

L i a m T A R D I E U

www.evogue.fr



➤ Sommaire

➤	Sommaire	2
➤	Éléments initiaux	3
➤	Base de données	3
➤	Tables.....	4
	membres.....	4
	commandes.....	4
	articles	4
	details_commandes.....	4
➤	L'arborescence.....	5
	Le dossier admin.....	5
	Le dossier photo	6
	Le dossier inc.....	6
➤	Les fichiers inclus	7
	connexion_bdd.inc.php	7
	fonctions.inc.php.....	7
	menu.inc.php	7
➤	Les pages	8
	inscription.php	8
	connexion.php.....	8
	profil.php	8
	boutique.php	9
	fiche_article.php.....	9
	panier.php.....	9
➤	Les pages d'admin.....	11
	gestion_membres.php	11
	gestion_articles.php	11
	gestion_commandes.php.....	11



Éléments initiaux

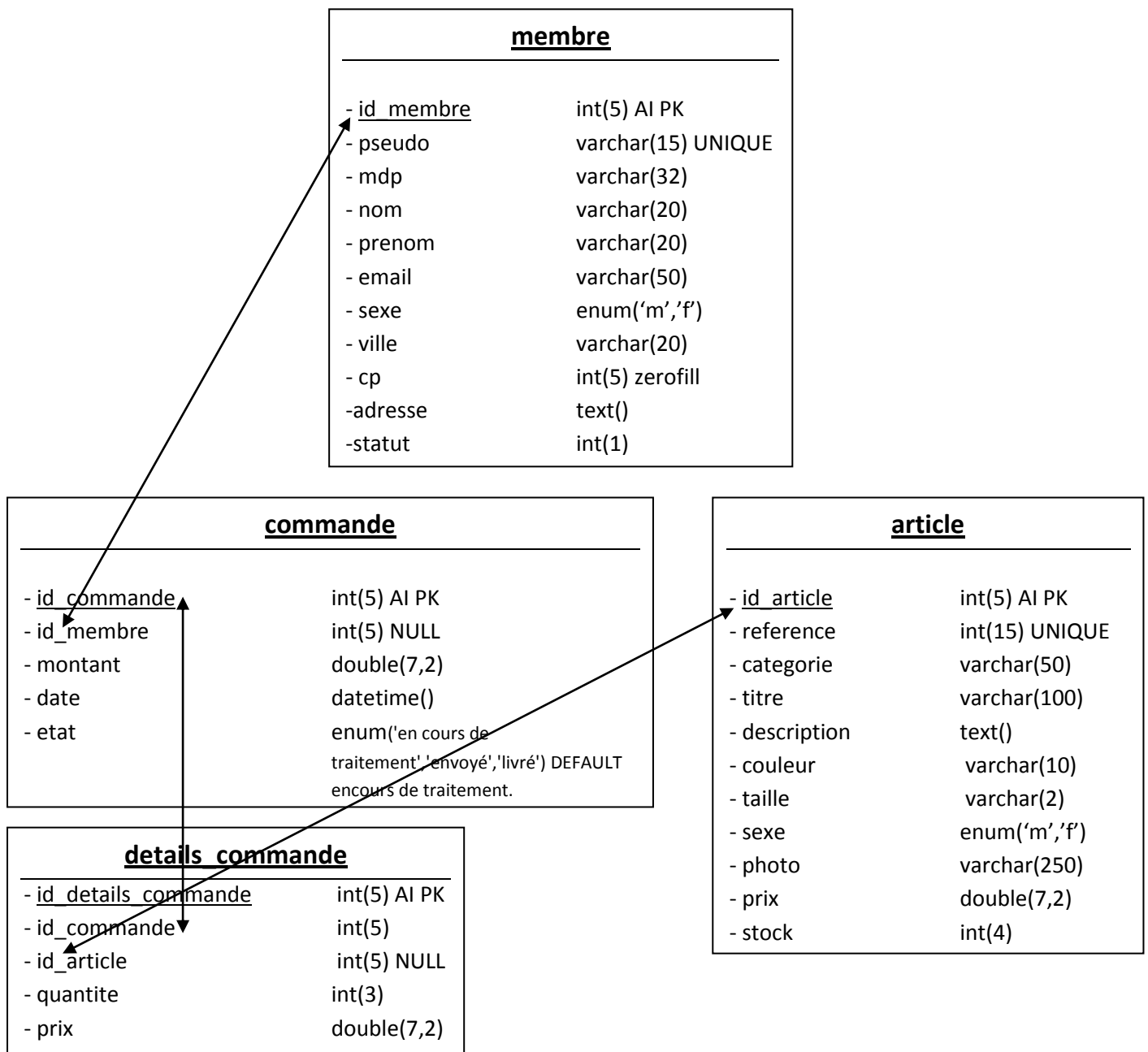
Dans cette partie, nous allons réaliser un site de ventes en ligne (e-commerce).

Nous nous plaçons comme les développeurs qui doivent créer un site Internet pour une boutique qui est déjà implantée et qui souhaite orienter son activité sur Internet afin de mettre son image au goût du jour et de maximiser le nombre de ses ventes.



Base de données

La base de données se nomme : tic_site





Tables

membre

```
CREATE TABLE membre (  
  id_membre int(5) NOT NULL auto_increment,  
  pseudo varchar(15) NOT NULL,  
  mdp varchar(32) NOT NULL,  
  nom varchar(20) NOT NULL,  
  prenom varchar(20) NOT NULL,  
  email varchar(20) NOT NULL,  
  sexe enum('m','f') NOT NULL,  
  ville varchar(20) NOT NULL,  
  cp int(5) NOT NULL,  
  adresse text NOT NULL,  
  statut int(1) NOT NULL,  
  PRIMARY KEY (id_membre),  
  UNIQUE KEY pseudo (pseudo)) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

commande

```
CREATE TABLE commande (  
  id_commande int(6) NOT NULL auto_increment,  
  id_membre int(5) default NULL,  
  montant double(7,2) NOT NULL,  
  date datetime NOT NULL,  
  etat enum('en cours de traitement','envoyé','livré') NOT NULL default 'en cours de  
traitement',  
  PRIMARY KEY (id_commande),  
  KEY id_membres (id_membre)) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

article

```
CREATE TABLE article (  
  id_article int(5) NOT NULL auto_increment,  
  reference int(15) NOT NULL,  
  categorie varchar(70) NOT NULL,  
  titre varchar(150) NOT NULL,  
  description text NOT NULL,  
  couleur varchar(10) NOT NULL,  
  taille varchar(2) NOT NULL,  
  sexe enum('m','f') NOT NULL,  
  photo varchar(250) NOT NULL,  
  prix double(7,2) NOT NULL,  
  stock int(4) NOT NULL,  
  PRIMARY KEY (id_article)) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

details_commande

```
CREATE TABLE details_commande (  
  id_details_commande int(5) NOT NULL auto_increment,  
  id_commande int(6) NOT NULL,  
  id_article int(5) default NULL,  
  quantite int(4) NOT NULL,  
  prix double(7,2) NOT NULL,  
  PRIMARY KEY (id_details_commande),  
  KEY id_articles (id_article)) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Nous gérerons également les relations ainsi que les contraintes d'intégrités.

➤ L'arborescence

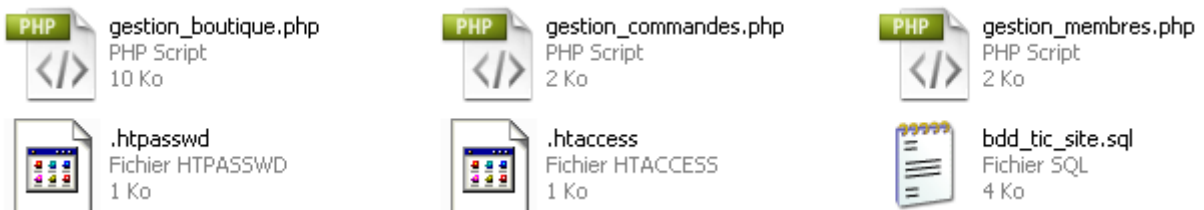
Nous créons un dossier « tic_site » sous wamp.



A l'intérieur de celui-ci, nous créons un dossier « admin », un dossier « photo », puis un dossier « inc ».

Le dossier admin

Ce dossier va contenir les pages du backoffice (il sera d'ailleurs de bon augure de mettre un fichier .htaccess et un fichier .htpasswd, *ces notions ont été abordées dans le support précédent*).



En informatique, le back office est une partie d'un site Web ou d'un système informatique. Pour un site e-commerce, on considérera qu'il s'agit de l'arrière-boutique.

Elle concerne la partie qui permet à l'entreprise d'administrer et de gérer son site. Nous pourrions retrouver comme service par exemple :

- ajout des produits et des services ;
- modifications de paramètres ;
- administration d'un forum ;
- gestion des utilisateurs ;
- ...etc.

Le dossier photo

Ce dossier contiendra les images de nos produits (mis en vente sur le site)



bottebeigef.jpg



bottenoirf.jpg



chemiseblanchem.jpg



chemisenoirm.jpg



pullgrism2.jpg



tshirtbleum.jpg



tshirtvertm.jpg



tshirtvioletm.jpg

Le dossier inc

Ce dossier contiendra tous les fichiers inclus tels que les fichiers de fonctions, feuille de style, etc.



connexion_bdd.inc.php

PHP Script
1 Ko



fonction.inc.php

PHP Script
14 Ko



haut.inc.php

PHP Script
1 Ko



init.inc.php

PHP Script
1 Ko



menu.inc.php

PHP Script
3 Ko



style.css

Document de feuille de style e...
2 Ko



Les fichiers inclus

Les fichiers contenus dans le dossier « inc » (abréviation de include)

connexion_bdd.inc.php

voir le code de connexion_bdd.inc.php

Comme son nom l'indique, ce fichier sert à créer une connexion avec la base de données. Nous allons l'inclure dans la page **fonctions.inc.php** en première position, car les fonctions que l'on utilise contiennent des requêtes et elles ont besoin de savoir sur quelle base de données elles doivent s'exécuter. Nous en profitons donc pour lui ajouter « session_start() » de manière à ne pas répéter cette ligne dans chaque page du site (*là aussi, les sessions ont été abordées dans le support de cours précédents*).

fonctions.inc.php

voir le code de fonctions.inc.php

Cette page répertorie toutes les fonctions dont nous avons besoin pour les fonctionnalités du site. Un peu à la manière des feuilles de style css qui permettent de mettre en forme un site, ces fonctions sont centralisées dans le même fichier afin de ne pas avoir à les chercher partout. C'est une organisation fréquemment préconisée. Nous allons l'inclure dans toutes nos pages.

L'ajout d'articles au panier est géré en session, nous aurions pu le faire avec une table dans la bdd mais cette pratique est de moins en moins utilisée car elle a tendance à surcharger la base avec les clients qui ajoutent des articles au panier mais qui ne le valident jamais pour acheter.

La gestion d'un panier par des sessions est une pratique courante et actuelle.

De cette manière, seulement le paiement final du panier par le client entrainera l'exécution de requêtes vers la bdd afin de mettre à jour le stock restant de vêtements.

Nous incluons également dans ce dossier « inc » la feuille de style css.

menu.inc.php

voir le code de menu.inc.php

Plus tard, nous créerons un fichier qui permettra de recenser toutes les pages de notre site sans que nous ayons à créer les liens à la main entres elles et à chaque fois.

Pour une meilleure navigation, ce fichier « **menu.inc.php** » sera inclus dans toutes les pages de notre site sans exception. Cela nous permettra de gagner du temps et de ne pas répéter des lignes indispensables.

Grâce à une fonction, nous attribueront une couleur différente en fonction du rôle de l'utilisateur connecté : admin – membre – visiteur. Nous n'appellerons jamais ce fichier dans le navigateur, il n'est destiné qu'à l'inclusion. C'est la raison pour laquelle nous avons précédé son extension de « .inc » (ceci n'est pas obligatoire, juste une convention de nommage).



Les pages

inscription.php

voir le code de `inscription.php`

Cette page permet d'inscrire des membres dans la base de données.

Elle affiche un formulaire avec les informations requises : pseudo, mdp, nom, prenom, email, sexe, ville, cp, adresse. Les valeurs sont récupérées puis transmises à des variables qui passent en paramètre de la fonction « `inscription_membres` ».

Nous vérifions que le pseudo fait bien au minimum 2 caractères. Il est important de vérifier si le pseudo est déjà pris ou s'il est bien disponible, cette possibilité n'est pas configurée de base pour être rejetée, c'est donc à nous de faire cette vérification qui s'impose.

Il est également possible d'ajouter des tests de sécurité pour éviter de mauvaise surprise comme réaliser un cryptage avant d'enregistrer le mot de passe en bdd, vérifier de manière automatisée si l'utilisateur est humain ou un ordinateur. Se protéger contre les injections sql (voir le chapitre consacré), et autres...

connexion.php

voir le code de `connexion.php`

Cette page affiche un formulaire avec deux cases, l'une pour saisir le pseudo, l'autre pour renseigner le mot de passe associé au compte de l'utilisateur.

Nous traiterons la fonction « `controle_membres` » pour savoir si l'utilisateur existe et si c'est le cas nous comparerons le mot de passe du membre récupéré de la base avec le mot de passe qui a été saisi dans le formulaire par l'utilisateur qui tente de se connecter.

Si c'est le cas nous chargeons ses données, nous les transmettons dans une variable de session et nous le redirigeons vers son compte (***profil.php***).

profil.php

voir le code de `profil.php`

Nous vérifions si l'utilisateur qui tente d'accéder à cette page s'est bien identifié avant grâce à la page ***connexion.php***. Effectivement, nous pouvons penser que quelqu'un puisse saisir directement l'adresse menant à ***profil.php*** directement dans l'url de son navigateur ou y accéder via un « favoris ». Cette page informe le membre sur ses informations et ses actions possibles, il est donc OBLIGATOIRE qu'il se connecte avant via la page ***connexion.php***.

Si l'utilisateur connecté possède le rôle d'admin, nous lui attribuons des droits supplémentaires tels que :

- Gestion des membres
- Gestion de la boutique
- Gestion des commandes

Ces liens mènent vers son backoffice (interface de gestion) car il est administrateur.

Sinon, un utilisateur « normal » (ou un utilisateur « admin », en plus du backoffice), aura accès aux actions suivantes :

- Voir votre panier
- Modifier vos informations
- Se déconnecter

boutique.php

voir le code de `boutique.php`

Voici une page qui affiche toutes les catégories de vêtements disponibles à la vente.

En cliquant sur une catégorie tels que « jean » ou « T-shirt » nous affichons les produits disponibles et le prix de vente associé (un lien permet de consulter la fiche détaillée de chaque produit, il s'agit de **fiche_article.php**).

Une zone de recherche est également présente afin de sélectionner les produits inférieurs au prix de vente maximum de 30€, 70€, ou autre montant...

fiche_article.php

voir le code de `fiche_article.php`

Cette page affiche toutes les informations d'un produit en fonction d'un paramètre reçu dans l'url en GET (si nous ne prenons pas de précautions, cette méthode n'est pas la plus sécurisée qu'il soit, voir le chapitre consacré sur la sécurité et les injections sql dans le support précédent). Si l'internaute est intéressé par le produit, il est invité à se connecter ou créer un compte s'il n'en possède pas afin de pouvoir l'ajouter au panier. A l'inverse si l'utilisateur est connecté, il peut ajouter le produit au panier en choisissant la quantité (la quantité d'un produit est renseignée grâce aux champs stock de la table articles). Après son ajout, Il sera directement redirigé vers la page **panier.php**.

panier.php

voir le code de `panier.php`

Cette page permet de visualiser les produits que l'on a ajoutés au panier. Trois possibilités s'offrent à l'utilisateur :

- Effacer son panier
- Continuer ses achats
- Payer les produits dans le panier

Si l'utilisateur choisit d'effacer son panier, c'est la session de son panier qui est détruite, son panier se retrouve vide.

S'il souhaite continuer ses achats, il peut ajouter autant d'articles qu'il le souhaite (dans la limite des stocks disponibles tout de même). Le prix unitaire d'un article est indiqué, le montant global des articles est calculé et visible par l'utilisateur. Le calcul de la TVA est pris en compte dans le montant.

Une fois que l'utilisateur a terminé ses achats, il peut naturellement payer ses articles dans le panier afin de les commander. Cette action déclenche une entrée dans la table « commande » et contient :

- Un numéro de suivi de commande.
- La référence de l'article (afin de savoir ce quel vêtement a été commandé).
- L'id du membre (qui nous permettra de récupérer son adresse dans la table membres pour savoir à qui nous devons envoyer le colis).
- La quantité d'articles achetés en ligne.
- Le montant total exprimé en Euros.
- La date d'achat.
- L'état (exemple : en cours de traitement).

Une fois ces opérations réalisées, il ne faut pas oublier de transmettre la référence de l'article commandé ainsi que la quantité choisie par l'acheteur à une fonction qui aura pour but de changer le stock disponible pour l'article en question dans la table « articles ».

Le panier du client se retrouve alors naturellement à nouveau vide.

Son numéro de suivi de commande lui est communiqué.



Les pages d'admin

Nous sommes dans notre backoffice (*définition en page 5*).

Ces pages ne sont accessibles que si l'on est enregistré en tant qu'administrateur dans la bdd.

Protection assurée à la fois par le htaccess et à la fois dans le code des pages.

gestion_membres.php

voir le code de `gestion_membre.php`

Cette page permet de gérer les utilisateurs inscrits sur notre site.

Nous pouvons à la fois visualiser leurs coordonnées, mais aussi les supprimer du site (s'ils ne respectent pas les règles d'utilisation par exemple).

gestion_articles.php

voir le code de `gestion_articles.php`

Cette page permet d'ajouter des articles dans notre boutique.

Cela permet d'avoir un affichage d'articles dynamiques dans **boutique.php** géré par le backoffice et non pas d'avoir un site statique où le contenu reste figé et, où il faudrait replonger dans le code pour créer de nouvelles pages d'articles. Le backoffice permet au client final d'être autonome.

Nous pouvons également modifier ou supprimer des articles (qui seraient en rupture de stock, ou en fin de série par exemple).

Les photos des articles associés sont stockées dans le dossier « **photo/** » sur notre serveur.

gestion_commandes.php

voir le code de `gestion_commandes.php`

Cette page liste les commandes réalisées sur le site (c'est-à-dire : les articles ajoutés au panier et payés par les clients/utilisateurs).

Elle indique aussi le Chiffre d'Affaires de la société (CA) en Euros.