

Réalisation D'un Site Web Complet

Sommaire

19. Réalisation D'un Site Web Complet

- 1- Étape 1 : Modélisation et création de la base de données, table et champs.
- 2- Étape 2. Création d'une arborescence
- 3- Étape 3. Écriture des fichiers en inclusion
- 4- Étape 4. Écriture de la page inscription.php
- 5- Étape 5. Écriture de la page connexion.php
- 6- Étape 6. Écriture de la page profil.php
- 7- Étape 7. Le BackOffice - Gestion Boutique - Ajout de produits
- 8- Étape 8. Le FrontOffice - La boutique
- 9- Étape 9. Boutique - La fiche produit
- 10- Étape 10. Boutique - Le panier
- 11- Les développements complémentaires pour faire évoluer le site

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

Pour le prochain exemple, nous allons réaliser la base d'un site web procédural avec une tendance e-commerce. Ce site aura le mérite de nous faire pratiquer le CRUD(Create Read Update Delete)

Le CRUD représente la base en PHP. Il s'agit de savoir librement consulter, insérer, modifier ou supprimer des données.

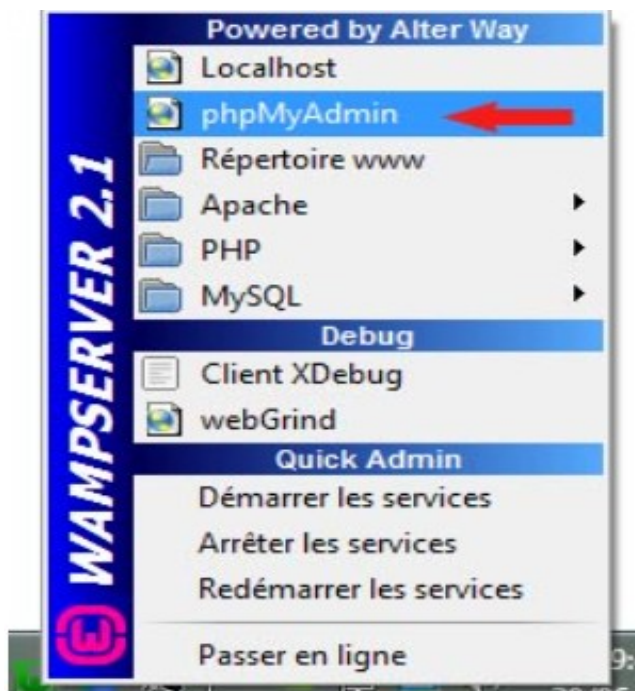
Étape 1 : Modélisation et création de la base de données, table et champs.

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.
prénom	Varchar	20		Ce champ correspond au prénom du membre.
email	Varchar	20		Ce champ correspond à l'email du membre.
civilité	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 Postale	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		Ce champ correspond au statut du membre. Par défaut il sera à zéro (ce

			DEFAULT 0	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 :

phpMyAdmin

localhost

Bases de données SQL État Variables Jeux de caractères

Actions

MySQL localhost

Créer une base de données ?

site Interclassement Créer

Interclassement pour la connexion MySQL: utf8_general_ci

Interface

Langue - Language : Français - French

Thème / Style: Original

Couleur au choix: Réinitialiser

Taille du texte: 82%

Création d'une nouvelle table :

phpMyAdmin

localhost > site

Structure SQL Rechercher Requête Exporter Importer Opérations Privilèges Supprimer

✓ La base de données site a été créée.

CREATE DATABASE 'site' ;

Aucune table n'a été trouvée dans cette base.

Créer une nouvelle table sur la base site

Nom: membre Nombre de colonnes: 11

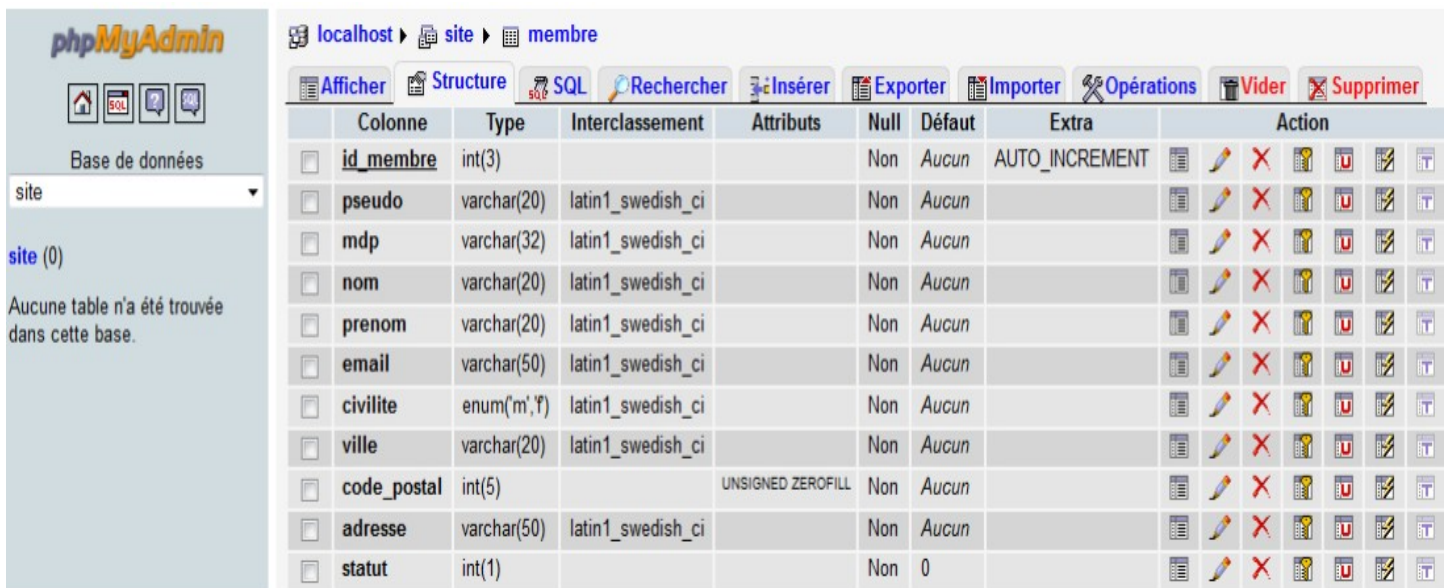
Création de la structure de la table (champs/colonnes) :

phpMyAdmin

localhost > site > membre

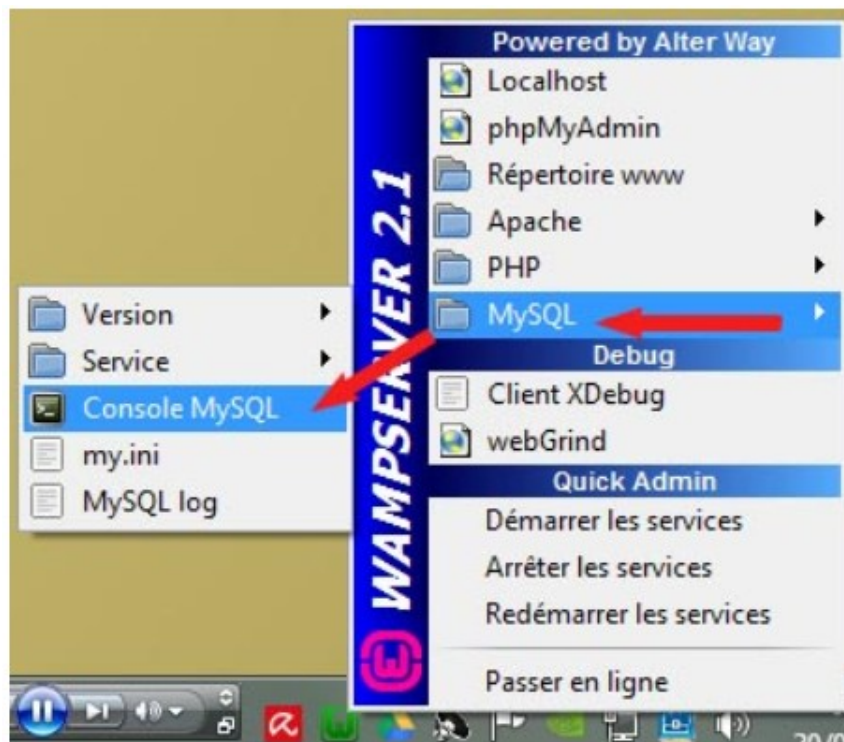
Colonne	Type	Taille/Valeurs	Défaut	Interclassement	Attributs	Null	Index	A.I.	Commentaires
id_membre	INT	3	Aucun			<input type="checkbox"/>	PRIMARY	<input checked="" type="checkbox"/>	
pseudo	VARCHAR	20	Aucun			<input type="checkbox"/>	UNIQUE	<input type="checkbox"/>	
mdp	VARCHAR	32	Aucun			<input type="checkbox"/>	---	<input type="checkbox"/>	
nom	VARCHAR	20	Aucun			<input type="checkbox"/>	---	<input type="checkbox"/>	
prenom	VARCHAR	20	Aucun			<input type="checkbox"/>	---	<input type="checkbox"/>	
email	VARCHAR	50	Aucun			<input type="checkbox"/>	---	<input type="checkbox"/>	
civilite	ENUM	'm','f'	Aucun			<input type="checkbox"/>	---	<input type="checkbox"/>	
ville	VARCHAR	20	Aucun			<input type="checkbox"/>	---	<input type="checkbox"/>	
code_postal	INT	5	Aucun		UNSIGNED ZEROFILL	<input type="checkbox"/>	---	<input type="checkbox"/>	
adresse	VARCHAR	50	Aucun			<input type="checkbox"/>	---	<input type="checkbox"/>	
statut	INT	1	Tel que défini: 0			<input type="checkbox"/>	---	<input type="checkbox"/>	

Structure de la table (relecture) :



Colonne	Type	Interclassement	Attributs	Null	Défaut	Extra	Action
<input type="checkbox"/> id_membre	int(3)			Non	Aucun	AUTO_INCREMENT	
<input type="checkbox"/> pseudo	varchar(20)	latin1_swedish_ci		Non	Aucun		
<input type="checkbox"/> mdp	varchar(32)	latin1_swedish_ci		Non	Aucun		
<input type="checkbox"/> nom	varchar(20)	latin1_swedish_ci		Non	Aucun		
<input type="checkbox"/> prenom	varchar(20)	latin1_swedish_ci		Non	Aucun		
<input type="checkbox"/> email	varchar(50)	latin1_swedish_ci		Non	Aucun		
<input type="checkbox"/> civilite	enum('m','f')	latin1_swedish_ci		Non	Aucun		
<input type="checkbox"/> ville	varchar(20)	latin1_swedish_ci		Non	Aucun		
<input type="checkbox"/> code_postal	int(5)		UNSIGNED ZEROFILL	Non	Aucun		
<input type="checkbox"/> adresse	varchar(50)	latin1_swedish_ci		Non	Aucun		
<input type="checkbox"/> statut	int(1)			Non	0		

Pour créer cette base de données, vous pouvez également passer par la console Mysql:



```
c:\wamp\bin\mysql\mysql5.5.8\bin\mysql.exe
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.5.8-log MySQL Community Server (GPL)

Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql>
```

Voici le code à insérer :

Base de données site - Table membre

```
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;
```

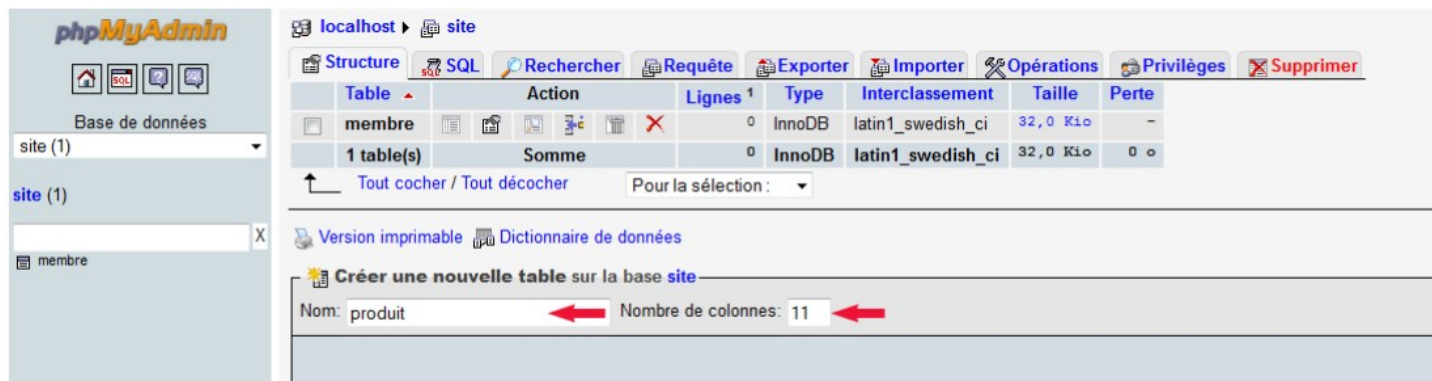
Modélisation et création de la base de données, table et champs. Produit

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
référence	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.
Catégorie	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 déterminer 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) :

Colonne	Type	Taille/Valeurs ¹	Défaut ²	Interclassement	Attributs	Null	Index	A.I.	Commentaire
id_produit	INT	3	Aucun				PRIMARY	<input checked="" type="checkbox"/>	
reference	VARCHAR	20	Aucun				UNIQUE	<input type="checkbox"/>	
categorie	VARCHAR	20	Aucun			<input type="checkbox"/>	---	<input type="checkbox"/>	
titre	VARCHAR	100	Aucun			<input type="checkbox"/>	---	<input type="checkbox"/>	
description	TEXT		Aucun			<input type="checkbox"/>	---	<input type="checkbox"/>	
couleur	VARCHAR	20	Aucun			<input type="checkbox"/>	---	<input type="checkbox"/>	
taille	VARCHAR	5	Aucun			<input type="checkbox"/>	---	<input type="checkbox"/>	
public	ENUM	m',f','mixte'	Aucun			<input type="checkbox"/>	---	<input type="checkbox"/>	
photo	VARCHAR	250	Aucun			<input type="checkbox"/>	---	<input type="checkbox"/>	
prix	INT	3	Aucun			<input type="checkbox"/>	---	<input type="checkbox"/>	
stock	INT	3	Aucun			<input type="checkbox"/>	---	<input type="checkbox"/>	

Structure de la table(relecture) :

	Colonne	Type	Interclassement	Attributs	Null	Défaut	Extra	Action						
<input type="checkbox"/>	id_produit	int(3)			Non	Aucun	AUTO_INCREMENT							
<input type="checkbox"/>	reference	varchar(20)	latin1_swedish_ci		Non	Aucun								
<input type="checkbox"/>	categorie	varchar(20)	latin1_swedish_ci		Non	Aucun								
<input type="checkbox"/>	titre	varchar(100)	latin1_swedish_ci		Non	Aucun								
<input type="checkbox"/>	description	text	latin1_swedish_ci		Non	Aucun								
<input type="checkbox"/>	couleur	varchar(20)	latin1_swedish_ci		Non	Aucun								
<input type="checkbox"/>	taille	varchar(5)	latin1_swedish_ci		Non	Aucun								
<input type="checkbox"/>	public	enum('m','f','mixte')	latin1_swedish_ci		Non	Aucun								
<input type="checkbox"/>	photo	varchar(250)	latin1_swedish_ci		Non	Aucun								
<input type="checkbox"/>	prix	int(3)			Non	Aucun								
<input type="checkbox"/>	stock	int(3)			Non	Aucun								

Pour créer cette base de données, vous pouvez également passer par la console Mysql comme précédemment:

```
c:\wamp\bin\mysql\mysql5.5.8\bin\mysql.exe
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.5.8-log MySQL Community Server (GPL)

Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql>
```

Voici le code à insérer :

```
Base de données site - Table produit

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;
```

Modélisation et création de la base de données, table et champs. Commande				
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.
état	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) :

Colonne	Type	Taille/Valeurs ¹	Défaut ²	Interclassement	Attributs	Null	Index	A.I.	Commentaires
id_commande	INT	3	Aucun			<input checked="" type="checkbox"/>	PRIMARY	<input checked="" type="checkbox"/>	
id_membre	INT	3	NULL			<input checked="" type="checkbox"/>	---	<input type="checkbox"/>	
montant	INT	3	Aucun			<input type="checkbox"/>	---	<input type="checkbox"/>	
date_enregistrement	DATETIME		Aucun			<input type="checkbox"/>	---	<input type="checkbox"/>	
etat	ENUM	'royé','livré'	Aucun			<input type="checkbox"/>	---	<input type="checkbox"/>	

Structure de la table (relecture) :

	Colonne	Type	Interclassement	Attributs	Null	Défaut	Extra	Action
<input type="checkbox"/>	id_commande	int(3)			Non	Aucun	AUTO_INCREMENT	
<input type="checkbox"/>	id_membre	int(3)			Oui	NULL		
<input type="checkbox"/>	montant	int(3)			Non	Aucun		
<input type="checkbox"/>	date_enregistrement	datetime			Non	Aucun		
<input type="checkbox"/>	etat	enum('en cours de traitement','envoyé','livré')	latin1_swedish_ci		Non	Aucun		

Pour créer cette base de données, vous pouvez également passer par la console Mysql:

```

c:\wamp\bin\mysql\mysql5.5.8\bin\mysql.exe
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.5.8-log MySQL Community Server (GPL)

Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>

```

Voici le code à insérer :

Base de données site - Table commande

```
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;
```

Modélisation et création de la base de données, table et champs.
détails_Commande

champ	type	taille	spécificité	Description
id_détails_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 :

 **Créer une nouvelle table** sur la base **site**

Nom: Nombre de colonnes:

Création de la structure de la table (champs/colonnes) :

Colonne	Type	Taille/Valeurs ¹	Défaut ²	Interclassement	Attributs	Null	Index	A 1
id_details_commande	INT	3	Aucun			<input checked="" type="checkbox"/>	PRIMARY	<input checked="" type="checkbox"/>
id_commande	INT	3	NULL			<input checked="" type="checkbox"/>	---	<input type="checkbox"/>
id_produit	INT	3	NULL			<input checked="" type="checkbox"/>	---	<input type="checkbox"/>
quantite	INT	3	Aucun			<input type="checkbox"/>	---	<input type="checkbox"/>
prix	INT	3	Aucun			<input type="checkbox"/>	---	<input type="checkbox"/>

Structure de la table (relecture) :

Pour créer cette base de données, vous pouvez également passer par la console Mysql:

```

c:\wamp\bin\mysql\mysql5.5.8\bin\mysql.exe
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.5.8-log MySQL Community Server (GPL)

Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>

```

Voici le code à insérer :

```
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 pourrions 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 entraînement, 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 e-commerce nous pourrions avoir des dizaines de tables.


Étape 2. Création d'une arborescence


Nous allons créer notre arborescence de site web :


```
- /site/
  ----- /photo/
  ----- /admin/
    ----- gestion_boutique.php
    ----- gestion_commande.php
    ----- gestion_membre.php
  ----- /inc/
    ----- /img/
    ----- /js/
    ----- /css/
    ----- style.php
    ----- init.inc.php
    ----- fonction.inc.php
    ----- haut.inc.php
    ----- bas.inc.php
- inscription.php
- connexion.php
- profil.php
- boutique.php
- fiche_produit.php
- panier.php
- .htaccess
```

Notre site web
se trouvera à








l'intérieur du dossier  /site/.

1 dossier  /photo/ sera présent pour contenir les photos de nos produits.

1 dossier  /admin/ sera présent pour contenir les pages d'administration (BackOffice).

1 dossier  /inc/ sera présent pour contenir les fichiers n'étant pas des pages web à part entière (bien souvent il s'agit de fichiers inclus dans des pages web).

Jusque là, comprenez vous notre arborescence ? elle est plutôt simple, non ? Et bien entendu, il y a aussi les pages web du site web côté FRONT :

-  inscription.php : page d'inscription pour les visiteurs
-  connexion.php : page de connexion pour les membres
-  profil.php : page de profil pour les membres connectés
-  boutique.php : catalogue des différents produits séparés par catégorie
-  fiche_produit.php : fiche d'un produit en particulier
-  panier.php : panier
-  .htaccess : fichier permettant de faire des réglages, notamment sur les URLs pour améliorer le référencement/li>

Étape 3. Écriture des fichiers en inclusion

Étape 3. Écriture 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

```
<?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.

Une 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 chaîne 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 pourrions 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 pourrions modifier cette constante une fois en ligne pour que cela ait une repercussion 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 un 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

<!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.

Étape 3. Écriture du fichier inc/bas.inc.php

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

```
inc/bas.inc.php

        </div>
    </section>
    <footer>
        <div class="conteneur">
            <?php echo date('Y'); ?> - Tous droits réservés - 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

*{ margin: 0; }
a{ text-decoration: none; color: #000; }
.conteneur{ margin: 0 auto; max-width: 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

Étape 3. Écriture 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 devons 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

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 un argument entrant (la requête SQL arrivera dans la variable de réception **\$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.
Message : " . \$mysqli->error . "
Code: " . \$req);** Dans le cas où la requête échoue, on lui demande d'adresser 1 message et d'arrêter l'exécution du code avec l'utilisation de die.

return \$resultat; en cas d'une requête de **SELECT**ion, 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

Étape 4. Écriture 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

<?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

Pseudo**Mot de passe****Nom****Prénom****Email****Sexe**☒ Homme ☐ Femme**Ville****Cp****Adresse**

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

```

<?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>
    <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 chaîne.

\$ désigne la fin de la chaîne.

+ est présent pour dire que les lettres autorisées peuvent apparaître 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 '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 chaîne 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'entraînement).

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 complètement 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. <u>Cliquez ici pour vous connecter</u></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

MONSITE.COM

[Inscription](#)
[Connexion](#)
[Accès à la boutique](#)
[Voir votre panier](#)

Vous êtes inscrit à notre site web. [Cliquez ici pour vous connecter](#)

Pseudo

juju

Mot de passe

•••••

Nom

Cottet

Prénom

Julien

Email

ulien.cottet@gmail.com

Civilité

☒ Homme
 ☐ Femme

Ville

Paris

Code Postal

75015

Adresse

300 rue de vaugirard

S'inscrire

Debug demandé dans le fichier : C:\wamp\www\dev\tutomooc\site\inscription.php à la ligne 5.
 Array
 (
 [pseudo] => juju
 [mdp] => soleil
 [nom] => Cottet
 [prenom] => Julien
 [email] => julien.cottet@gmail.com
 [civilite] => m
 [ville] => Paris
 [code_postal] => 75015
 [adresse] => 300 rue de vaugirard
 [inscription] => S'inscrire
)

Vous devriez retrouver la trace de votre 1er membre dans la base de données :

	id_membre	pseudo	mdp	nom	prenom	email	civilite	ville	code_postal	adresse	statut
<input type="checkbox"/>	1	juju	soleil	Cottet	Julien	julien.cottet@gmail.com	m	Paris	75015	300 rue de vaugirard	0

Maintenant que votre formulaire fonctionne, inscrivez plusieurs membres différents histoire que nous ayons plusieurs enregistrements pour faire des tests plus tard :

id_membre	pseudo	mdp	nom	prenom	email	civilite	ville	code_postal	adresse	statut
1	juju	soleil	Cottet	Julien	julien.cottet@gmail.com	m	Paris	75015	300 rue de vaugirard	0
2	lamarie	planete	thoyer	marie	marie.thoyer@yahoo.fr	f	Lyon	69003	10 rue paul bert	0
3	fab	avatar13	grand	fabrice	fabrice.grand@gmail.com	m	Marseille	13009	70 rue de la république	0
4	membre	membre	membre	membre	membre@exemple.com	f	Toulouse	31000	55 rue bayard	0
5	admin	admin	admin	admin	admin@exemple.com	m	Paris	75015	33 rue mademoiselle	1

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).

Étape 5. Écriture 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

<?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 :

```

<?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 détecter si l'internaute a cliqué sur le bouton submit pour se connecter.

`$resultat = executeRequete("SELECT * FROM membre WHERE pseudo='".$_POST[pseudo]'");` Nous allons utiliser notre fonction `executeRequete` pour aller 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 connaître 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 (puisque'il est maintenant connecté !)

Résultat - connexion.php

MONSITE.COM	Inscription	Connexion	Accès à la boutique	Voir votre panier
-------------	-------------	-----------	---------------------	-------------------

Pseudo

Mot de passe

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.*

Étape 6. Écriture 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 réservé aux membres sur le site web.

Nous allons donc prévoir une page de profil avec quelques informations ! Pour récupérer les informations, nous pourrions 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 accéder a 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 accéder à votre boîte de réception d'e-mail 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

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 réserver sur les autres pages web (plutôt que re-trimbalier un même morceau de 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 réserver sur les autres pages web (plutôt que re-trimbalier un même morceau de 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 :

```

<?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 vérifions `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

MONSITE.COM

Voir votre profil

Accès à la boutique

Voir votre panier

Se déconnecter

Bonjour **Juju**

Voici vos informations de profil

votre email est: julien.cottet@gmail.com
votre ville est: Paris
votre cp est: 75015
votre adresse est: 300 rue de vaugirard

2015 - Tous droits réservés.

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

Étape 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

<!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 :

➤ 1 visiteur aura donc accès à 4 liens

➤ 1 membre aura également 4 liens affichés

➤ 1 administrateur verra 7 liens sur la page web (1 administrateur est aussi 1 membre).

Au passage, pour gérer la déconnexion, nous enverrons dans l'url une information ?action=déconnexion 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 déconnexion sur la page de connexion.

Étape 6. La déconnexion 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 :

```

    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"); ?>

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

```

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'un 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:

- Connectez vous et déconnectez vous ! accédez au site en tant qu'Admin, en tant que Membre, en tant que Visiteur.
- Tenter d'aller sur la page de profil en étant seulement visiteur.
- Vous 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.

Notre espace membre est terminé !

Bien-sur nous pouvons améliorer cet espace membre, le sécuriser, etc. mais pour plus tard...

Étape 7. Le BackOffice - Gestion Boutique - Ajout de produits

Le BackOffice d'un site web est une interface de gestion pour les réglages, réservée seulement à l'administrateur (ou aux administrateurs). Nous avons prévu une partie (dans le dossier /admin/) réservée 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 dit : un administrateur). Imaginez si 1 visiteur ou si 1 membre arrivait à accéder 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 avons prévu un code permettant de renvoyer TRUE (vrai tu es admin) ou FALSE (faux tu n'est pas admin). Nous prévoyons 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

<?php
require_once("../inc/init.inc.php");
//----- TRAITEMENTS PHP -----//
//--- VERIFICATION ADMIN ---//
if(!utilisateurEstConnecteEtEstAdmin())
{
    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");
<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

MONSITE.COM
Gestion des membres
Gestion des commandes
Gestion de la boutique
Voir votre profil
Accès à la boutique
Voir votre panier
Se déconnecter

Formulaire Produits

reference

11-d-23

categorie

tshirt

titre

Tshirt Col V

description

Tee-shirt en coton
flammé liseré
contrastant

couleur

bleu

Taille

M

public

☒ Homme
☐ Femme

photo

Parcourir...

bleu.jpg

prix

20

stock

53

enregistrement

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 decode 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($_POST))`) (autrement dit s'il a été rempli par un clic sur le bouton submit). Nous initialisons une variable `$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 uploadée (`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 uploadée :

- 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'accéder 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).

Étape 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.

Affichage des Articles

Nombre de produit(s) dans la boutique : 8

id_produit	reference	categorie	titre	description	couleur	taille	public	photo	prix	stock	Modification	Supression
1	11-d-23	tshirt	Tshirt Col V	Tee-shirt en coton flammé liseré contrastant	bleu	M	m		20	53		
2	66-f-15	tshirt	Tshirt Col V rouge	c'est vraiment un super tshirt en soirée !	rouge	L	m		15	230		
3	88-g-77	tshirt	Tshirt Col rond vert	Il vous faut ce tshirt Made In France !!!	vert	L	m		29	63		
4	55-b-38	tshirt	Tshirt jaune	le jaune reviens à la mode, non? :-)	jaune	S	m		20	3		
5	31-p-33	tshirt	Tshirt noir original	voici un tshirt noir très original :p	noir	XL	m		25	80		
6	56-a-65	chemise	Chemise Blanche	Les chemises c'est bien mieux que les tshirts	blanc	L	m		49	73		
7	63-s-63	chemise	Chemise Noir	Comme vous pouvez le voir c'est une chemise noir...	noir	M	m		59	120		
8	77-p-79	pull	Pull gris	Pull gris pour l'hiver	gris	XL	m		79	99		

admin/gestion_boutique.php

```
<?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>Suppression</th>';
    $contenu .= '</tr>';

    while ($ligne = $resultat->fetch_assoc())
    {
        $contenu .= '<tr>';
        foreach ($ligne as $indice => $information)
        {
            if($indice == "photo")
            {
                $contenu .= '<td></td>';
            }
            else
            {
                $contenu .= '<td>' . $information . '</td>';
            }
        }
        $contenu .= '<td><a href="?action=modification&id_produit=' . $ligne['id_produit'] . '">
</a></td>';
        $contenu .= '<td><a href="?action=suppression&id_produit=' . $ligne['id_produit'] . '"
OnClick="return(confirm(\'En êtes vous certain ?\'));"></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 détecter l'action à déclencher. L'administrateur peut donc choisir l'action qu'il souhaite mettre en œuvre : 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](#).

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.

Étape 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 :


```

<?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>Suppression</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>';
            }
        }
    }
}

```

```

    }
    $contenu .= '<td><a href="?action=modification&id_produit=' . $ligne['id_produit'] . '">
</a></td>';
    $contenu .= '<td><a href="?action=suppression&id_produit=' . $ligne['id_produit'] . '"
OnClick="return(confirm(\'En êtes vous certain ?\'))";></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 détectons l'action suppression dans l'url
 if(isset(\$_GET['action']) && \$_GET['action'] ==
 "suppression").

Nous allons d'abord sélectionner 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.

Étape 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

```
<?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 ---//
```

```

//--- 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($valeur));
    }
    executeRequete("REPLACE INTO produit (id_produit, reference, categorie, titre, description,
couleur, taille, public, prix, stock) values ('$_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>';
$contenu .= '<a href="?action=ajout">Ajout d\'un produit</a><br><br><br>';
//--- AFFICHAGE PRODUITS ---//
if(isset($_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())
    {
        $contenu .= '<tr>';
        foreach ($ligne as $indice => $information)
        {
            if($indice == "photo")
            {
                $contenu .= '<td></td>';
            }
            else
            {
                $contenu .= '<td>' . $information . '</td>';
            }
        }
        $contenu .= '<td><a href="?action=modification&id_produit=' . $ligne['id_produit'] . '">
</a></td>';
        $contenu .= '<td><a href="?action=suppression&id_produit=' . $ligne['id_produit'] . '"
OnClick="return(confirm(\'En êtes vous certain ?\'));"></a></td>';
        $contenu .= '</tr>';
    }
    $contenu .= '</table><br><br><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

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 uploadée (si elle a été uploadée) pour écraser la photo actuelle.

Nous faisons évoluer la requête SQL afin que celle-ci devienne REPLACE.

➤ REPLACE se comporte en INSERT (ajout) s'il n'y a pas d'id_produit connu

➤ REPLACE 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:

MONSITE.COM Gestion des membres Gestion des commandes **Gestion de la boutique** Voir votre profil Accès à la boutique Voir votre panier Se déconnecter

Affichage des produits
Ajout d'un produit

Formulaire Produits

reference
11-d-23

categorie
tshirt

titre
Tshirt Col V

description
Tee-shirt en coton
flamé liseré
contrastant


couleur
bleu

Taille
M

public
☒ Homme ☐ Femme

photo
 Aucun fichier sélectionné.

Vous pouvez uploader une nouvelle photo si vous souhaitez la changer



Vous pouvez uploader une nouvelle photo si vous souhaitez la changer



prix
20

stock
53

Modification du produit

Étape 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 réellement notre boutique, nous allons travailler dans le fichier : boutique.php

boutique.php

```
<?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 sélectionner 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?catégorie=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


MONSITE.COM [Gestion des membres](#) [Gestion des commandes](#) [Gestion de la boutique](#) [Voir votre profil](#) [Accès à la boutique](#) [Voir votre panier](#) [Se déconnecter](#)

tshirt

chemise


pull

Tshirt Col V




20 €
[Voir la fiche](#)

Tshirt Col V rouge




15 €
[Voir la fiche](#)

Tshirt Col rond vert




29 €
[Voir la fiche](#)

Tshirt jaune



20 €
[Voir la fiche](#)

Tshirt noir original



25 €
[Voir la fiche](#)

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

```

*{ margin: 0; }
a{ text-decoration: none; color: #000; }
.conteneur{ margin: 0 auto; max-width: 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; width: 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; width: 60%; }
.boutique-produit{ float: left; width: 30%; text-align: center; padding: 5px; 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 !

Étape 9. Boutique - La fiche produit

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

```

<?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><br><br>";
$contenu .= "<p>Categorie: $produit[categorie]</p>";
$contenu .= "<p>Couleur: $produit[couleur]</p>";
$contenu .= "<p>Taille: $produit[taille]</p>";
$contenu .= "<img src='$produit[photo]' width='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 !';
}

```

```

$contentu .= "<br><a href='boutique.php?categorie=" . $produit['categorie'] . "'>Retour vers la
sélection de " . $produit['categorie'] . "</a>";
//----- AFFICHAGE HTML -----//
require_once("inc/haut.inc.php");
echo $contentu;
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 \$contentu, 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 sélection 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 interroge 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 catégorie du produit courant.

Résultat

MONSITE.COM Gestion des membres Gestion des commandes Gestion de la boutique Voir votre profil Accès à la boutique Voir votre panier Se déconnecter

Titre : Tshirt Col V rouge

Categorie: tshirt
Couleur: rouge
Taille: L



Description: c'est vraiment un super tshirt en soirée !

Prix : 15 €

Nombre produit(s) disponible : 230

Quantité : 1

[Retour vers la sélection de tshirt](#)

Étape 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 schéma 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

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

Création 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 où 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 désirée 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 rentre 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 dernière  
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
<?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'][$id_produit])) // panier vide
{
    echo "<tr><td colspan='5'>Votre panier est vide</td></tr>";
}
else
{
    echo "<tr><td colspan='5'>Votre panier contient des produits</td></tr>";
}
echo "</table><br>";
echo "<i>Règlement par CHÈQUE uniquement à l'adresse suivante : 300 rue de vaugirard 75015 PARIS</i>";
echo "<br>";
echo "<hr>session panier:<br>"; debug($_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 entraîne 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 quantité 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

MONSITE.COM	Voir votre profil	Accès à la boutique	Voir votre panier	Se déconnecter
-------------	-------------------	---------------------	-------------------	----------------

Panier				
Titre	produit	Quantité	Prix Unitaire	Action
Votre panier contient des produits				

Règlement par CHÈQUE uniquement à l'adresse suivante : 300 rue de vaugirard 75015 PARIS

Étape 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

<?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>Règlement par CHÈQUE uniquement à l'adresse suivante : 300 rue de vaugirard 75015 PARIS</i>";
// echo "<hr>session panier:<br>"; debug($_SESSION);
include("inc/bas.inc.php");
?>
```

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 à 10h07 pour en prendre 2 dans son panier.

Olivier arrive à 10h08 pour en prendre 1 dans son panier.

À 10h10, Olivier valide son panier et son achat du tshirt rouge.

À 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 carrément le produit. On ne va pas remplacer son 6 par un 0

Résultat

MONSITE.COM

[Voir votre profil](#) [Accès à la boutique](#) [Voir votre panier](#) [Se déconnecter](#)

Merci pour votre commande. votre n° de suivi est le 1

Panier				
Titre	Produit	Quantité	Prix Unitaire	Action
Votre panier est vide				

Règlement par CHÈQUE uniquement à l'adresse suivante : 300 rue de vaugirard 75015 PARIS

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'hésitez 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.

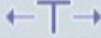









Nous pourrions réalisé la décrémentation du stock mais le paiement se déroule par cheque alors nous attendrons que l'administrateur reçoive ce cheque et décrémente lui même son stock via le BackOffice. Une fois que l'internaute à déclaré le paiement, nous vidons son panier car tous les produits sont 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

	id_commande	id_membre	montant	date_enregistrement	etat
  	1	4	301	2015-07-10 14:44:46	en cours de traitement

	id_details_commande	id_commande	id_produit	quantite	prix
  	1	1	2	1	15
  	2	1	6	1	49
  	3	1	8	3	79

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. Ça prend du temps de créer 1 site web complet ! Entraînement - Développer davantage votre site et vos connaissances pour d'avantage développer notre site, je vous propose plusieurs suggestions d'exercices 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 e-mail de notification pour l'informer de l'évolution de sa commande. L'affichage du chiffre d'affaires doit également apparaître 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 / Amélioration 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 e-mail.

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.

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 système 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. Prévoyez 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.