internaute est déjà connecté mais qu'il tente d'aller sur la page de connexion (volontairement ou involontairement), nous le renverrons automatiquement dans son espace de profil : header("location:profil.php");.  
  
Cela parait logique, est-ce que vous voyez la page de connexion GMAIL alors que vous êtes connecté ? non, cela a été prévu dans le code...  
  
Et oui le code c'est aussi ça, PENSER A TOUT ! le fonctionnel, les contrôles, la sécurité, traquer les éventuelles incohérences, la recherche, les tests, etc.  
On dit qu'1 site est totalement terminé lorsque tous les cas sont prévus par le script ! Autant dire qu'il y a beaucoup de sites qui ne sont pas terminés...  
  
**Vous pouvez faire des tests** :  
flécheConnectez vous et déconnectez vous ! accèdez au site en tant qu'Admin, en tant que Membre, en tant que Visiteur.  
flécheTenter d'aller sur la page de profil en étant seulement visiteur.  
flécheVous pouvez aussi tenter d'aller sur la page de connexion en étant déjà connecté en tant que membre.  
Cela vous permettra de voir comment le site réagi.  
  
**Erreur header** :  
Si vous avez une erreur du type : "Cannot modify header information - headers already sent by ..., Cannot send session cookie - headers already sent by ...", il faudra faire attention à ne pas mettre d'echo ou d'espace avant d'effectuer les redirections header.  
  
Généralement, si vous regardez bien ces erreurs, vous y verrez 2 lignes concernées, la ligne correspondant à l'appel de la fonction header qui ne peut pas se faire dans de bonnes conditions et la ligne correspondant à la partie de code qui pose problème.  
  
Si vous m'avez suivi jusque là, félicitations ! Notre espace membre est terminé !  
  
*Bien-sur nous pouvons améliorer cet espace membre, le sécuriser, etc. mais ça, ce sera pour plus tard...*

**Etape 7. Le BackOffice - Gestion Boutique - Ajout de produits**

**Le BackOffice - Gestion Boutique - Ajout de produits**

Le BackOffice d'un site web est une interface de gestion pour les réglages, reservée seulement à l'administateur (ou aux administrateurs).  
  
Nous avons prévu une partie (dans le dossier **/admin/**) reservée à l'administrateur pour la gestion de son site web (produits, commandes, membres).  
  
Cette partie ne doit être accessible que par un internaute connecté ayant son statut fixé à 1 (autrement dis : un administrateur).  
  
Imaginez si 1 visiteur ou si 1 membre arrivait à accèder à cette partie, il pourrait supprimer des produits, modifier des prix, etc. ça pourrait vite être un carnage !  
  
Pour cela, nous allons utiliser la fonction utilisateurEstConnecteEtEstAdmin puisque nous avions prévu un code permettant de renvoyer TRUE (vrai tu est admin) ou FALSE (faux tu n'est pas admin).  
  
Nous prévoirons également un formulaire d'ajout de produit avec récupération des données en POST et requête d'enregistrement (INSERT) dans la base de données :

admin/gestion\_boutique.php

[?](file:///H:\IUSTE\PHP\Eprojet%20Php.htm)

|  |
| --- |
| <?php  require\_once("../inc/init.inc.php");  //--------------------------------- TRAITEMENTS PHP ---------------------------------//  //--- VERIFICATION ADMIN ---//  if(!internauteEstConnecteEtEstAdmin())  {      header("location:../connexion.php");      exit();  }  //--- ENREGISTREMENT PRODUIT ---//  if(!empty($\_POST))  {   // debug($\_POST);      $photo\_bdd = "";      if(!empty($\_FILES['photo']['name']))      {   // debug($\_FILES);          $nom\_photo = $\_POST['reference'] . '\_' .$\_FILES['photo']['name'];          $photo\_bdd = RACINE\_SITE . "photo/$nom\_photo";          $photo\_dossier = $\_SERVER['DOCUMENT\_ROOT'] . RACINE\_SITE . "/photo/$nom\_photo";          copy($\_FILES['photo']['tmp\_name'],$photo\_dossier);      }      foreach($\_POST as $indice => $valeur)      {          $\_POST[$indice] = htmlEntities(addSlashes($valeur));      }      executeRequete("INSERT INTO produit (id\_produit, reference, categorie, titre, description, couleur, taille, public, photo, prix, stock) values ('', '$\_POST[reference]', '$\_POST[categorie]', '$\_POST[titre]', '$\_POST[description]', '$\_POST[couleur]', '$\_POST[taille]', '$\_POST[public]',  '$photo\_bdd',  '$\_POST[prix]',  '$\_POST[stock]')");      $contenu .= '<div class="validation">Le produit a été ajouté</div>';  }  //--------------------------------- AFFICHAGE HTML ---------------------------------//  require\_once("../inc/haut.inc.php");  echo $contenu;  ?>  <h1> Formulaire Produits </h1>  <form method="post" enctype="multipart/form-data" action="">      <label for="reference">reference</label><br>      <input type="text" id="reference" name="reference" placeholder="la référence de produit"> <br><br>        <label for="categorie">categorie</label><br>      <input type="text" id="categorie" name="categorie" placeholder="la categorie de produit"><br><br>        <label for="titre">titre</label><br>      <input type="text" id="titre" name="titre" placeholder="le titre du produit"> <br><br>        <label for="description">description</label><br>      <textarea name="description" id="description" placeholder="la description du produit"></textarea><br><br>        <label for="couleur">couleur</label><br>      <input type="text" id="couleur" name="couleur" placeholder="la couleur du produit"> <br><br>        <label for="taille">Taille</label><br>      <select name="taille">          <option value="S">S</option>          <option value="M">M</option>          <option value="L">L</option>          <option value="XL">XL</option>      </select><br><br>        <label for="public">public</label><br>      <input type="radio" name="public" value="m" checked>Homme      <input type="radio" name="public" value="f">Femme<br><br>        <label for="photo">photo</label><br>      <input type="file" id="photo" name="photo"><br><br>        <label for="prix">prix</label><br>      <input type="text" id="prix" name="prix" placeholder="le prix du produit"><br><br>        <label for="stock">stock</label><br>      <input type="text" id="stock" name="stock" placeholder="le stock du produit"><br><br>        <input type="submit" value="enregistrement du produit">  </form>  <?php require\_once("../inc/bas.inc.php"); ?> |

**Résultat**  


**Quelques Explications**  
  
**Le contrôle admin**  
Nous demandons dès les premières lignes **if(!internauteEstConnecteEtEstAdmin())** si l'internaute n'est pas admin, alors nous effectuons une redirection vers la page de connexion.  
  
Nous en profitons pour mettre un exit() de manière à ce que l'interpréteur ne décode pas la suite du code et que la redirection se fasse immédiatement.  
  
  
**Le formulaire d'ajout de produit**  
La balise <form> comporte l'attribut enctype qui est indispensable pour permettre l'upload de fichier au sein de ce formulaire.  
  
Les tailles de produit ont été mises en lettre, mais cela peut être changé.  
  
  
**Le traitement PHP pour l'ajout de produit**  
Nous prévoyons un traitement PHP si le contenu du POST n'est pas vide if(!empty($\_POST)) (autrement dis s'il a été rempli par un clic sur le bouton submit).  
  
Nous initialisons une variable $photo\_bdd à vide pour éviter plus tard une erreur undefined si aucune photo n'est ajoutée.  
  
Un fichier transmis par upload ne se récupère pas avec $\_POST mais avec $\_FILES (il s'agit d'une autre superglobale).  
  
Si une photo a été uplaodée if(!empty($\_FILES['photo']['name'])), nous changeons le nom de la photo $nom\_photo = $\_POST['reference'] . '\_' .$\_FILES['photo']['name']; car par défaut 2 photos du même nom s'écrasent et se remplacent.  
Il est nécessaire que le dossier /photo/ soit existant sur le serveur.  
  
Il y a une sauvegarde du chemin de la photo dans la base de données, et une sauvegarde physique du fichier photo sur le serveur dans le dossier prévu à cet effet.  
  
Dans l'idéal, il serait intéressant d'engager plusieurs traitements sur une photo uplaodée :

* nom : donner 1 nom plus cohérent
* dimension : faire attention à ce qu'on ne vous envoie pas des photos miniatures ou au contraire une taille trop grande
* poids : des poids trop volumineux peuvent encombrer votre serveur
* extension : attention, on peut très bien vous envoyer des .exe ou des fichiers d'attaques) (même si effectivement l'administrateur n'est pas censé pirater son propre site)
* *etc.*

Faites des tests :  
- Tenter d'accèder à cette page d'administration en tant que visiteur, membre ou admin.  
- Ajouter des produits (avec photo de préférence) et vérifier les enregistrements de votre base (via PhpMyAdmin).

**Etape 7. Le BackOffice - Gestion Boutique - Affichage de produits**

Maintenant que vous avez pu ajouter des produits, nous allons les afficher sur la page web  
  
En effet, cela évitera d'aller dans PhpMyAdmin et surtout le commerçant pourra voir sa liste de produits en cliquant sur un lien prévu à cet effet.  
  


admin/gestion\_boutique.php

[?](file:///H:\IUSTE\PHP\Eprojet%20Php.htm)

|  |
| --- |
| <?php  require\_once("../inc/init.inc.php");  //--------------------------------- TRAITEMENTS PHP ---------------------------------//  //--- VERIFICATION ADMIN ---//  if(!internauteEstConnecteEtEstAdmin())  {      header("location:../connexion.php");      exit();  }  //--- ENREGISTREMENT PRODUIT ---//  if(!empty($\_POST))  {   // debug($\_POST);      $photo\_bdd = "";      if(!empty($\_FILES['photo']['name']))      {   // debug($\_FILES);          $nom\_photo = $\_POST['reference'] . '\_' .$\_FILES['photo']['name'];          $photo\_bdd = RACINE\_SITE . "photo/$nom\_photo";          $photo\_dossier = $\_SERVER['DOCUMENT\_ROOT'] . RACINE\_SITE . "/photo/$nom\_photo";          copy($\_FILES['photo']['tmp\_name'],$photo\_dossier);      }      foreach($\_POST as $indice => $valeur)      {          $\_POST[$indice] = htmlEntities(addSlashes($valeur));      }      executeRequete("INSERT INTO produit (id\_produit, reference, categorie, titre, description, couleur, taille, public, photo, prix, stock) values ('', '$\_POST[reference]', '$\_POST[categorie]', '$\_POST[titre]', '$\_POST[description]', '$\_POST[couleur]', '$\_POST[taille]', '$\_POST[public]',  '$photo\_bdd',  '$\_POST[prix]',  '$\_POST[stock]')");      $contenu .= '<div class="validation">Le produit a été ajouté</div>';  }  //--- LIENS PRODUITS ---//  $contenu .= '<a href="?action=affichage">Affichage des produits</a><br>';  $contenu .= '<a href="?action=ajout">Ajout d\'un produit</a><br><br><hr><br>';  //--- AFFICHAGE PRODUITS ---//  if(isset($\_GET['action']) && $\_GET['action'] == "affichage")  {      $resultat = executeRequete("SELECT \* FROM produit");        $contenu .= '<h2> Affichage des Produits </h2>';      $contenu .= 'Nombre de produit(s) dans la boutique : ' . $resultat->num\_rows;      $contenu .= '<table border="1"><tr>';        while($colonne = $resultat->fetch\_field())      {          $contenu .= '<th>' . $colonne->name . '</th>';      }      $contenu .= '<th>Modification</th>';      $contenu .= '<th>Supression</th>';      $contenu .= '</tr>';        while ($ligne = $resultat->fetch\_assoc())      {          $contenu .= '<tr>';          foreach ($ligne as $indice => $information)          {              if($indice == "photo")              {                  $contenu .= '<td><img src="' . $information . '" ="70" height="70"></td>';              }              else              {                  $contenu .= '<td>' . $information . '</td>';              }          }          $contenu .= '<td><a href="?action=modification&id\_produit=' . $ligne['id\_produit'] .'"><img src="../inc/img/edit.png"></a></td>';          $contenu .= '<td><a href="?action=suppression&id\_produit=' . $ligne['id\_produit'] .'" OnClick="return(confirm(\'En êtes vous certain ?\'));"><img src="../inc/img/delete.png"></a></td>';          $contenu .= '</tr>';      }      $contenu .= '</table><br><hr><br>';  }  //--------------------------------- AFFICHAGE HTML ---------------------------------//  require\_once("../inc/haut.inc.php");  echo $contenu;  if(isset($\_GET['action']) && ($\_GET['action'] == 'ajout'))  {      echo '      <h1> Formulaire Produits </h1>      <form method="post" enctype="multipart/form-data" action="">          <label for="reference">reference</label><br>          <input type="text" id="reference" name="reference" placeholder="la référence de produit"> <br><br>            <label for="categorie">categorie</label><br>          <input type="text" id="categorie" name="categorie" placeholder="la categorie de produit"><br><br>            <label for="titre">titre</label><br>          <input type="text" id="titre" name="titre" placeholder="le titre du produit"> <br><br>            <label for="description">description</label><br>          <textarea name="description" id="description" placeholder="la description du produit"></textarea><br><br>            <label for="couleur">couleur</label><br>          <input type="text" id="couleur" name="couleur" placeholder="la couleur du produit"> <br><br>            <label for="taille">Taille</label><br>          <select name="taille">              <option value="S">S</option>              <option value="M">M</option>              <option value="L">L</option>              <option value="XL">XL</option>          </select><br><br>            <label for="public">public</label><br>          <input type="radio" name="public" value="m" checked>Homme          <input type="radio" name="public" value="f">Femme<br><br>            <label for="photo">photo</label><br>          <input type="file" id="photo" name="photo"><br><br>            <label for="prix">prix</label><br>          <input type="text" id="prix" name="prix" placeholder="le prix du produit"><br><br>            <label for="stock">stock</label><br>          <input type="text" id="stock" name="stock" placeholder="le stock du produit"><br><br>            <input type="submit" value="enregistrement du produit">      </form>';  }  require\_once("../inc/bas.inc.php"); ?> |

**Quelques Explications**  
  
Nous avons prévue 2 liens permettant soit d'afficher les produits, soit d'ajouter 1 produit.  
  
Techniquement, cela permet de passer dans l'url **?action=affichage** ou **?action=ajout** et donc en récupérant l'information véhiculer dans l'url (via $\_GET), le site peut detecter l'action à déclencher.  
  
L'administrateur peut donc choisir l'action qu'il souhaite mettre en oeuvre : ajout ou affichage.  
  
L'affichage des produits se fait dans 1 table (1 tableau), si cela n'est pas clair [vous pouvez reconsulter cette partie en suivant ce lien](https://www.eprojet.fr/cours/php/418-php-requete-sql-avec-pdo-et-mysqli#php18.13).  
  
Nous ajoutons 2 liens (représentés d'images pour l'occasion) afin de proposer la modification et la suppression de produits pour l'administrateur (le commerçant).  
  
Pour la modification et la suppression, nous passerons par l'action par l'url ainsi que l'id du produit correspondant.  
  
L'action nous permettra de savoir que nous devons supprimer/modifier un produit et l'id nous permettra de savoir du quel il s'agit.

**Etape 7. Le BackOffice - Gestion Boutique - Suppression des produits**

Le commerçant (administrateur) peut ajouter et observer ses produits mais pour lui offrir une gestion complète il est important de lui proposer les options de suppression et de modification de produit.  
  
Commençons avec la suppression des produits :

admin/gestion\_boutique.php

[?](file:///H:\IUSTE\PHP\Eprojet%20Php.htm)

|  |
| --- |
| <?php  require\_once("../inc/init.inc.php");  //--------------------------------- TRAITEMENTS PHP ---------------------------------//  //--- VERIFICATION ADMIN ---//  if(!internauteEstConnecteEtEstAdmin())  {      header("location:../connexion.php");      exit();  }  //--- SUPPRESSION PRODUIT ---//  if(isset($\_GET['action']) && $\_GET['action'] == "suppression")  {   // $contenu .= $\_GET['id\_produit']      $resultat = executeRequete("SELECT \* FROM produit WHERE id\_produit=$\_GET[id\_produit]");      $produit\_a\_supprimer = $resultat->fetch\_assoc();      $chemin\_photo\_a\_supprimer = $\_SERVER['DOCUMENT\_ROOT'] . $produit\_a\_supprimer['photo'];      if(!empty($produit\_a\_supprimer['photo']) && file\_exists($chemin\_photo\_a\_supprimer)) unlink($chemin\_photo\_a\_supprimer);      executeRequete("DELETE FROM produit WHERE id\_produit=$\_GET[id\_produit]");      $contenu .= '<div class="validation">Suppression du produit : ' . $\_GET['id\_produit'] . '</div>';      $\_GET['action'] = 'affichage';  }  //--- ENREGISTREMENT PRODUIT ---//  if(!empty($\_POST))  {   // debug($\_POST);      $photo\_bdd = "";      if(!empty($\_FILES['photo']['name']))      {   // debug($\_FILES);          $nom\_photo = $\_POST['reference'] . '\_' .$\_FILES['photo']['name'];          $photo\_bdd = RACINE\_SITE . "photo/$nom\_photo";          $photo\_dossier = $\_SERVER['DOCUMENT\_ROOT'] . RACINE\_SITE . "/photo/$nom\_photo";          copy($\_FILES['photo']['tmp\_name'],$photo\_dossier);      }      foreach($\_POST as $indice => $valeur)      {          $\_POST[$indice] = htmlEntities(addSlashes($valeur));      }      executeRequete("INSERT INTO produit (id\_produit, reference, categorie, titre, description, couleur, taille, public, photo, prix, stock) values ('', '$\_POST[reference]', '$\_POST[categorie]', '$\_POST[titre]', '$\_POST[description]', '$\_POST[couleur]', '$\_POST[taille]', '$\_POST[public]',  '$photo\_bdd',  '$\_POST[prix]',  '$\_POST[stock]')");      $contenu .= '<div class="validation">Le produit a été enregistré</div>';  }  //--- LIENS PRODUITS ---//  $contenu .= '<a href="?action=affichage">Affichage des produits</a><br>';  $contenu .= '<a href="?action=ajout">Ajout d\'un produit</a><br><br><hr><br>';  //--- AFFICHAGE PRODUITS ---//  if(isset($\_GET['action']) && $\_GET['action'] == "affichage")  {      $resultat = executeRequete("SELECT \* FROM produit");        $contenu .= '<h2> Affichage des Produits </h2>';      $contenu .= 'Nombre de produit(s) dans la boutique : ' . $resultat->num\_rows;      $contenu .= '<table border="1"><tr>';        while($colonne = $resultat->fetch\_field())      {          $contenu .= '<th>' . $colonne->name . '</th>';      }      $contenu .= '<th>Modification</th>';      $contenu .= '<th>Supression</th>';      $contenu .= '</tr>';        while ($ligne = $resultat->fetch\_assoc())      {          $contenu .= '<tr>';          foreach ($ligne as $indice => $information)          {              if($indice == "photo")              {                  $contenu .= '<td><img src="' . $information . '" ="70" height="70"></td>';              }              else              {                  $contenu .= '<td>' . $information . '</td>';              }          }          $contenu .= '<td><a href="?action=modification&id\_produit=' . $ligne['id\_produit'] .'"><img src="../inc/img/edit.png"></a></td>';          $contenu .= '<td><a href="?action=suppression&id\_produit=' . $ligne['id\_produit'] .'" OnClick="return(confirm(\'En êtes vous certain ?\'));"><img src="../inc/img/delete.png"></a></td>';          $contenu .= '</tr>';      }      $contenu .= '</table><br><hr><br>';  }  //--------------------------------- AFFICHAGE HTML ---------------------------------//  require\_once("../inc/haut.inc.php");  echo $contenu;  if(isset($\_GET['action']) && ($\_GET['action'] == 'ajout'))  {      echo '      <h1> Formulaire Produits </h1>      <form method="post" enctype="multipart/form-data" action="">          <label for="reference">reference</label><br>          <input type="text" id="reference" name="reference" placeholder="la référence de produit"> <br><br>            <label for="categorie">categorie</label><br>          <input type="text" id="categorie" name="categorie" placeholder="la categorie de produit"><br><br>            <label for="titre">titre</label><br>          <input type="text" id="titre" name="titre" placeholder="le titre du produit"> <br><br>            <label for="description">description</label><br>          <textarea name="description" id="description" placeholder="la description du produit"></textarea><br><br>            <label for="couleur">couleur</label><br>          <input type="text" id="couleur" name="couleur" placeholder="la couleur du produit"> <br><br>            <label for="taille">Taille</label><br>          <select name="taille">              <option value="S">S</option>              <option value="M">M</option>              <option value="L">L</option>              <option value="XL">XL</option>          </select><br><br>            <label for="public">public</label><br>          <input type="radio" name="public" value="m" checked>Homme          <input type="radio" name="public" value="f">Femme<br><br>            <label for="photo">photo</label><br>          <input type="file" id="photo" name="photo"><br><br>            <label for="prix">prix</label><br>          <input type="text" id="prix" name="prix" placeholder="le prix du produit"><br><br>            <label for="stock">stock</label><br>          <input type="text" id="stock" name="stock" placeholder="le stock du produit"><br><br>            <input type="submit" value="enregistrement du produit">      </form>';  }  require\_once("../inc/bas.inc.php"); ?> |

**Quelques Explications**  
  
Nous detectons l'action suppression dans l'url if(isset($\_GET['action']) && $\_GET['action'] == "suppression").  
  
Nous allons d'abord selectionner toutes les informations sur ce produit dans la base, avec notamment le chemin vers la photo afin de la supprimer de notre serveur (puisque nous ne gérons pas l'attribution d'une même image pour plusieurs produits).  
  
Nous formulons une 2e requête pour supprimer réellement le produit de notre base et nous rebasculons vers l'action d'affichage pour observer tous les produits.

**Etape 7. Le BackOffice - Gestion Boutique - modification des produits**

Nous devons également proposer la modification de produits, pour cela nous aurons besoin d'un formulaire permettant d'accueillir les données en vue d'une modification.  
  
Pour factoriser le code (factoriser = réduire le code, comme ici on ré-utilise du code pour plusieurs choses différentes), nous pouvons nous servir d'un seul formulaire pour l'ajout et la modification.  
  
Et oui, qu'on ajoute ou que l'on modifie, il s'agit des mêmes champs.  
  
Seule différence, lorsque l'on modifie il nous faudra les champs de formulaire pré-saisis avec les informations actuelles.  
  
Voici une version modifiée :

admin/gestion\_boutique.php

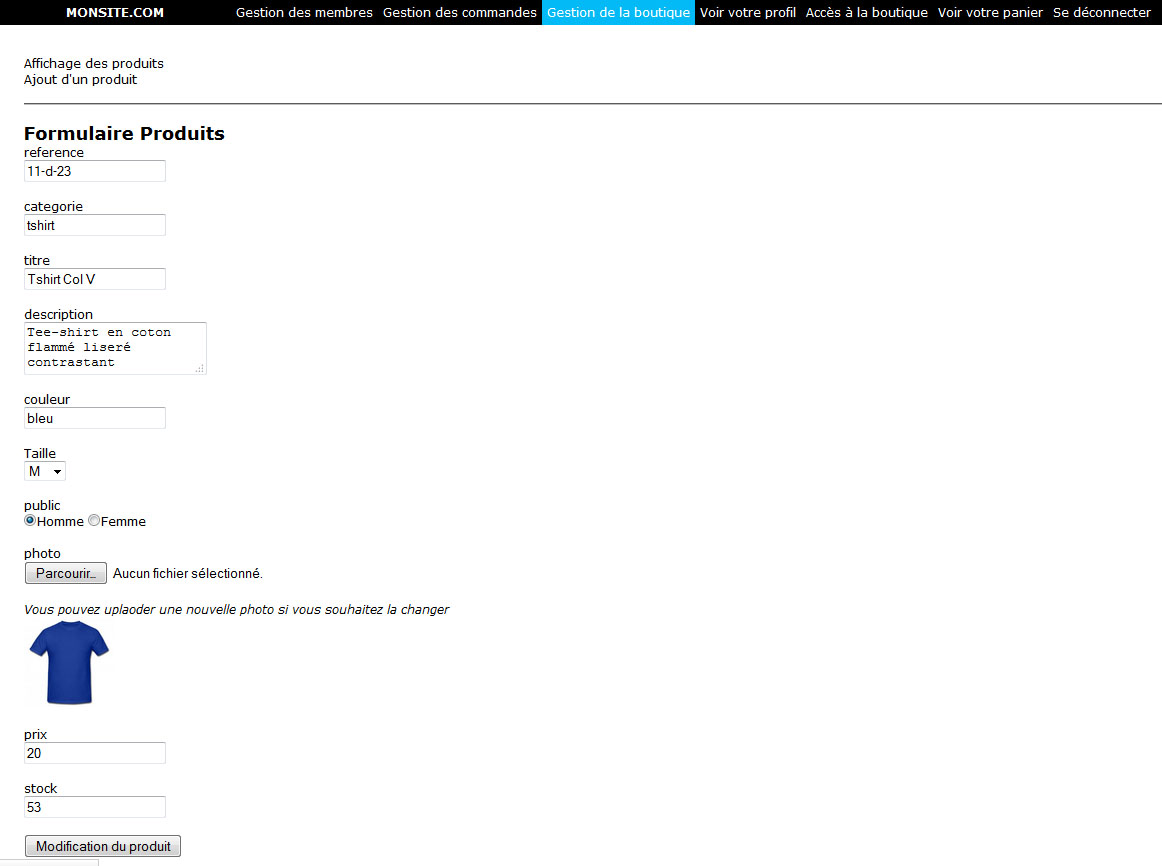
[?](file:///H:\IUSTE\PHP\Eprojet%20Php.htm)

|  |
| --- |
| <?php  require\_once("../inc/init.inc.php");  //--------------------------------- TRAITEMENTS PHP ---------------------------------//  //--- VERIFICATION ADMIN ---//  if(!internauteEstConnecteEtEstAdmin())  {      header("location:../connexion.php");      exit();  }    //--- SUPPRESSION PRODUIT ---//  if(isset($\_GET['action']) && $\_GET['action'] == "suppression")  {   // $contenu .= $\_GET['id\_produit']      $resultat = executeRequete("SELECT \* FROM produit WHERE id\_produit=$\_GET[id\_produit]");      $produit\_a\_supprimer = $resultat->fetch\_assoc();      $chemin\_photo\_a\_supprimer = $\_SERVER['DOCUMENT\_ROOT'] . $produit\_a\_supprimer['photo'];      if(!empty($produit\_a\_supprimer['photo']) && file\_exists($chemin\_photo\_a\_supprimer)) unlink($chemin\_photo\_a\_supprimer);      $contenu .= '<div class="validation">Suppression du produit : ' . $\_GET['id\_produit'] . '</div>';      executeRequete("DELETE FROM produit WHERE id\_produit=$\_GET[id\_produit]");      $\_GET['action'] = 'affichage';  }  //--- ENREGISTREMENT PRODUIT ---//  if(!empty($\_POST))  {   // debug($\_POST);      $photo\_bdd = "";      if(isset($\_GET['action']) && $\_GET['action'] == 'modification')      {          $photo\_bdd = $\_POST['photo\_actuelle'];      }      if(!empty($\_FILES['photo']['name']))      {   // debug($\_FILES);          $nom\_photo = $\_POST['reference'] . '\_' .$\_FILES['photo']['name'];          $photo\_bdd = RACINE\_SITE . "photo/$nom\_photo";          $photo\_dossier = $\_SERVER['DOCUMENT\_ROOT'] . RACINE\_SITE . "/photo/$nom\_photo";          copy($\_FILES['photo']['tmp\_name'],$photo\_dossier);      }      foreach($\_POST as $indice => $valeur)      {          $\_POST[$indice] = htmlEntities(addSlashes($valeur));      }      executeRequete("REPLACE INTO produit (id\_produit, reference, categorie, titre, description, couleur, taille, public, photo, prix, stock) values ('$\_POST[id\_produit]', '$\_POST[reference]', '$\_POST[categorie]', '$\_POST[titre]', '$\_POST[description]', '$\_POST[couleur]', '$\_POST[taille]', '$\_POST[public]',  '$photo\_bdd',  '$\_POST[prix]',  '$\_POST[stock]')");      $contenu .= '<div class="validation">Le produit a été ajouté</div>';      $\_GET['action'] = 'affichage';  }  //--- LIENS PRODUITS ---//  $contenu .= '<a href="?action=affichage">Affichage des produits</a><br>';  $contenu .= '<a href="?action=ajout">Ajout d\'un produit</a><br><br><hr><br>';  //--- AFFICHAGE PRODUITS ---//  if(isset($\_GET['action']) && $\_GET['action'] == "affichage")  {      $resultat = executeRequete("SELECT \* FROM produit");        $contenu .= '<h2> Affichage des produits </h2>';      $contenu .= 'Nombre de produit(s) dans la boutique : ' . $resultat->num\_rows;      $contenu .= '<table border="1" cellpadding="5"><tr>';        while($colonne = $resultat->fetch\_field())      {          $contenu .= '<th>' . $colonne->name . '</th>';      }      $contenu .= '<th>Modification</th>';      $contenu .= '<th>Supression</th>';      $contenu .= '</tr>';        while ($ligne = $resultat->fetch\_assoc())      {          $contenu .= '<tr>';          foreach ($ligne as $indice => $information)          {              if($indice == "photo")              {                  $contenu .= '<td><img src="' . $information . '" ="70" height="70"></td>';              }              else              {                  $contenu .= '<td>' . $information . '</td>';              }          }          $contenu .= '<td><a href="?action=modification&id\_produit=' . $ligne['id\_produit'] .'"><img src="../inc/img/edit.png"></a></td>';          $contenu .= '<td><a href="?action=suppression&id\_produit=' . $ligne['id\_produit'] .'" OnClick="return(confirm(\'En êtes vous certain ?\'));"><img src="../inc/img/delete.png"></a></td>';          $contenu .= '</tr>';      }      $contenu .= '</table><br><hr><br>';  }  //--------------------------------- AFFICHAGE HTML ---------------------------------//  require\_once("../inc/haut.inc.php");  echo $contenu;  if(isset($\_GET['action']) && ($\_GET['action'] == 'ajout' || $\_GET['action'] == 'modification'))  {      if(isset($\_GET['id\_produit']))      {          $resultat = executeRequete("SELECT \* FROM produit WHERE id\_produit=$\_GET[id\_produit]");          $produit\_actuel = $resultat->fetch\_assoc();      }      echo '      <h1> Formulaire Produits </h1>      <form method="post" enctype="multipart/form-data" action="">            <input type="hidden" id="id\_produit" name="id\_produit" value="'; if(isset($produit\_actuel['id\_produit'])) echo $produit\_actuel['id\_produit']; echo '">            <label for="reference">reference</label><br>          <input type="text" id="reference" name="reference" placeholder="la référence de produit" value="'; if(isset($produit\_actuel['reference'])) echo $produit\_actuel['reference']; echo '"><br><br>            <label for="categorie">categorie</label><br>          <input type="text" id="categorie" name="categorie" placeholder="la categorie de produit" value="'; if(isset($produit\_actuel['categorie'])) echo $produit\_actuel['categorie']; echo '" ><br><br>            <label for="titre">titre</label><br>          <input type="text" id="titre" name="titre" placeholder="le titre du produit" value="'; if(isset($produit\_actuel['titre'])) echo $produit\_actuel['titre']; echo '" > <br><br>            <label for="description">description</label><br>          <textarea name="description" id="description" placeholder="la description du produit">'; if(isset($produit\_actuel['description'])) echo $produit\_actuel['description']; echo '</textarea><br><br>            <label for="couleur">couleur</label><br>          <input type="text" id="couleur" name="couleur" placeholder="la couleur du produit"  value="'; if(isset($produit\_actuel['couleur'])) echo $produit\_actuel['couleur']; echo '"> <br><br>            <label for="taille">Taille</label><br>          <select name="taille">              <option value="S"'; if(isset($produit\_actuel) && $produit\_actuel['taille'] == 'S') echo ' selected '; echo '>S</option>              <option value="M"'; if(isset($produit\_actuel) && $produit\_actuel['taille'] == 'M') echo ' selected '; echo '>M</option>              <option value="L"'; if(isset($produit\_actuel) && $produit\_actuel['taille'] == 'L') echo ' selected '; echo '>L</option>              <option value="XL"'; if(isset($produit\_actuel) && $produit\_actuel['taille'] == 'XL') echo ' selected '; echo '>XL</option>          </select><br><br>            <label for="public">public</label><br>          <input type="radio" name="public" value="m"'; if(isset($produit\_actuel) && $produit\_actuel['public'] == 'm') echo ' checked '; elseif(!isset($produit\_actuel) && !isset($\_POST['public'])) echo 'checked'; echo '>Homme          <input type="radio" name="public" value="f"'; if(isset($produit\_actuel) && $produit\_actuel['public'] == 'f') echo ' checked '; echo '>Femme<br><br>            <label for="photo">photo</label><br>          <input type="file" id="photo" name="photo"><br><br>';          if(isset($produit\_actuel))          {              echo '<i>Vous pouvez uplaoder une nouvelle photo si vous souhaitez la changer</i><br>';              echo '<img src="' . $produit\_actuel['photo'] . '"  ="90" height="90"><br>';              echo '<input type="hidden" name="photo\_actuelle" value="' . $produit\_actuel['photo'] . '"><br>';          }            echo '          <label for="prix">prix</label><br>          <input type="text" id="prix" name="prix" placeholder="le prix du produit"  value="'; if(isset($produit\_actuel['prix'])) echo $produit\_actuel['prix']; echo '"><br><br>            <label for="stock">stock</label><br>          <input type="text" id="stock" name="stock" placeholder="le stock du produit"  value="'; if(isset($produit\_actuel['stock'])) echo $produit\_actuel['stock']; echo '"><br><br>            <input type="submit" value="'; echo ucfirst($\_GET['action']) . ' du produit">      </form>';  }  require\_once("../inc/bas.inc.php"); ?> |

**Quelques Explications**

flécheAttention  
Plusieurs lignes ont changé, si vous ne récupérez pas l'intégralité du code, à vous d'être vigilant et ne rien oublier.

Nous demandons à ce que le formulaire produit s'affiche également en cas de modification.  
  
Nous récupérons les informations actuelles d'un produit pour les disposer et les pré-saisir dans le formulaire afin que le commerçant puisse modifier seulement ce qu'il souhaite (sans pour autant perdre les autres informations).  
  
Nous utilisons l'action se trouvant dans l'url ($\_GET) pour définir le texte du bouton submit.  
  
Nous créons un champ hidden (caché) pour transmettre l'id du produit devant être modifié, cela n'aura pas d'impact en cas d'ajout car ce champ n'enverra pas d'id (puisque le produit n'existera pas encore au moment de la création).  
  
En ce qui concerne la photo, nous récupérons dans cet ordre : la photo actuelle du produit, et ensuite la photo qui aura été uplaodée (si elle a été uplaodé) pour écraser la photo actuelle.  
  
Nous faisons évoluer la requête SQL afin que celle-ci devienne REPLACE.  
flécheREPLACE se comporte en INSERT (ajout) s'il n'y a pas d'id\_produit connu  
flécheREPLACE se comporte en UPDATE (modification) s'il y a 1 id\_produit connu (transmis par le champ id\_produit caché en hidden)  
  
Autant dire que le code est assez diffus mais nous avons l'avantage d'utiliser 1 seul formulaire et 1 seule requête SQL pour 2 choses différentes (l'ajout et la modification).

**Résultat**  


**Etape 8. Le FrontOffice - La boutique**

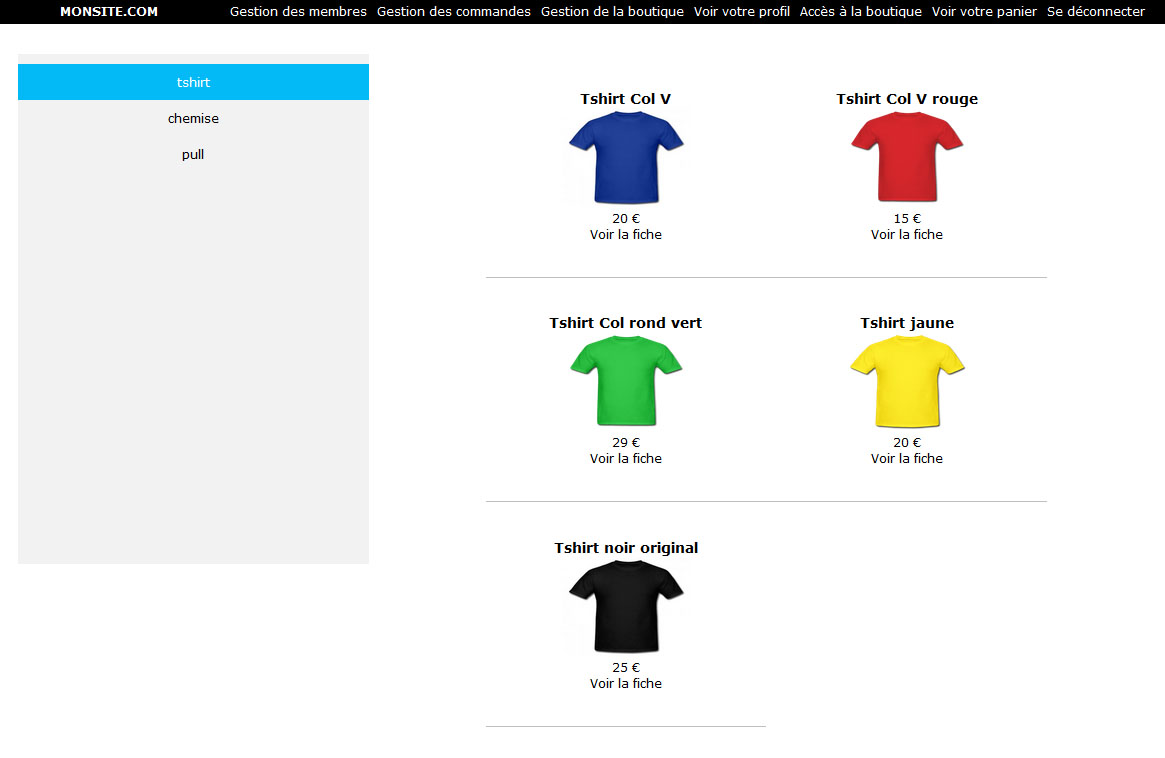
Maintenant que nous avons des produits en boutique, il faut que l'on puisse les exposer aux internautes dans la partie FRONT.  
  
Pour créer rééllement notre boutique, nous allons travailler dans le fichier : boutique.php

boutique.php

[?](file:///H:\IUSTE\PHP\Eprojet%20Php.htm)

|  |
| --- |
| <?php  require\_once("inc/init.inc.php");  //--------------------------------- TRAITEMENTS PHP ---------------------------------//  //--- AFFICHAGE DES CATEGORIES ---//  $categories\_des\_produits = executeRequete("SELECT DISTINCT categorie FROM produit");  $contenu .= '<div class="boutique-gauche">';  $contenu .= "<ul>";  while($cat = $categories\_des\_produits->fetch\_assoc())  {      $contenu .= "<li><a href='?categorie=" . $cat['categorie'] . "'>" . $cat['categorie'] . "</a></li>";  }  $contenu .= "</ul>";  $contenu .= "</div>";  //--- AFFICHAGE DES PRODUITS ---//  $contenu .= '<div class="boutique-droite">';  if(isset($\_GET['categorie']))  {      $donnees = executeRequete("select id\_produit,reference,titre,photo,prix from produit where categorie='$\_GET[categorie]'");      while($produit = $donnees->fetch\_assoc())      {          $contenu .= '<div class="boutique-produit">';          $contenu .= "<h2>$produit[titre]</h2>";          $contenu .= "<a href=\"fiche\_produit.php?id\_produit=$produit[id\_produit]\"><img src=\"$produit[photo]\" =\"130\" height=\"100\"></a>";          $contenu .= "<p>$produit[prix] €</p>";          $contenu .= '<a href="fiche\_produit.php?id\_produit=' . $produit['id\_produit'] . '">Voir la fiche</a>';          $contenu .= '</div>';      }  }  $contenu .= '</div>';  //--------------------------------- AFFICHAGE HTML ---------------------------------//  require\_once("inc/haut.inc.php");  echo $contenu;  require\_once("inc/bas.inc.php"); ?> |

**Quelques Explications**  
  
Nous commençons par selectionner les catégories de produits différents (DISTINCT) afin de les afficher dans une liste de liens.  
  
Si l'internaute clique sur l'une des catégories que nous proposons, nous passerons l'information dans l'url, par exemple ?categorie=tshirt.  
  
Lorsque l'url change, la page se recharge et le code va détecter qu'une information est passée dans l'url if(isset($\_GET['categorie']))  
  
Nous ferons donc une 2e requête SQL consistant à récupérer tous les produits relatifs à une catégorie dans la base (par exemple : tous les tshirts).  
  
Nous prévoyons également des liens sur les produits afin de proposer l'affichage d'un produit particulier dans une fiche produit dédiée.

**Résultat**  


Libre à vous d'adapter la présentation comme vous le souhaitez, voici le code CSS correspondant à notre résultat :

inc/css/style.css

[?](file:///H:\IUSTE\PHP\Eprojet%20Php.htm)

|  |
| --- |
| \*{ margin: 0; }  a{ text-decoration: none; color: #000; }  .conteneur{ margin: 0 auto; max-: 1170px; }  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* HAUT \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  header { background: #000000; padding: 5px; text-align: center; }  header  span{ color: #fff; font-weight: bold; text-transform: uppercase; margin-right: 5%; }  header  nav{ display: inline; }  header  a{ color: #ffffff; text-decoration: none; padding: 5px; }  header  nav  a:hover{ background: #04baf6; }  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* MILIEU \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  section{ padding: 30px; min-height: 800px; }  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* BAS \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  footer{ background: #000; color: white; text-align: center; padding: 7px 0;}  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* GENERAL \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  .erreur{ background: #ff0000; padding: 5px; margin: 5px; }  .validation{ background: #669933; padding: 5px; margin: 5px; }  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* BOUTIQUE \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  .boutique-gauche{ float: left; : 30%; background: #f2f2f2; min-height: 500px; margin-right: 10%; padding-top: 10px; text-align: center; }  .boutique-gauche  ul{ list-style: none; padding: 0; }  .boutique-gauche  ul  li  a{ display: block; padding: 10px; }  .boutique-gauche  ul  li  a:hover{ background: #04baf6; color: #fff; }  .boutique-droite{float: left; : 60%; }  .boutique-produit{ float: left; : 30%; text-align: center; padding: 5%; border-bottom: 1px solid #c0c0c0; }  .boutique-produit:hover{ background: #c0c0c0; } |

Vous pouvez ajouter un produit en BackOffice sur la page gestion\_boutique, et vous le verrez automatiquement sur la page produit en FrontOffice.  
C'est ce qu'on appelle de l'affichage dynamique !

**Etape 9. Boutique - La fiche produit**

Avec la fiche produit, nous allons créer une page qui permettra de visualiser plus en détail un produit.

fiche\_boutique.php

[?](file:///H:\IUSTE\PHP\Eprojet%20Php.htm)

|  |
| --- |
| <?php  require\_once("inc/init.inc.php");  //--------------------------------- TRAITEMENTS PHP ---------------------------------//  if(isset($\_GET['id\_produit']))  { $resultat = executeRequete("SELECT \* FROM produit WHERE id\_produit = '$\_GET[id\_produit]'"); }  if($resultat->num\_rows <= 0) { header("location:boutique.php"); exit(); }    $produit = $resultat->fetch\_assoc();  $contenu .= "<h2>Titre : $produit[titre]</h2><hr><br>";  $contenu .= "<p>Categorie: $produit[categorie]</p>";  $contenu .= "<p>Couleur: $produit[couleur]</p>";  $contenu .= "<p>Taille: $produit[taille]</p>";  $contenu .= "<img src='$produit[photo]' ='150' height='150'>";  $contenu .= "<p><i>Description: $produit[description]</i></p><br>";  $contenu .= "<p>Prix : $produit[prix] €</p><br>";    if($produit['stock'] > 0)  {      $contenu .= "<i>Nombre d'produit(s) disponible : $produit[stock] </i><br><br>";      $contenu .= '<form method="post" action="panier.php">';          $contenu .= "<input type='hidden' name='id\_produit' value='$produit[id\_produit]'>";          $contenu .= '<label for="quantite">Quantité : </label>';          $contenu .= '<select id="quantite" name="quantite">';              for($i = 1; $i <= $produit['stock'] && $i <= 5; $i++)              {                  $contenu .= "<option>$i</option>";              }          $contenu .= '</select>';          $contenu .= '<input type="submit" name="ajout\_panier" value="ajout au panier">';      $contenu .= '</form>';  }  else  {      $contenu .= 'Rupture de stock !';  }  $contenu .= "<br><a href='boutique.php?categorie=" . $produit['categorie'] . "'>Retour vers la séléction de " . $produit['categorie'] . "</a>";  //--------------------------------- AFFICHAGE HTML ---------------------------------//  require\_once("inc/haut.inc.php");  echo $contenu;  require\_once("inc/bas.inc.php"); ?> |

**Quelques Explications**  
  
Si l'id d'un produit se trouve dans l'url if(isset($\_GET['id\_produit'])), nous le récupérons via une requête SQL.  
  
Si nous ne récupérons pas de produit dans la base if($resultat->num\_rows <= 0), cela indique que le produit n'existe pas ou plus, nous redirigeons l'internaute vers la page boutique.  
  
Si l'interpréteur continue d'exécuter la suite des instructions, c'est que nous devons afficher le produit. Pour cela nous réalisons un traitement $produit = $resultat->fetch\_assoc(); afin de rendre le résultat exploitable sous forme de tableau ARRAY. Nous ne faisons pas de boucle dans la mesure où nous récupérons 1 seul enregistrement de la base de données.  
  
Nous affichons les informations du produit, toujours par l'intermédiaire de la variable $contenu, afin que cela n'apparaisse pas avant les premières balises type doctype, html, body, etc.  
  
Nous vérifions s'il y a du stock disponible afin de proposer la selection d'une quantité et l'ajout au panier.  
  
Si nous avons un stock disponible de 230 exemplaires pour un même produit, nous proposerons de prendre une quantité 5 par 5 (sauf si le stock est inférieur à 5, nous proposerons le stock restant).  
  
La boucle for intéroge la variable $stock et le 5 à chaque tour de boucle pour savoir si on ne le dépasse pas. cela prend donc le plus petit des deux nombres pour délimiter la boucle.  
  
En cliquant sur le bouton "ajout au panier" l'internaute sera envoyé vers la page panier.php.  
  
Si le produit n'a plus de stock disponible, nous informons l'internaute de la rupture.  
  
Pour terminer, si l'internaute ne souhaite pas acheter ce produit, nous proposons un lien permettant de le relancer dans notre boutique directement dans la categorie du produit courant

**Résultat**  


**Etape 10. Boutique - Le panier**

**Etape 10. Boutique - Le panier - Ajout de produit**

Maintenant que notre boutique existe, nous allons créer une fonctionnalité panier avec une simulation de paiement.  
  
Techniquement, nous allons créer un panier dans notre fichier de session.  
Nous gérons le panier dans un fichier de session afin que les informations soient gardées de page en page durant la navigation.  
Nous ne gérons pas cela en base de données car d'une part ces informations ne sont pas destinées à rester très longtemps, et d'autre part, beaucoup de paniers sont abandonnés (il est donc préférable de ne pas polluer la base avec des données temporaires, ni alourdir le site web avec des requêtes SQL trop gourmandes).  
  
Voici un schema représentatif de nos tableaux ARRAY multidimensionnels de session :

|  |  |
| --- | --- |
| **$\_SESSION['panier']** | |
| clé | valeur |
| titre | ARRAY |
| id\_produit | ARRAY |
| quantite | ARRAY |
| prix | ARRAY |

|  |  |
| --- | --- |
| **$\_SESSION['panier']['titre']** | |
| clé | valeur |
| 0 | Tshirt Rouge Col V |
| 1 | Chemise Blanche |
| 2 | Pull gris |

|  |  |
| --- | --- |
| **$\_SESSION['panier']['id\_produit']** | |
| clé | valeur |
| 0 | 2 |
| 1 | 6 |
| 2 | 8 |

|  |  |
| --- | --- |
| **$\_SESSION['panier']['quantite']** | |
| clé | valeur |
| 0 | 1 |
| 1 | 1 |
| 2 | 3 |

|  |  |
| --- | --- |
| **$\_SESSION['panier']['prix']** | |
| clé | valeur |
| 0 | 15 |
| 1 | 49 |
| 2 | 79 |

De cette manière, nous savons que le produit en position 1 dans chaque sous tableau array correspond à une chemise blanc, il s'agit de l'id\_produit n°6, avec une quantité de 1 et un prix de 49€.   
  
  
Avant de créer notre page panier.php, nous allons prévoir quelques fonctions afin de préparer au mieux cette fonctionnalité.  
  
Voici la mise à jour du fichier inc/fonction.inc.php

inc/fonction.inc.php

[?](file:///H:\IUSTE\PHP\Eprojet%20Php.htm)

|  |
| --- |
| function executeRequete($req)  {      global $mysqli;      $resultat = $mysqli->query($req);      if(!$resultat)      {          die("Erreur sur la requete sql.<br>Message : " . $mysqli->error . "<br>Code: " . $req);      }      return $resultat;  }  //------------------------------------  function debug($var, $mode = 1)  {      echo '<div style="background: orange; padding: 5px; float: right; clear: both; ">';      $trace = debug\_backtrace();      $trace = array\_shift($trace);      echo 'Debug demandé dans le fichier : $trace[file] à la ligne $trace[line].';      if($mode === 1)      {          print '<pre>'; print\_r($var); print '</pre>';      }      else      {          print '<pre>'; var\_dump($var); print '</pre>';      }      echo '</div>';  }  //------------------------------------  function internauteEstConnecte()  {      if(!isset($\_SESSION['membre'])) return false;      else return true;  }  //------------------------------------  function internauteEstConnecteEtEstAdmin()  {      if(internauteEstConnecte() && $\_SESSION['membre']['statut'] == 1) return true;      else return false;  }  //------------------------------------  function creationDuPanier()  {     if(!isset($\_SESSION['panier']))     {        $\_SESSION['panier'] = array();        $\_SESSION['panier']['titre'] = array();        $\_SESSION['panier']['id\_produit'] = array();        $\_SESSION['panier']['quantite'] = array();        $\_SESSION['panier']['prix'] = array();     }  }  //------------------------------------  function ajouterProduitDansPanier($titre, $id\_produit, $quantite, $prix)  {      creationDuPanier();      $position\_produit = array\_search($id\_produit,  $\_SESSION['panier']['id\_produit']);      if($position\_produit !== false)      {           $\_SESSION['panier']['quantite'][$position\_produit] += $quantite ;      }      else      {          $\_SESSION['panier']['titre'][] = $titre;          $\_SESSION['panier']['id\_produit'][] = $id\_produit;          $\_SESSION['panier']['quantite'][] = $quantite;          $\_SESSION['panier']['prix'][] = $prix;      }  }  //------------------------------------  function montantTotal()  {     $total=0;     for($i = 0; $i < count($\_SESSION['panier']['id\_produit']); $i++)     {        $total += $\_SESSION['panier']['quantite'][$i] \* $\_SESSION['panier']['prix'][$i];     }     return round($total,2);  }  //------------------------------------  function retirerProduitDuPanier($id\_produit\_a\_supprimer)  {      $position\_produit = array\_search($id\_produit\_a\_supprimer,  $\_SESSION['panier']['id\_produit']);      if ($position\_produit !== false)      {          array\_splice($\_SESSION['panier']['titre'], $position\_produit, 1);          array\_splice($\_SESSION['panier']['id\_produit'], $position\_produit, 1);          array\_splice($\_SESSION['panier']['quantite'], $position\_produit, 1);          array\_splice($\_SESSION['panier']['prix'], $position\_produit, 1);      }  }  //------------------------------------ |

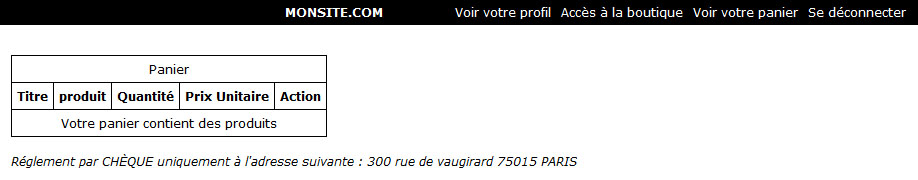
**Quelques Explications**  
  
**Creation du panier**  
  
Nous créons une première fonction creationDuPanier() afin de créer l'espace nécessaire dans le fichier de session pour accueillir les données des futurs produits.  
Nous vérifierons si la session existe déjà, et seulement dans le cas ou elle n'existe pas nous la créerons : if(!isset($\_SESSION['panier']))  
  
**Ajout d'un produit dans le panier**  
  
Nous exécutons d'abord creationDuPanier() afin que l'espace permettant d'accueillir les données produits soit créé.  
  
Notre fonction ajouterProduitDansPanier() est destinée à recevoir 4 arguments : le titre, l'id du produit, la quantité desirée et le prix *(ce sont les informations que nous garderons sur un produit dans notre panier).*  
  
Avec la fonction prédéfinie array\_search, nous allons d'abord procéder à une vérification. Avant de l'ajouter, est-ce que le produit est déjà présent dans le panier ?  
Par exemple, cela pourrait arriver si l'internaute ajoute le produit 2 (tshirt rouge) dans le panier, qu'il repart de la page panier, qu'il rerentre dans la fiche produit 2 et qu'il ajoute à nouveau un autre tshirt rouge (donc toujours le produit 2).  
Pour éviter d'avoir 2 lignes de produits différentes dans notre panier (comme s'il s'agissait de 2 produits différents), nous mettrons à jour uniquement la quantité du produit en question (déjà présente dans le panier).  
La fonction prédéfinie array\_search est pratique car elle nous renvoie précisément la position du produit dans le tableau ARRAY et nous savons donc à quel endroit intervenir pour adapter la quantité.  
Le symbole += permet de ne pas perdre la quantité précédente. S'il y avait 1 en quantité, et que l'internaute ajoute 2 autres tshirt rouges (toujours le produit n°2) et bien nous ne voulons pas remplacer le chiffre 1 par 2 mais faire l'opération suivante : 1 + 2 = 3.  
  
Dans le cas contraire, si l'id du produit n'est pas connue dans le panier, nous l'ajouterons normalement dans notre fichier session.  
  
**Montant Total**  
  
La fonction montantTotal() va nous permettre de multiplier chaque prix par chaque quantité et donc de faire la somme des prix contenus dans le panier.  
  
for($i = 0; $i < count($\_SESSION['panier']['id\_produit']); $i++) tant que $i est inférieur au nombre de produits contenus dans la session panier, nous allons parcourir les éléments.  
  
$total += $\_SESSION['panier']['quantite'][$i] \* $\_SESSION['panier']['prix'][$i]; on multiplie la quantite par le prix; ex 1\*10€ ou 3\*10€ sans remplacer pour autant la derniere valeur contenue dans la variable $total (+=). Addition et multiplication.  
  
return round($total,2); nous retournons le prix total pour tous les produits (avec round nous demandons à avoir 2 chiffres après la virgule)  
  
**retirerProduitDuPanier**  
  
Cette fonction permet de retirer un produit du panier.  
La fonction prédéfinie Array\_Search retourne un chiffre afin de savoir à quel indice se trouve le produit à supprimer du panier. Lorsqu'un produit est retiré du panier, son indice dans les sous tableaux array titre, id\_produit, quantite et prix vont être vidés.  
Pour éviter de garder des indices vides dans nos tableaux ARRAY, nous nous servons de la fonction prédéfinie array\_splice pour faire glisser les produits des indices supérieurs vers des indices numériques du tableau inférieur.  
  
  
Voici notre page panier qui exploitera notre fichier de fonction :

panier.php

[?](file:///H:\IUSTE\PHP\Eprojet%20Php.htm)

|  |
| --- |
| <?php  require\_once("inc/init.inc.php");  //--------------------------------- TRAITEMENTS PHP ---------------------------------//  //--- AJOUT PANIER ---//  if(isset($\_POST['ajout\_panier']))  {   // debug($\_POST);      $resultat = executeRequete("SELECT \* FROM produit WHERE id\_produit='$\_POST[id\_produit]'");      $produit = $resultat->fetch\_assoc();      ajouterProduitDansPanier($produit['titre'],$\_POST['id\_produit'],$\_POST['quantite'],$produit['prix']);  }  //--------------------------------- AFFICHAGE HTML ---------------------------------//  include("inc/haut.inc.php");  echo $contenu;  echo "<table border='1' style='border-collapse: collapse' cellpadding='7'>";  echo "<tr><td colspan='5'>Panier</td></tr>";  echo "<tr><th>Titre</th><th>Produit</th><th>Quantité</th><th>Prix Unitaire</th></tr>";  if(empty($\_SESSION['panier']['id\_produit'])) // panier vide  {      echo "<tr><td colspan='5'>Votre panier est vide</td></tr>";  }  else  {      echo "<tr><td colspan='5'>Votre panier contient des produits</td></tr>";  }  echo "</table><br>";  echo "<i>Réglement par CHÈQUE uniquement à l'adresse suivante : 300 rue de vaugirard 75015 PARIS</i><br>";  echo "<hr>session panier:<br>"; debug($\_SESSION);  include("inc/bas.inc.php");  ?> |

**Quelques Explications**  
  
Si le POST a été réalisé if(isset($\_POST['ajout\_panier'])), les informations proviennent de la page fiche\_produit.php  
  
Nous réalisons une requête SQL pour récupérer des informations sur le produit.  
  
Nous nous servons de la fonction ajouterProduitDansPanier qui entraine l'exécution de la fonction creationDuPanier.  
  
Le prix et le titre viennent de la base de données par l'intermédiaire de la variable $produit, tandis que l'id\_produit et la quantite proviennent de la fiche\_produit.php.  
  
Un peu plus bas dans le code, nous regardons si le produit est vide ou non et adressons un message à l'internaute.

**Résultat**  


**Etape 10. Boutique - Le panier - Affichage des produits et enregistrement de la commande**

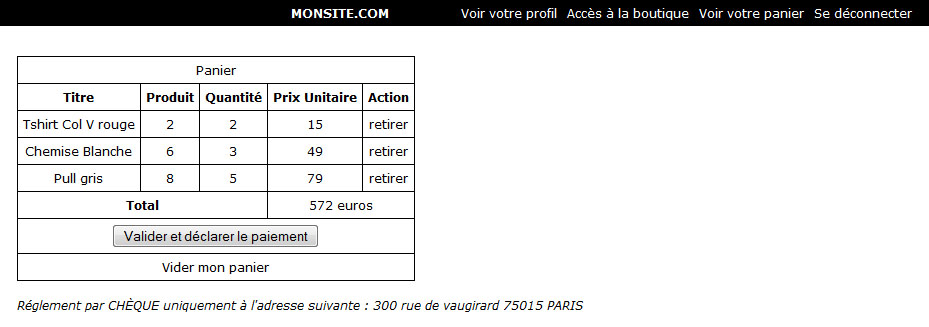
Si nous avons réussi à rentrer des produits dans notre panier, nous allons maintenant les afficher afin que l'internaute puisse les observer sur la page web.

panier.php

[?](file:///H:\IUSTE\PHP\Eprojet%20Php.htm)

|  |
| --- |
| <?php  require\_once("inc/init.inc.php");  //--------------------------------- TRAITEMENTS PHP ---------------------------------//  //--- AJOUT PANIER ---//  if(isset($\_POST['ajout\_panier']))  {   // debug($\_POST);      $resultat = executeRequete("SELECT \* FROM produit WHERE id\_produit='$\_POST[id\_produit]'");      $produit = $resultat->fetch\_assoc();      ajouterProduitDansPanier($produit['titre'],$\_POST['id\_produit'],$\_POST['quantite'],$produit['prix']);  }  //--------------------------------- AFFICHAGE HTML ---------------------------------//  include("inc/haut.inc.php");  echo $contenu;  echo "<table border='1' style='border-collapse: collapse' cellpadding='7'>";  echo "<tr><td colspan='5'>Panier</td></tr>";  echo "<tr><th>Titre</th><th>Produit</th><th>Quantité</th><th>Prix Unitaire</th></tr>";  if(empty($\_SESSION['panier']['id\_produit'])) // panier vide  {      echo "<tr><td colspan='5'>Votre panier est vide</td></tr>";  }  else  {      for($i = 0; $i < count($\_SESSION['panier']['id\_produit']); $i++)      {          echo "<tr>";          echo "<td>" . $\_SESSION['panier']['titre'][$i] . "</td>";          echo "<td>" . $\_SESSION['panier']['id\_produit'][$i] . "</td>";          echo "<td>" . $\_SESSION['panier']['quantite'][$i] . "</td>";          echo "<td>" . $\_SESSION['panier']['prix'][$i] . "</td>";          echo "</tr>";      }      echo "<tr><th colspan='3'>Total</th><td colspan='2'>" . montantTotal() . " euros</td></tr>";      if(internauteEstConnecte())      {          echo '<form method="post" action="">';          echo '<tr><td colspan="5"><input type="submit" name="payer" value="Valider et déclarer le paiement"></td></tr>';          echo '</form>';      }      else      {          echo '<tr><td colspan="3">Veuillez vous <a href="inscription.php">inscrire</a> ou vous <a href="connexion.php">connecter</a> afin de pouvoir payer</td></tr>';      }      echo "<tr><td colspan='5'><a href='?action=vider'>Vider mon panier</a></td></tr>";  }  echo "</table><br>";  echo "<i>Réglement par CHÈQUE uniquement à l'adresse suivante : 300 rue de vaugirard 75015 PARIS</i><br>";  echo "<hr>session panier:<br>"; debug($\_SESSION);  include("inc/bas.inc.php");  ?> |

**Quelques Explications**  
  
Nous faisons plusieurs tours avec la boucle for afin de parcourir tous les produits présents dans notre panier.  
  
L'affichage se fait par l'intermédiaire de la superglobale $\_SESSION (puisque toutes les informations sont enregistrées dans le fichier de session).  
  
Faites des tests et remplissez votre panier avec plusieurs produits et plusieurs quantités !

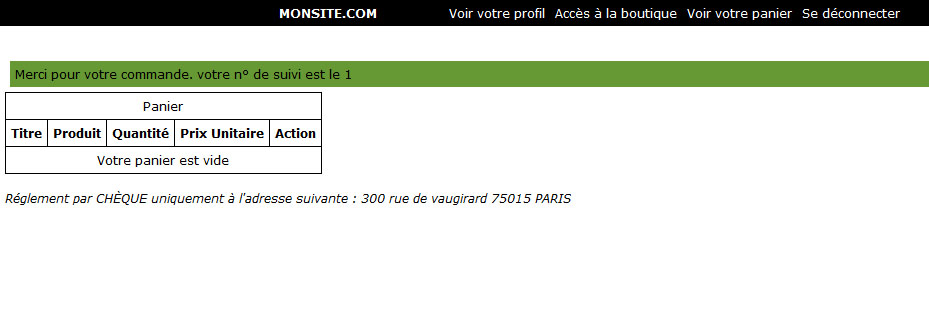
**Résultat**  


Nous prévoyons un formulaire permettant de déclarer le paiement avec un bouton submit "Valider et déclarer le paiement".  
  
1 autre lien est disponible pour vider l'ensemble du panier.  
  
Voici les traitements associés :

panier.php

[?](file:///H:\IUSTE\PHP\Eprojet%20Php.htm)

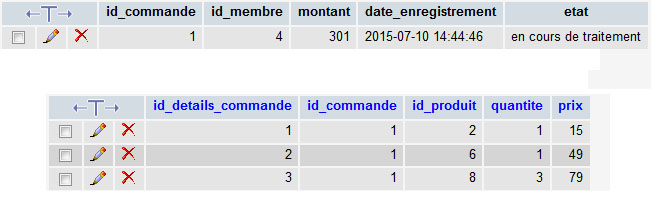
|  |
| --- |
| <?php  require\_once("inc/init.inc.php");  //--------------------------------- TRAITEMENTS PHP ---------------------------------//  //--- AJOUT PANIER ---//  if(isset($\_POST['ajout\_panier']))  {   // debug($\_POST);      $resultat = executeRequete("SELECT \* FROM produit WHERE id\_produit='$\_POST[id\_produit]'");      $produit = $resultat->fetch\_assoc();      ajouterProduitDansPanier($produit['titre'],$\_POST['id\_produit'],$\_POST['quantite'],$produit['prix']);  }  //--- VIDER PANIER ---//  if(isset($\_GET['action']) && $\_GET['action'] == "vider")  {      unset($\_SESSION['panier']);  }  //--- PAIEMENT ---//  if(isset($\_POST['payer']))  {      for($i=0 ;$i < count($\_SESSION['panier']['id\_produit']) ; $i++)      {          $resultat = executeRequete("SELECT \* FROM produit WHERE id\_produit=" . $\_SESSION['panier']['id\_produit'][$i]);          $produit = $resultat->fetch\_assoc();          if($produit['stock'] < $\_SESSION['panier']['quantite'][$i])          {              $contenu .= '<hr><div class="erreur">Stock Restant: ' . $produit['stock'] . '</div>';              $contenu .= '<div class="erreur">Quantité demandée: ' . $\_SESSION['panier']['quantite'][$i] . '</div>';              if($produit['stock'] > 0)              {                  $contenu .= '<div class="erreur">la quantité de l\'produit ' . $\_SESSION['panier']['id\_produit'][$i] . ' à été réduite car notre stock était insuffisant, veuillez vérifier vos achats.</div>';                  $\_SESSION['panier']['quantite'][$i] = $produit['stock'];              }              else              {                  $contenu .= '<div class="erreur">l\'produit ' . $\_SESSION['panier']['id\_produit'][$i] . ' à été retiré de votre panier car nous sommes en rupture de stock, veuillez vérifier vos achats.</div>';                  retirerProduitDuPanier($\_SESSION['panier']['id\_produit'][$i]);                  $i--;              }              $erreur = true;          }      }      if(!isset($erreur))      {          executeRequete("INSERT INTO commande (id\_membre, montant, date\_enregistrement) VALUES (" . $\_SESSION['membre']['id\_membre'] . "," . montantTotal() . ", NOW())");          $id\_commande = $mysqli->insert\_id;          for($i = 0; $i < count($\_SESSION['panier']['id\_produit']); $i++)          {              executeRequete("INSERT INTO details\_commande (id\_commande, id\_produit, quantite, prix) VALUES ($id\_commande, " . $\_SESSION['panier']['id\_produit'][$i] . "," . $\_SESSION['panier']['quantite'][$i] . "," . $\_SESSION['panier']['prix'][$i] . ")");          }          unset($\_SESSION['panier']);          mail($\_SESSION['membre']['email'], "confirmation de la commande", "Merci votre n° de suivi est le $id\_commande", "From:vendeur@dp\_site.com");          $contenu .= "<div class='validation'>Merci pour votre commande. votre n° de suivi est le $id\_commande</div>";      }  }    //--------------------------------- AFFICHAGE HTML ---------------------------------//  include("inc/haut.inc.php");  echo $contenu;  echo "<table border='1' style='border-collapse: collapse' cellpadding='7'>";  echo "<tr><td colspan='5'>Panier</td></tr>";  echo "<tr><th>Titre</th><th>Produit</th><th>Quantité</th><th>Prix Unitaire</th></tr>";  if(empty($\_SESSION['panier']['id\_produit'])) // panier vide  {      echo "<tr><td colspan='5'>Votre panier est vide</td></tr>";  }  else  {      for($i = 0; $i < count($\_SESSION['panier']['id\_produit']); $i++)      {          echo "<tr>";          echo "<td>" . $\_SESSION['panier']['titre'][$i] . "</td>";          echo "<td>" . $\_SESSION['panier']['id\_produit'][$i] . "</td>";          echo "<td>" . $\_SESSION['panier']['quantite'][$i] . "</td>";          echo "<td>" . $\_SESSION['panier']['prix'][$i] . "</td>";          echo "</tr>";      }      echo "<tr><th colspan='3'>Total</th><td colspan='2'>" . montantTotal() . " euros</td></tr>";      if(internauteEstConnecte())      {          echo '<form method="post" action="">';          echo '<tr><td colspan="5"><input type="submit" name="payer" value="Valider et déclarer le paiement"></td></tr>';          echo '</form>';      }      else      {          echo '<tr><td colspan="3">Veuillez vous <a href="inscription.php">inscrire</a> ou vous <a href="connexion.php">connecter</a> afin de pouvoir payer</td></tr>';      }      echo "<tr><td colspan='5'><a href='?action=vider'>Vider mon panier</a></td></tr>";  }  echo "</table><br>";  echo "<i>Réglement par CHÈQUE uniquement à l'adresse suivante : 300 rue de vaugirard 75015 PARIS</i><br>";  // echo "<hr>session panier:<br>"; debug($\_SESSION);  include("inc/bas.inc.php");  ?> |

**Résultat**  


**Quelques Explications**  
  
Pour tous les produits dans le panier, nous allons observer si les quantités demandées sont encore en stock.  
Pour cela, nous devons formuler une requête SQL qui va chercher en base les informations sur 1 produit, nous devons faire ça pour tous les produits c'est la raison pour laquelle la requête SQL se trouve dans une boucle afin d'entamer une répétition sur le traitement.  
Logiquement cela devrait être le cas mais imaginez la situation suivante :  
Le produit n° 2 tshirt rouge est présent 2 fois en stock.  
Julien arrive a 10h07 pour en prendre 2 dans son panier.  
Olivier arrive a 10h08 pour en prendre 1 dans son panier.  
A 10h10, Olivier valide son panier et son achat du tshirt rouge.  
A 10h20 Julien souhaite valider l'achat de ses 2 tshirts mais il n'en reste plus assez puisque le site web en aura vendu 1 à Olivier quelques minutes avant. Par conséquent Julien pourra acheter qu'1 seul tshirt rouge.  
Comme dans un magasin physique, le produit ne vous appartient pas au moment où vous le mettez dans votre panier mais bien quand vous avez payé en caisse. C'est bien au moment du paiement qu'il faut vérifier la disponibilité du produit (et non pas au moment de l'ajout dans le panier)  
  
Voici 3 situations qui peuvent arriver :

* Si la quantité demandée est couverte par le stock, il n'y a pas de soucis.
* Si la quantité demandée est inférieure au stock mais qu'il reste du stock, nous diminuons la quantité demandée par le stock restant (afin de vendre les derniers exemplaires).  
  l'internaute demande 6 tshirts, il en reste 2. Si le stock est supérieur à zéro, on lui remplace le 6 par un 2.
* Si la quantité demandée est inférieure au stock et qu'il n'y a plus de stock, nous retirons totalement le produit du panier.  
  sinon, nous sommes en rupture de stock, on lui retire carrèment le produit. On ne va pas remplacer son 6 par un 0

Nous aurions pu faire une boucle à l'envers au niveau du code $i--;. Nous avons choisi d'ajouter un retour dans la boucle, voici l'explication :  
si le produit en indice zéro pose problème pour cause de rupture de stock, il est retiré.  
Du coup le produit en indice 1 passe en indice 0 (réordonner via array splice de la fonction retirerproduitDuPanier) et ne sera pas vérifié (car le tour de boucle 0 sera passé, nous serons à 1).  
Il faut donc que je reparcours l'indice 0 pour me préoccuper du produit qui vient de se décaler à l'indice 0 et qui n'est pas encore testé.  
Ceci explique la présence du code $i--;.  
  
Quoi qu'il en soit (stock réduit ou rupture totale de stock), s'il y a eu un souci dans le traitement, nous mettrons la variable $erreur à TRUE $erreur = true; N'hesitez pas à faire des tests, vous prenez le navigateur firefox en tant que membre et trouvez un produit avec seulement 3 exemplaires en stock.  
Vous ajoutez les 3 derniers exemplaires dans votre panier.  
Dans le même temps, vous prenez le navigateur chrome en tant qu'admin (pour simuler la présence de l'administrateur sur un autre ordinateur), modifiez le stock de ce même produit à 1 (car on imagine que vous venez d'en vendre 2).  
Revenez sur Firefox en tant que membre et tentez de valider votre panier pour voir si cela vous bloque et vous informe bien du stock réduit.  
  
Pour traiter et enregistrer la commande, la variable $erreur ne doit pas exister if(!isset($erreur)) (sinon cela voudrait dire que nous sommes passés dans le stock réduit ou dans la rupture de stock totale).  
  
Nous enregistrons la commande dans la table commande : l'id du membre est récupéré dans le fichier de session (membre actuellement connecté), le prix est calculé par notre fonction montantTotal(), la date d'enregistrement est générée via la fonction SQL NOW()  
  
Pour une commande il peut y avoir plusieurs produits et donc plusieurs lignes d'informations (dans le cas où il s'agit d'une commande avec plusieurs produits différents).  
Nous récupérons donc l'id de la commande (qui a été générée en AUTO\_INCREMENT) : $id\_commande = $mysqli->insert\_id;  
Il y a 1 seule commande mais plusieurs détails de commande à enregistrer, nous refaisons donc une boucle pour tous les enregistrer en les reliant au numéro de commande principale.  
  
Nous pourrions réalisé la décrémentation du stock mais le paiement se déroule par cheque alors nous attendrons que l'administrauter reçoive ce cheque et décrémente lui même son stock via le BackOffice.  
  
Une fois que l'internaute à déclaré le paiement, nous vidons son panier car tous les produits sont censés être payés !  
  
Nous envoyons un email de confirmation au membre qui vient de réaliser la commande.  
  
Au sein de ce fichier, nous mélangeons à la fois des instructions d'affichage echo et la variable $contenu.  
La variable $contenu correspond à des affichages que nous devons faire lors des traitements mais que l'on doit retarder pour garder une structure viable.

**Résultat**  
  
  
*il y a bien 1 commande principale et 3 produits rattachés à cette commande*.

Si vous en êtes arrivés jusque-là, je vous dis BRAVO !  
  
Cette première version représente une base ré-utilisable et surtout un rassemblement de tous les sujets que nous avons pu voir durant le cours PHP, dans un contexte précis de site web.  
  
Bien entendu cela doit être amélioré, testé, sécurisé et continué d'être développé. Ca prend du temps de créer 1 site web complet !

**Entrainement - Développer davantage votre site et vos connaissances**

Pour d'avantage développer notre site, je vous propose plusieurs suggestions d'exerices pour être plus performant :

**Les développements complémentaires pour faire évoluer le site**

**Exercice 1 - BackOffice / Gestion des commandes**

Ajoutez (dans le dossier /admin/) une page que vous nommerez « gestion\_commandes.php » dans le dossier « /admin » afin d’afficher toutes les commandes passées sur le site ainsi que le détail des commandes.  
L’affichage de toutes les informations aideront le commerçant :

* Le numéro de commande (id\_commande) ainsi que la date et le montant afin que le commerçant puisse faire le rapprochement avec les chèques qu’il reçoit.
* Les informations sur les articles commandés (id\_article) ainsi que le titre, la photo, la quantité demandée, etc. afin qu’il puisse honorer la commande du client.
* Le numéro du membre (id\_membre) ainsi que son pseudo, adresse, ville et cp afin que le commerçant puisse envoyer le colis.

L’état de la commande doit s’afficher et doit pouvoir être modifiable par l’administrateur (s’il souhaite changer une commande jusqu’à présent « en cours de traitement » par « en cours d’envoi » par exemple).  
L’idéal serait que le membre reçoive un email de notification pour l’informer de l’évolution de sa commande.  
  
L’affichage du chiffre d’affaires doit également apparaitre sur cette page.  
De l’accessibilité et de la navigation seraient un plus pour le commerçant, en effet il serait bon de lui offrir la possibilité de trier les commandes par date, état, montant, etc.  
  
Seul l’administrateur doit avoir accès à cette page.

**Exercice 2 - BackOffice / Gestion des membres**

Ajoutez une page que vous nommerez « gestion\_membres.php » dans le dossier « /admin » afin d’afficher tous les membres inscrits sur le site. Seul l’administrateur doit avoir accès à cette page.  
  
Dans cette même page, ajoutez la possibilité à l’administrateur de pouvoir supprimer un membre inscrit au site.  
  
Donnez-lui la possibilité d’ajouter d’autres comptes « administrateur ».

**Exercice 3 - FrontOffice / Modification du compte membre**

Ajoutez un lien « mettre à jour mes informations » dans l’espace « profil.php » afin que les personnes inscrites au site aient la possibilité de modifier leurs informations relatives à leurs comptes tels que leur adresse, mot de passe, ville, cp, adresse, etc.  
Cela implique la création d’une nouvelle page « membres.php ».

**Exercice 4 - FrontOffice / Suppression du compte membre**

Ajoutez un lien « se désinscrire » dans l’espace « profil.php » afin que les personnes inscrites au site aient la possibilité de supprimer leur compte membre.

**Exercice 5 - FrontOffice / Améioration de l’espace membre**

Donner la possibilité aux membres d’avoir des avatars de profil.  
Permettre à l’internaute de solliciter le site en cas de mot de passe perdu.  
Sur la page « profil.php », affichez le suivi des commandes en cours, l’historique des commandes passées, etc.  
Permettre à l’internaute de pouvoir se connecter avec son pseudo ou son adresse email.

**Exercice 6 - FrontOffice / Boutique**

Sur la page « boutique.php » :  
Effectuer un moteur de recherche par mots-clés avec une suggestion de saisie.  
Effectuer une zone de recherche permettant de trier par tranche de prix.  
  
Pour la sélection des produits, prévoyez une pagination s’il y en a beaucoup afin de les afficher par tranche de 10 produits.

**Exercice 7 - Suggestion de produits**

Sur la page « fiche\_produit.php », affichez une suggestion de produits similaires au produit qui est sélectionné par l’internaute afin de faciliter les recherches croisées.  
  
Sur la page « profil.php », affichez une suggestion de produits similaires aux produits que l'internaute a déjà achetés.  
Dans le cas où il n’a pas encore acheté sur notre site, affichez une suggestion de produits correspondants à la dernière catégorie de produits qu’il a consultés sur notre boutique.

**Exercice 8 - Amélioration du panier**

Sur la page « panier.php », affichez le prix HT et TTC unitaire et aussi multiplié par le nombre de produits.  
Affichez le titre de du produit, ainsi que la photo à coté des autres informations déjà présentes.  
Donner la possibilité à l'internaute d’augmenter ou réduire les quantités directement sur la page panier.  
  
Nous allons améliorer notre système permettant de retirer ou baisser les articles si les quantités demandées par notre panier sont supérieures à notre stock restant en base de données.  
Pour cela, il va falloir modifier la page « fiche\_produit.php ».  
Actuellement, s’il reste 3 jeans, l’internaute peut prendre les 3 derniers et les mettre dans son panier.  
En revanche, il peut revenir sur la fiche détaillée de cet article et les reprendre 3 fois autant de fois qu’il le souhaite.  
Nous allons faire en sorte que si l’internaute ait déjà cet article 1 fois dans son panier, le site lui propose d’en prendre 2 seulement. Si l’internaute en possède déjà 2 dans son panier, le site lui propose d’en prendre 1 supplémentaire seulement.  
Dans le panier, il y aura moins de risques pour l’internaute de demander des quantités supérieures au stock restant en base de données.

**Exercice 9 - Amélioration du systeme de commande**

Intégrez la solution paypal sandbox pour réaliser des tests de paiement. Ce qui permettra d'aller plus loin qu'une simple simulation de paiement par chèque  
Prevoyez de générer des factures PDF avec les produits commandés par l'internaute.

**Exercice 10 - Amélioration du graphisme**

Afin d’avoir un meilleur retour des internautes, nous allons améliorer le visuel :   
Intégrez un design dans le contexte du site.  
Ajoutez de l’animation et des effets pour éviter d’avoir un site figé où rien ne bouge.