# Introduction

Il faut moderniser son site web une problématique pour chaque entreprise qui possède un site web qui a fait son temps, de nouvelle norme et technique apparaisse rendant les sites plus fluide et esthétique. Les sites E-Commerce sont fleurissant de nos jours et banale il est donc important de réussir à se démarquer des autres par des fonctionnalité et une esthétique propre. Ces sites sont une fenêtre qui mettent en contact une entreprise avec ses clients afin de partager les nouveaux produits, promotions et nouvelles modalités.

Nous avons donc eu pour projet, de développer un site E-Commerce pour la société P.M.U qui en posséder déjà un mais vieillissant. Ainsi l'objectif est de développer un site web moderne avec une charte graphique et toutes les fonctionnalités que l’on attend d’un site E-Commerce en 2020.

Nous avons suivi les caractéristiques des sites de commerces en ligne telle que Amazon, comme une barre de recherche, un espace client etc… un espace de gestion pour les produits, conseils et commande du site.

Pour réussir ce projet, nous avons tout d'abord listé les tâches à faire et nous nous les sommes répartis, afin d’être efficace compte tenu du temps qui nous était impartie. Ce qui nous a permis de développer plus efficacement les différentes fonctionnalités du site. Enfin, nous avons procédé à un rapport sur notre projet.

# Description du projet

P.M.U souhaite donc créer un site web afin de pouvoir vendre ses produits de construction.

Ce site permettra à l’entreprise de prendre des commandes, gérer ses clients et ses produits.

Ce site possédera donc un portail de connexion, un affichage de promotion, un espace utilisateur et administrateur, un panier d’achat, un espace de vente par catégorie, une barre de recherche produit etc…

Un client pourra donc prendre des articles dans son panier et commander, et accéder a ses informations dans son espace réservé.

Un administrateur pourra gérer l’ensemble du site, que ce soit les produits, les conseils, les autorisations etc.… depuis un interface spécial accessible depuis le site.

# Exigence non fonctionnelle

Exigences de qualité :

Afin que le site donne envie aux clients de l’utiliser et l’explorer, il est important de rendre le site dynamique.

Ergonomie efficace :

Le client doit utiliser le site de manière naturel sans trop chercher ce qu’il veut faire.

Interface graphique :

Les différentes couleurs et choix typographiques doivent permettre à un client de repérer les différentes fonctionnalités qui s’offrent à lui que ce soit dans le panier, dans les catégories de produit et dans son espace réservé.

# Conception technique :

Le site web est basé sur une architecture MVC.

Le pattern MVC permet de bien organiser son code source. Il aide à savoir quels fichiers créer, mais surtout à définir leur rôle. Le but de MVC est justement de séparer la logique du code en trois parties que l'on retrouve dans des fichiers distincts.

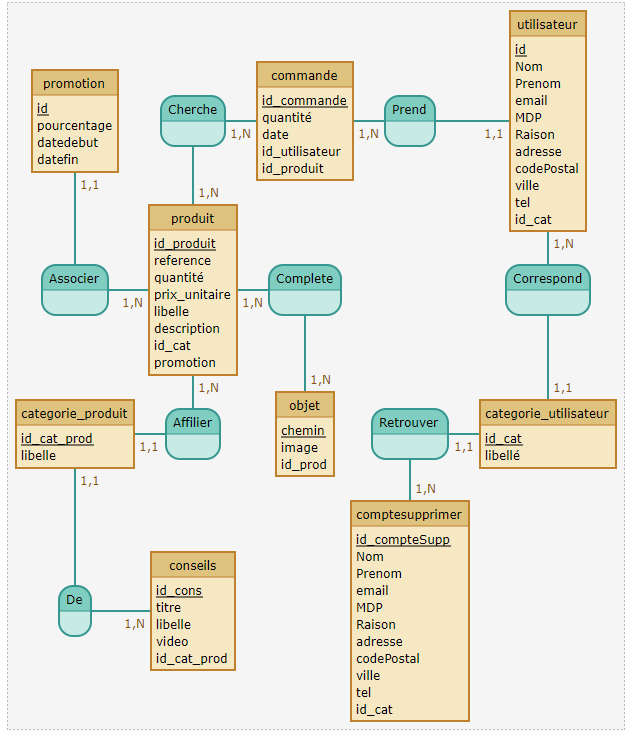
Une application conforme au modèle MVC comporte trois types de modules : les modèles, les vues et les contrôleurs :

**Modèle :** Élément qui contient les données ainsi que de la logique en rapport avec les données : validation, lecture et enregistrement. Il peut, dans sa forme la plus simple, contenir uniquement une simple valeur, ou une structure de données plus complexe. Le modèle représente l'univers dans lequel s'inscrit l'application. Par exemple pour une application de banque, le modèle représente des comptes, des clients, ainsi que les opérations telles que dépôt et retraits, et vérifie que les retraits ne dépassent pas la limite de crédit.

**Vue :** Partie visible d'une [interface graphique](https://fr.wikipedia.org/wiki/Interface_graphique). La vue se sert du modèle, et peut être un diagramme, un formulaire, des boutons, etc. Une vue contient des éléments visuels ainsi que la logique nécessaire pour afficher les données provenant du modèle. Dans une application de bureau classique, la vue obtient les données nécessaires à la présentation du modèle en posant des questions. Elle peut également mettre à jour le modèle en envoyant des messages appropriés. Dans une application web une vue contient des balises [HTML](https://fr.wikipedia.org/wiki/Hypertext_Markup_Language).

**Contrôleur :** Module qui traite les actions de l'utilisateur, modifie les données du modèle et de la vue.

# Chapitre BDD



**promotion** ( id, pourcentage, datedebut, datefin, *id\_produit* )

**Cherche** ( *id\_produit*, *id\_commande* )

**commande** ( id\_commande, quantité, date, id\_utilisateur, id\_produit )

**utilisateur** ( id, Nom, Prenom, email, MDP, Raison, adresse, codePostal, ville, tel, id\_cat, *id\_commande* )

**produit** ( id\_produit, reference, quantité, prix\_unitaire, libelle, description, id\_cat, promotion )

**Complete** ( *id\_produit*, *chemin* )

**categorie\_produit** ( id\_cat\_prod, libelle, *id\_produit*, *id\_cons* )

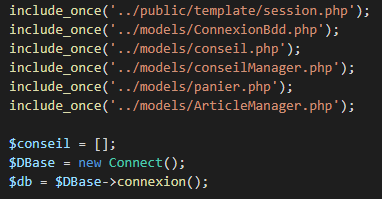
**objet** ( chemin, image, id\_prod )

**categorie\_utilisateur** ( id\_cat, libellé, *id*, *id\_compteSupp* )

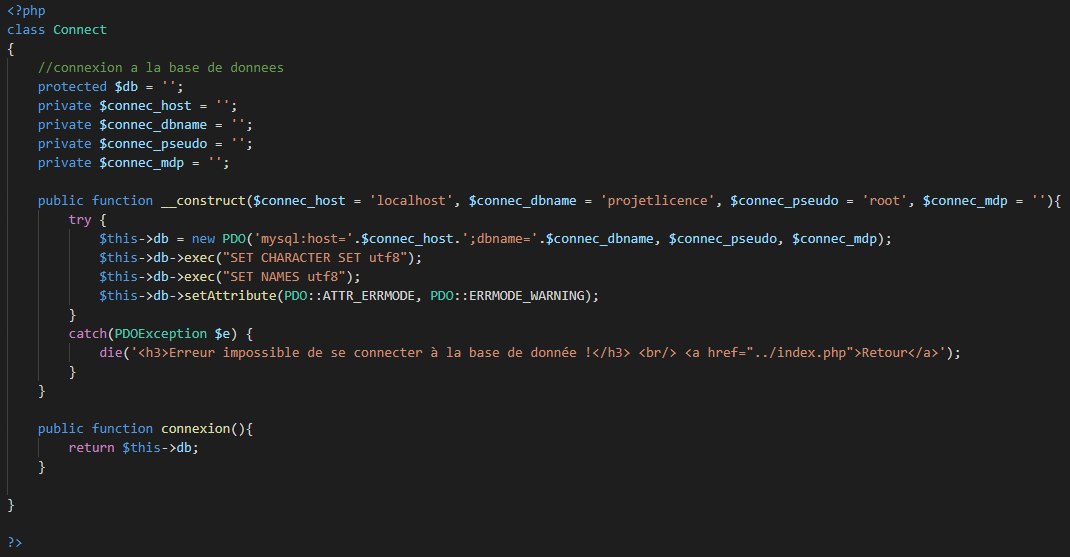
**conseils** ( id\_cons, titre, libelle, video, id\_cat\_prod )

**comptesupprimer** ( id\_compteSupp, Nom, Prenom, email, MDP, Raison, adresse, codePostal, ville, tel, id\_cat )

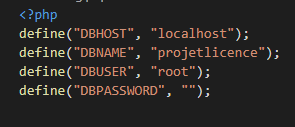
Nous faisons appel à la base de données de deux manières sur le site web :

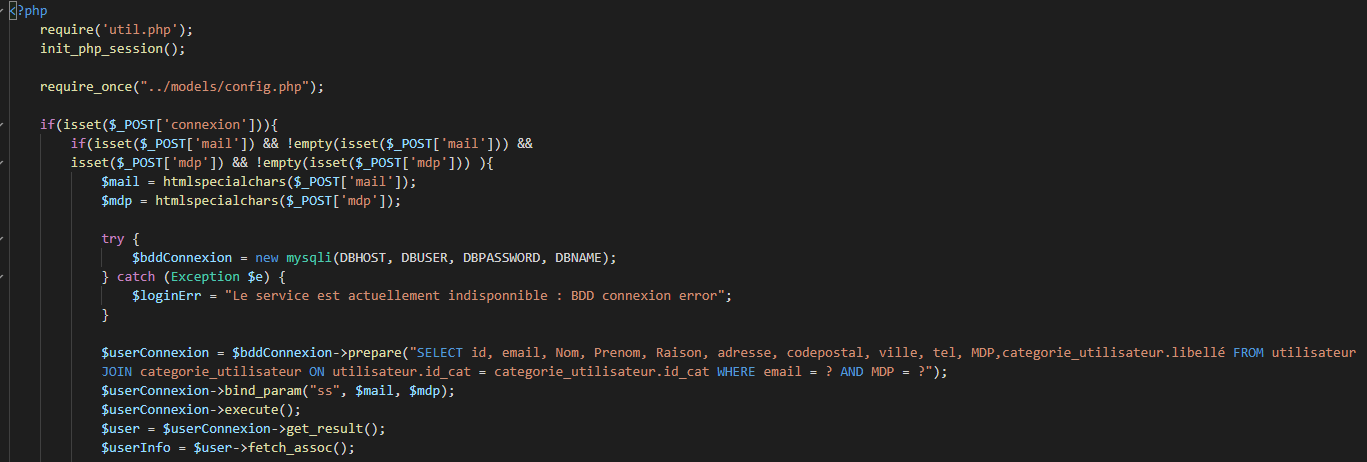


L’objet Connect :



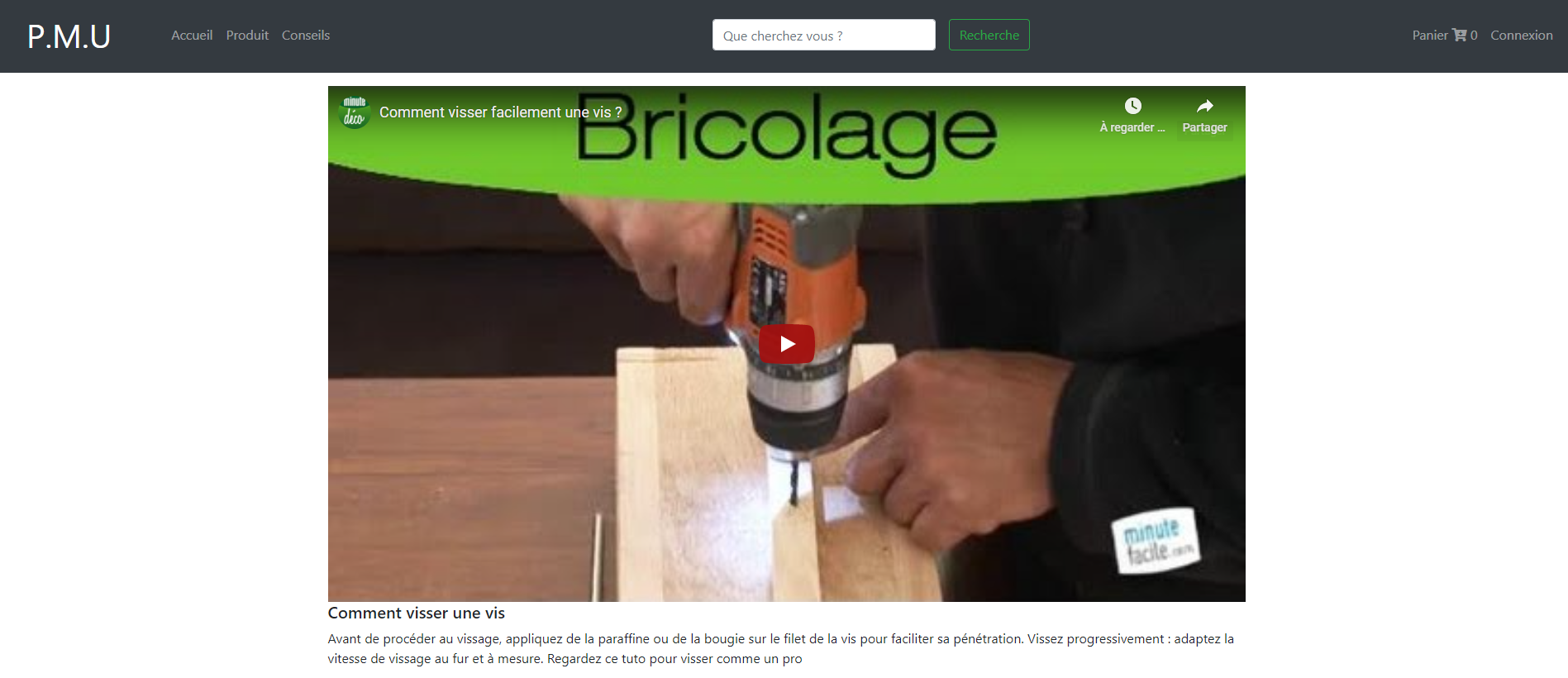
Deuxième méthode :





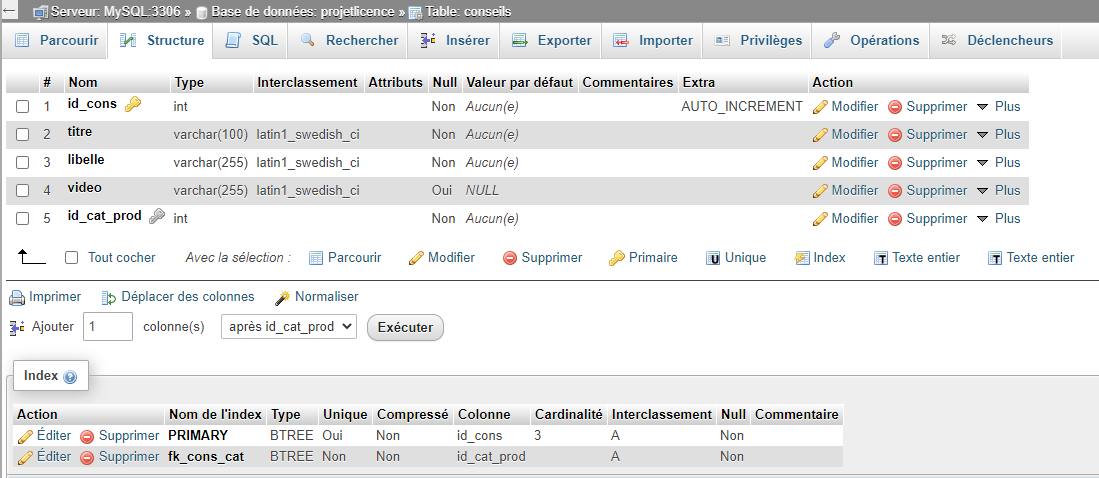
# Chapitre Conseil

Voici ce qui apparait lorsque l’on clique sur la rubrique conseil.

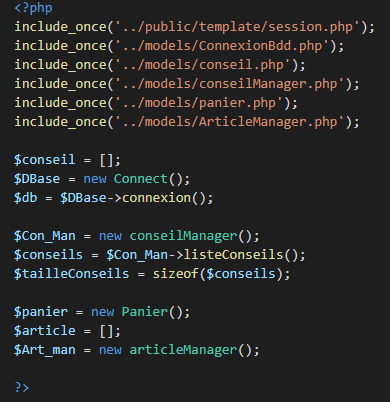


Une vidéo explicative tirée d’un site de vidéo, ainsi qu’un titre et un paragraphe contenant des conseils si les images ne suffisent pas.

Les données d’une vidéo de conseil sont stockées dans la base de données, dans la table conseils et possède la structure suivante : (Pour des infos complémentaires consulter le MCD ou la bibliothèque de base de données fournies en annexe)



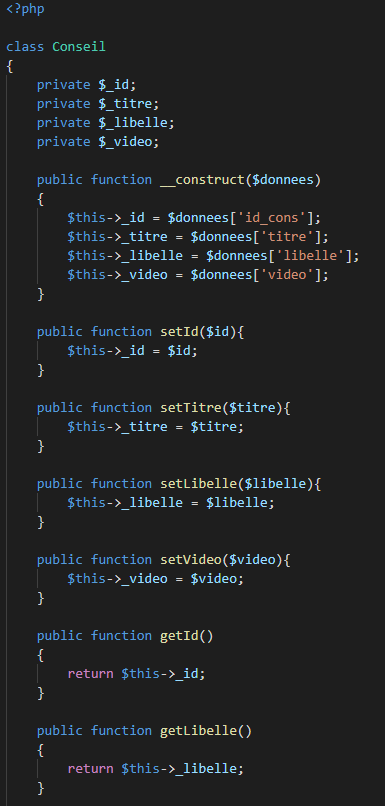
La page est codée de la manière suivante :

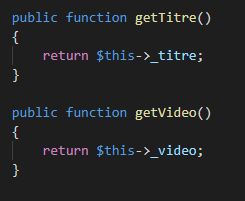


On importe d’abord les conseils dans un tableau et je les affiche grâce au code PHP ci-dessous :



Voyons comment est construit notre classe conseils :





Pour gérer mes objets conseils nous avons créé le fichier Conseils Manageur :

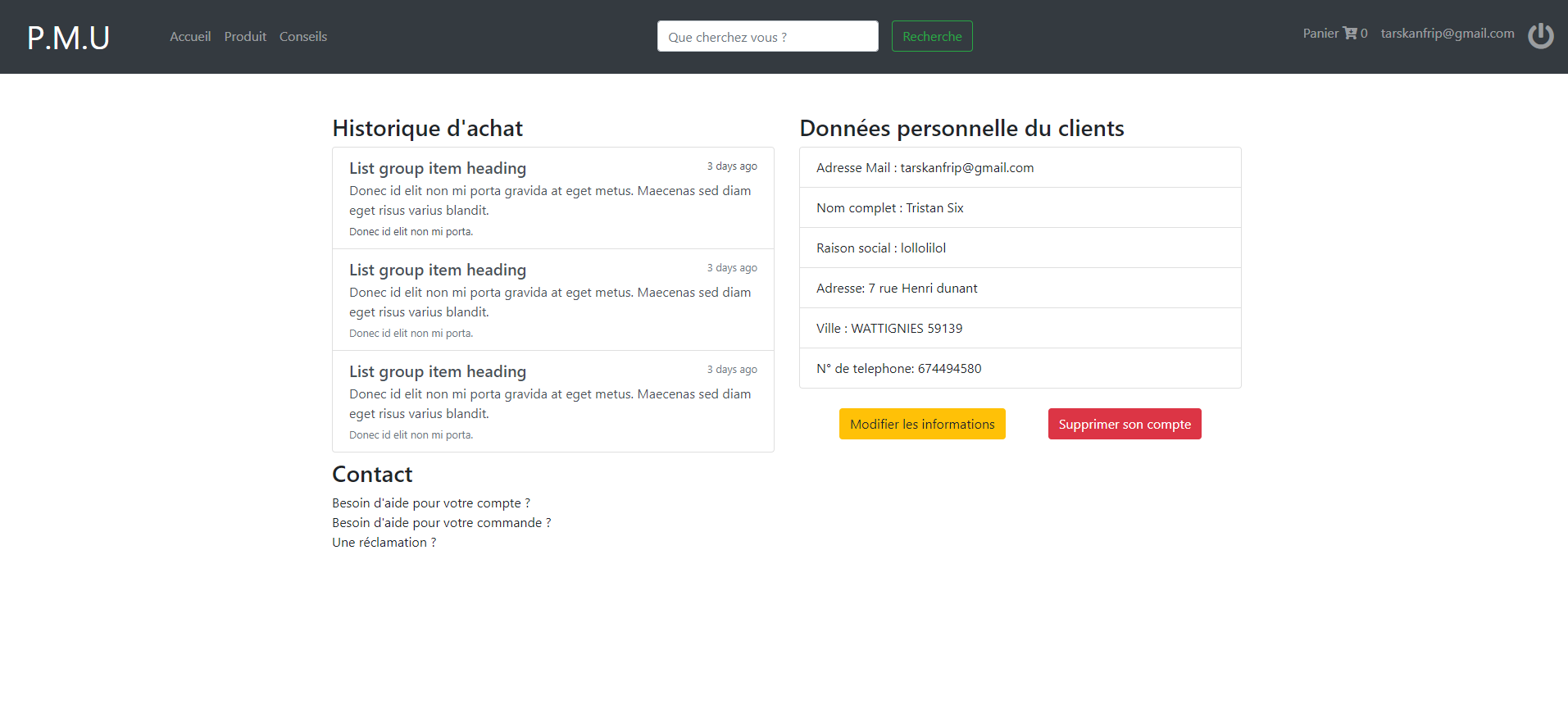


# Chapitre utilisateur

Lorsque l’utilisateur est connecté, il peut aller consulter ses informations en cliquant sur son adresse mail affichée de la manière suivante :

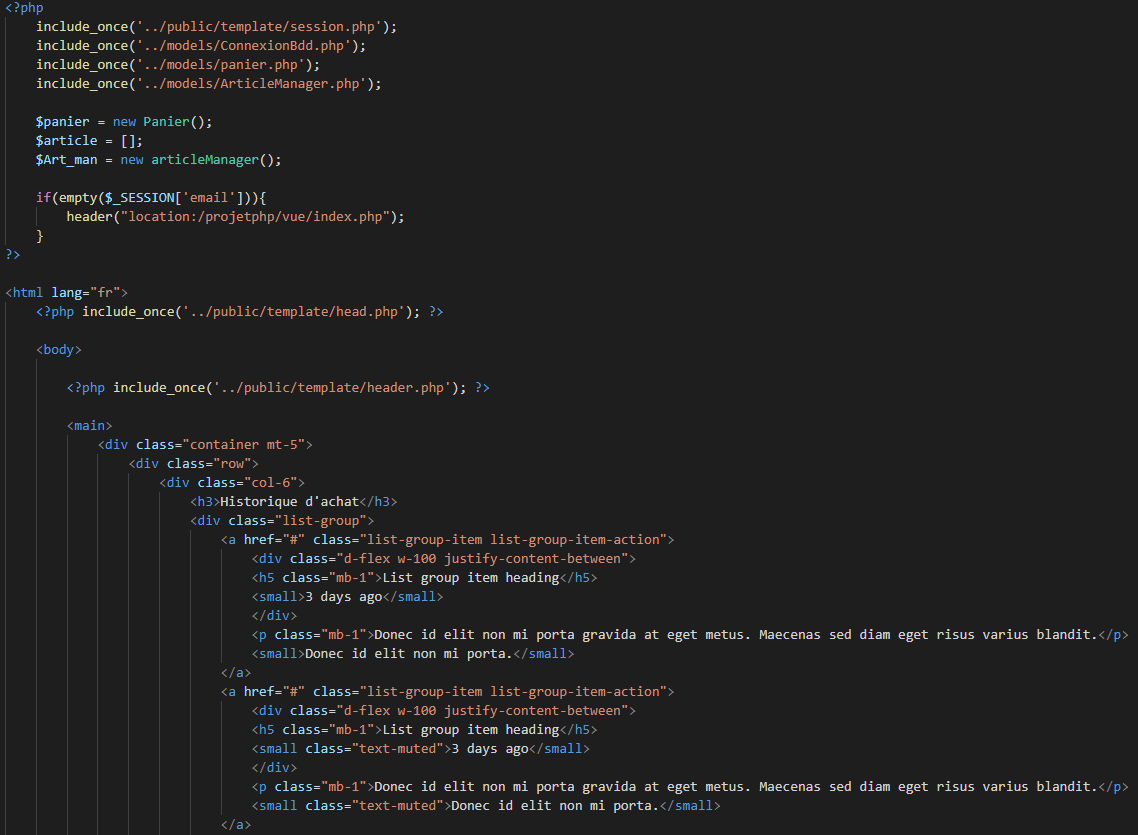


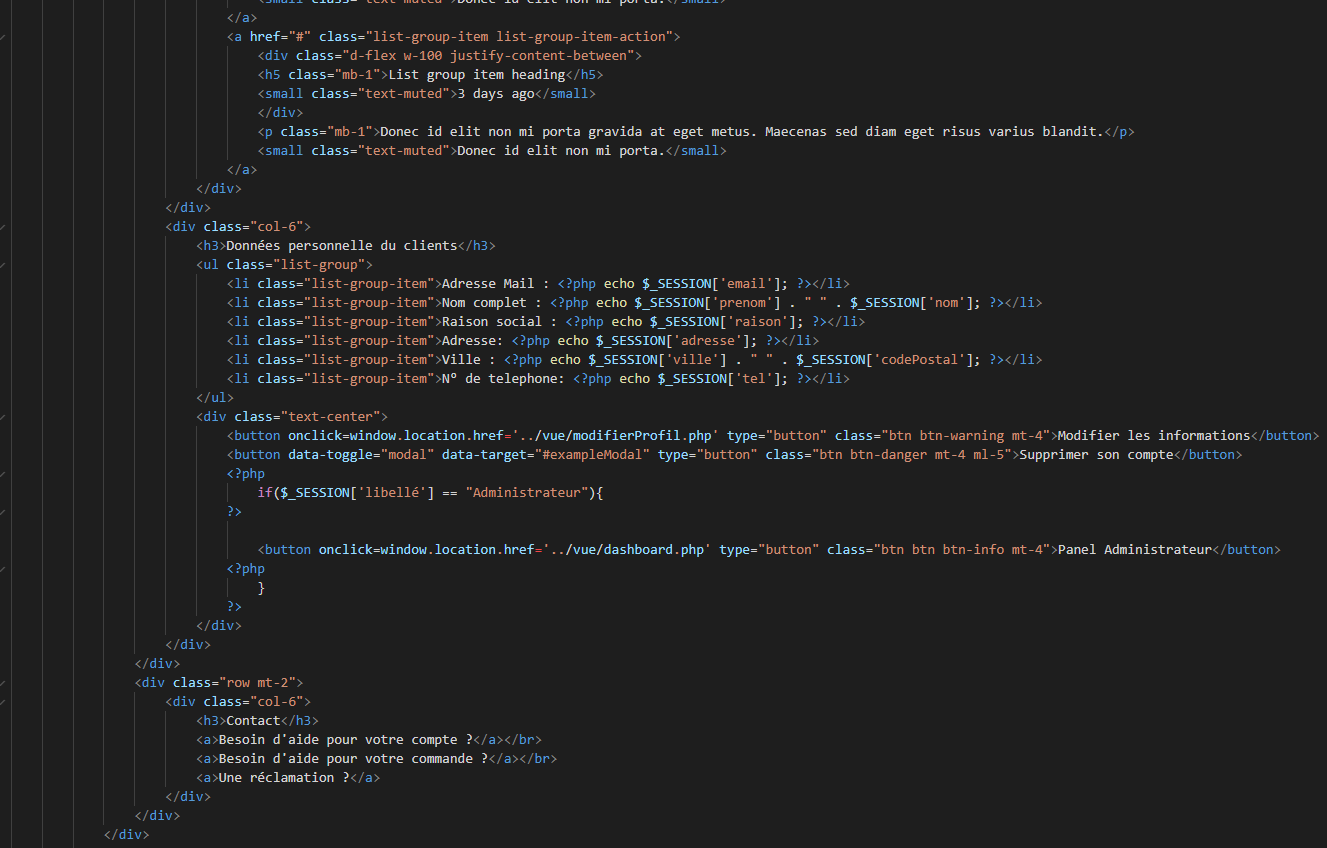
Une fois ceci fait, vous allez arriver dans la page de gestion utilisateur qui se présente comme ceci :

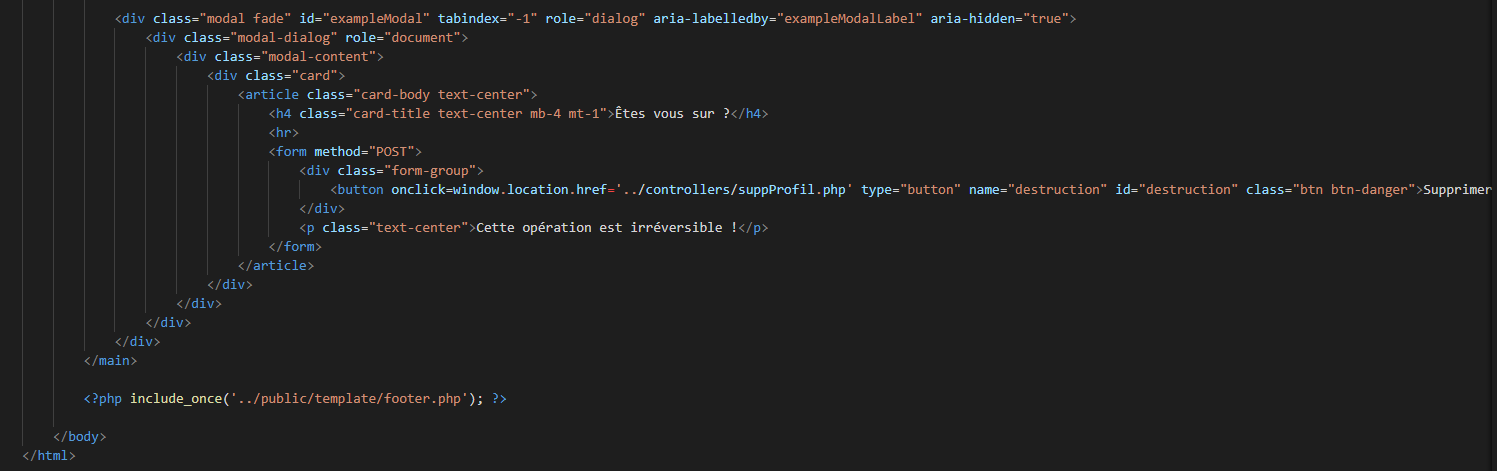


Comme vous pouvez le voir toutes les informations personnelles du client apparaissent, ainsi que ses achats et des options de contacte en cas de soucis.

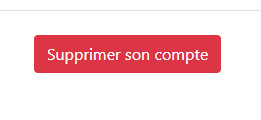
Voici le code de la page :

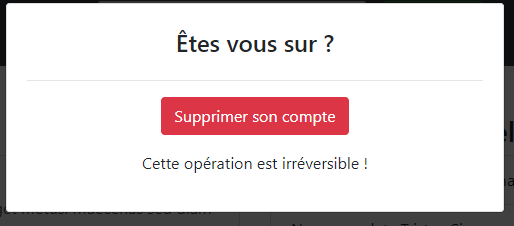




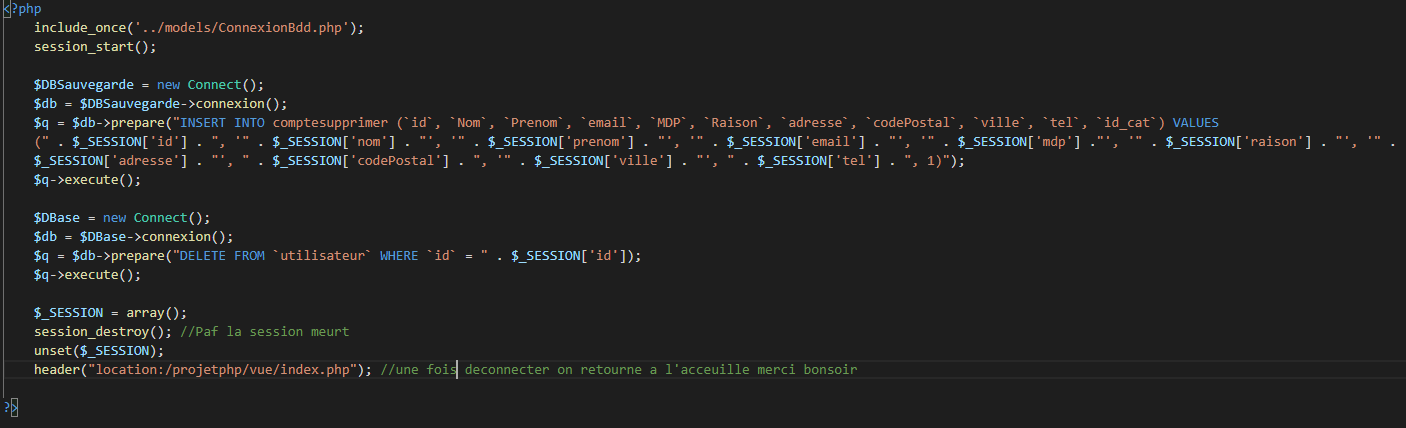


Lorsque vous voulez supprimer votre compte, vous n’avez qu’à cliquer sur le bouton supprimer qui fera apparaitre un modal, vous demandant de confirmer votre action comme ceci :





Une fois que vous cliquez dessus votre compte est supprimé et sauvegardé dans une table spéciale compteSupprimer le code suivant s’exécute :



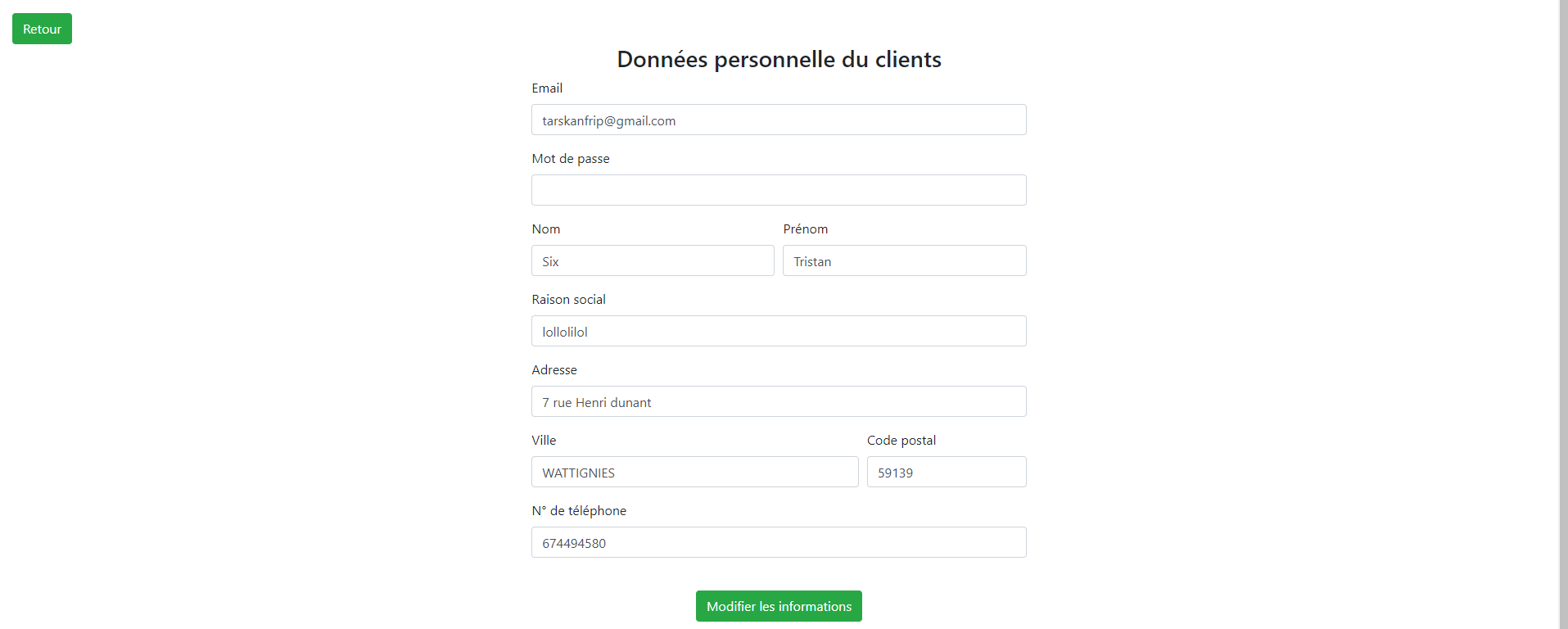
Le compte est sauvegardé à la fin de l’opération, voici la structure de la table qui est en tout point similaire à la table utilisateur :



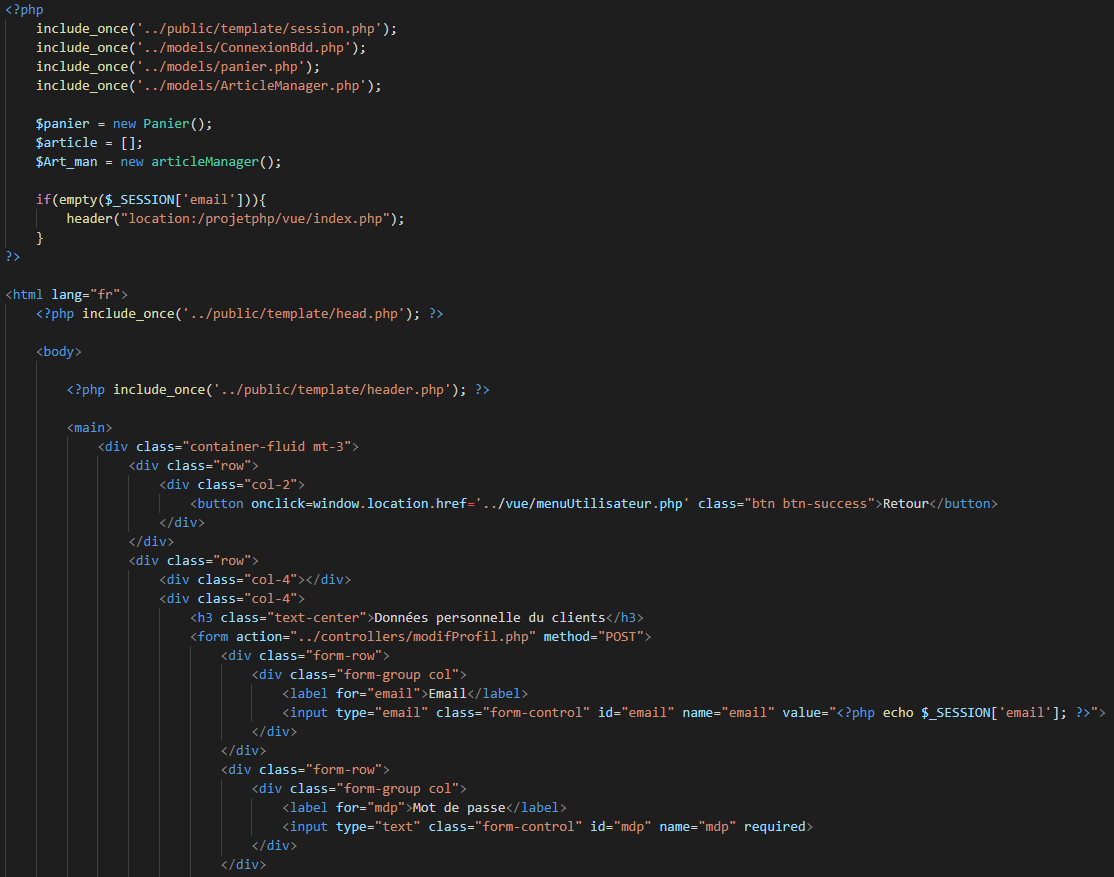
Passons à la seconde option, la modification de données :

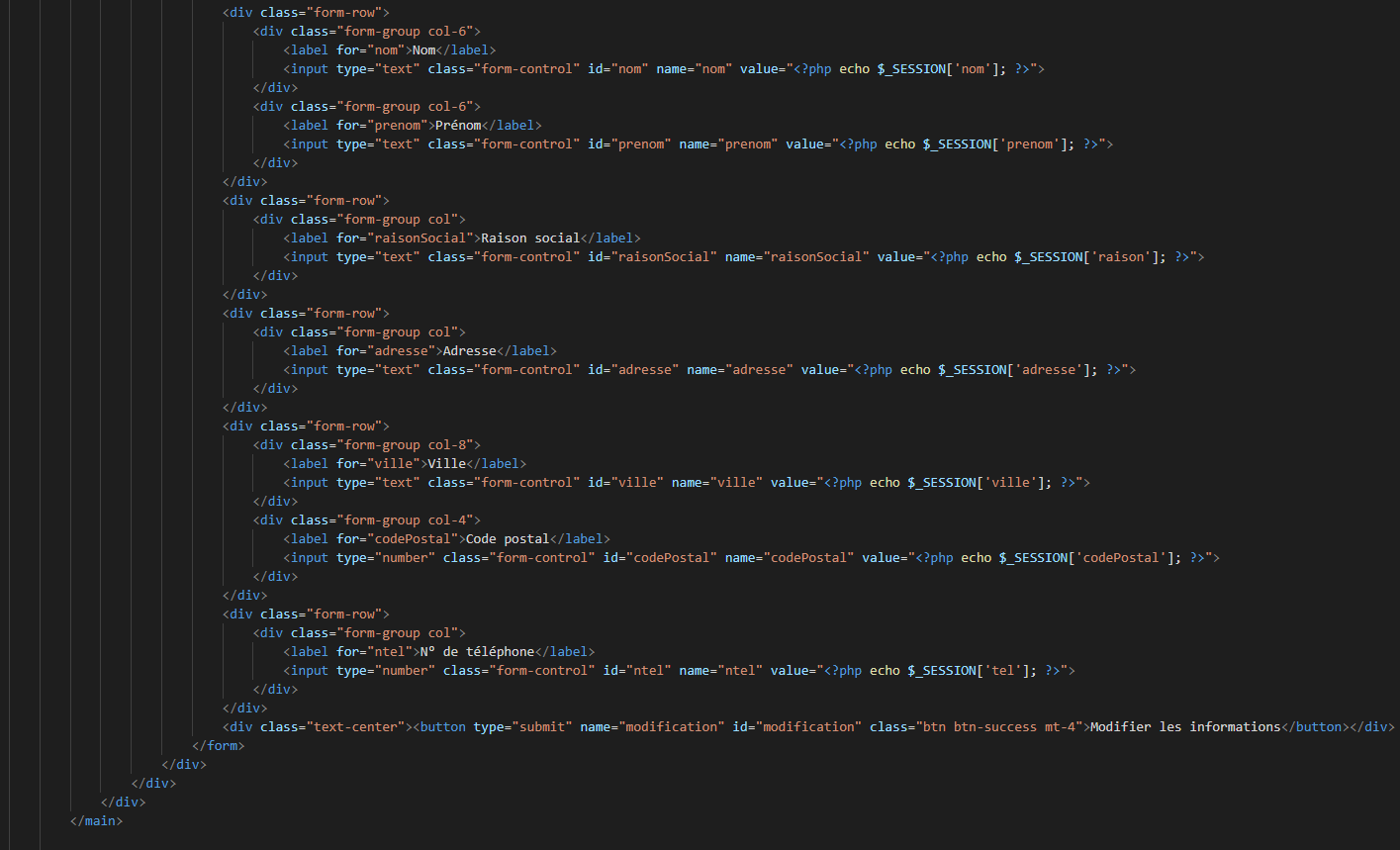


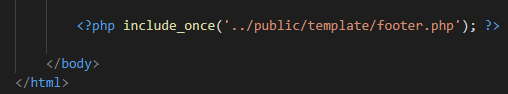
Une fois que vous avez cliqué sur le bouton vous êtes dirigé sur la page suivante :



Dont le code est le suivant :

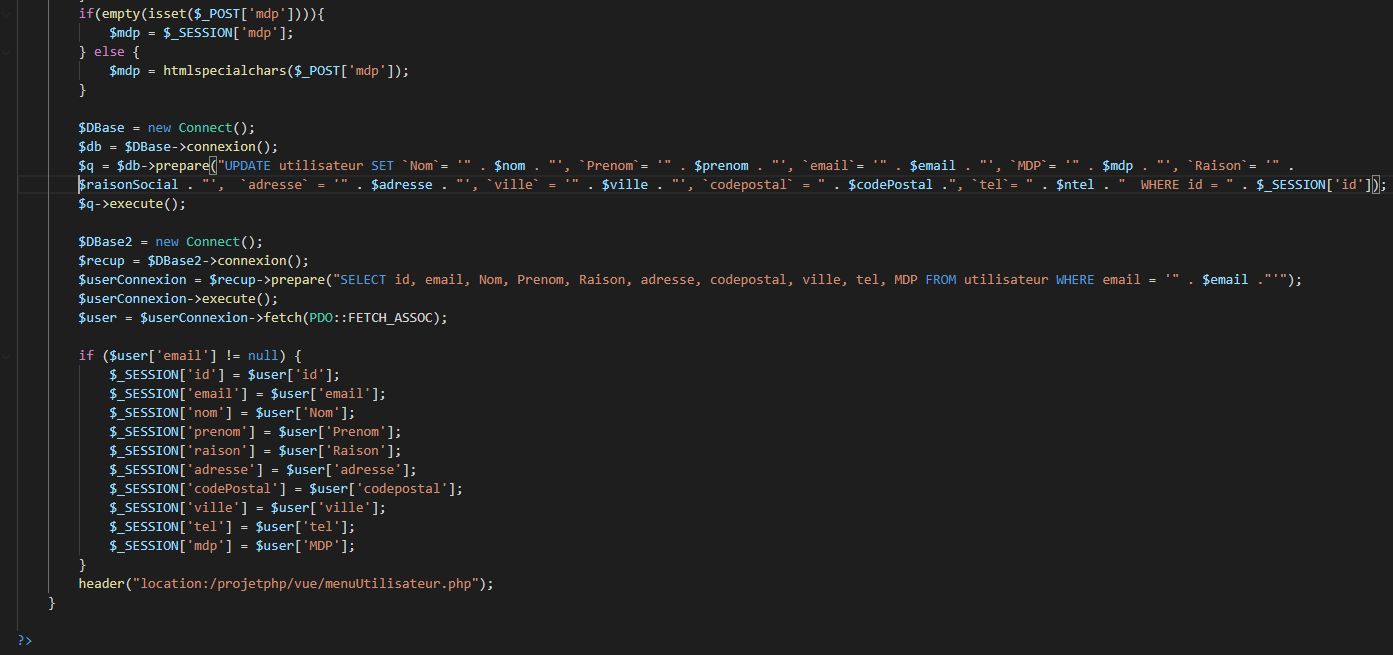






On y retrouve toutes nos données sauf notre mot de passe pour des raisons de sécurité, vous pouvez à partir de là, modifier vos données personnelles voici le code pour la modification :



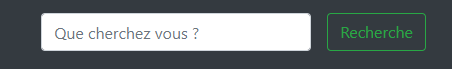


Comme vous pouvez le voir depuis plusieurs formulaires on utilise htmlspecialchars qui est une autre sécurité qui nous protège des injection SQL dans notre base de données.

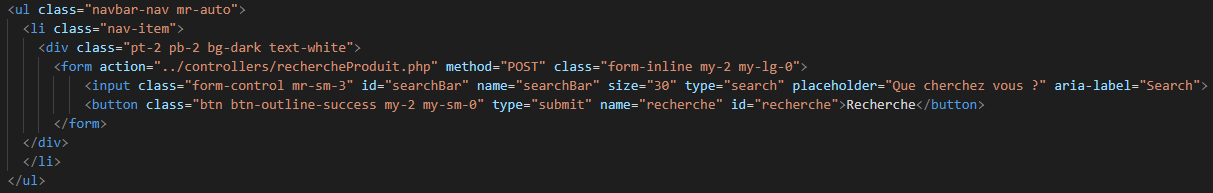
Une fois vos données modifiées, vous êtes renvoyé sur votre page utilisateur pour constater vos changements d’information.

# Chapitre Barre de recherche produit

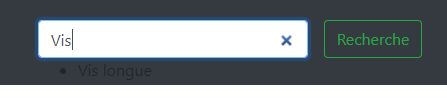
La barre de recherche produit se présente de la manière suivante :



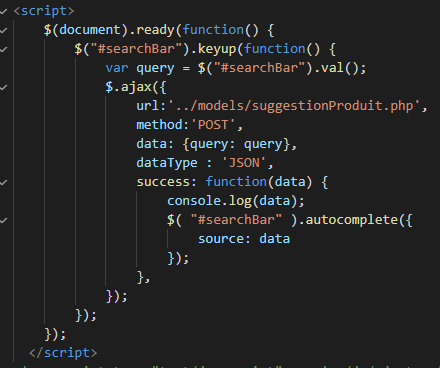
Avec le code suivant :



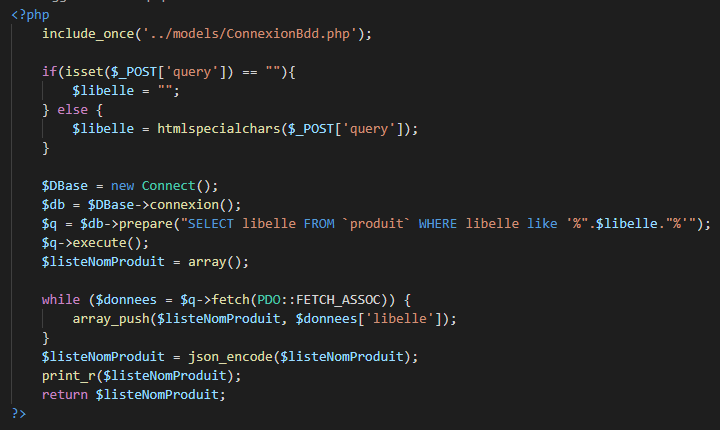
Il propose une suggestion de produit grâce au Framework jQuery qui se manifeste de la manière suivante quand on écrit dans la barre de recherche :



Voici le code Javascript utilisé pour rendre les recherches dynamiques :



Voici le code PHP pour faire les requêtes du côté serveur :



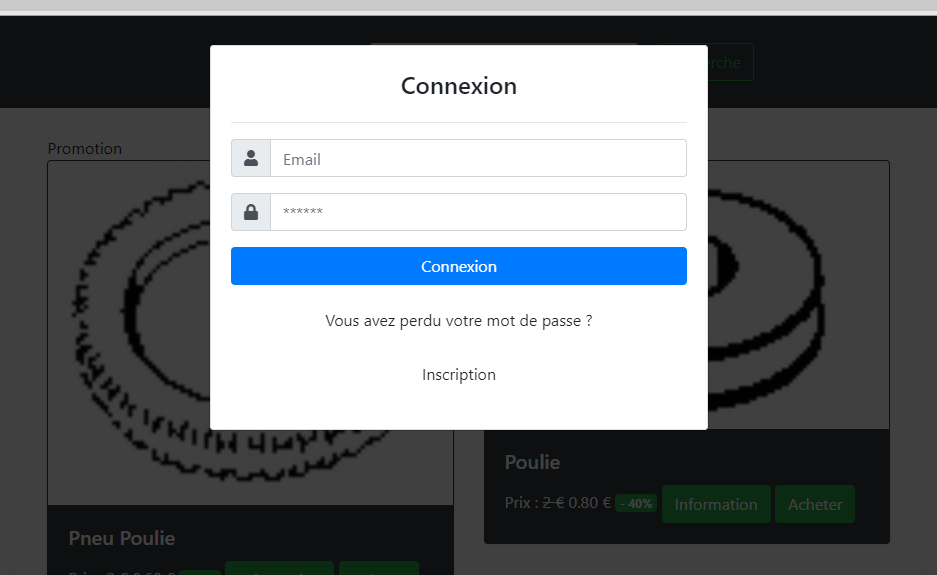
Avec celui-ci nous récupérons les noms des produits en liste puis une fois que l’on a cliqué dessus il suffit de cliquer sur le bouton recherche pour activer le code suivant :



Qui nous redirige sur la page du produit qui se présente comme ceci et sera expliqué dans un autre chapitre :



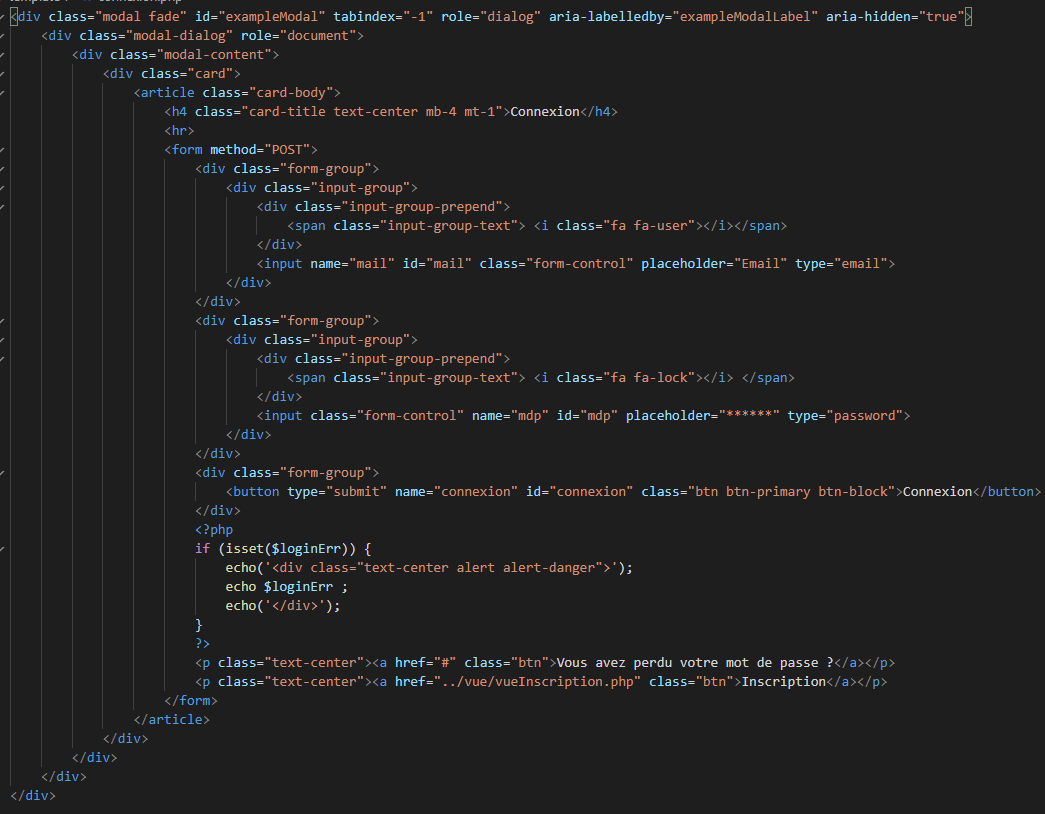
# Chapitre Portail de Connexion



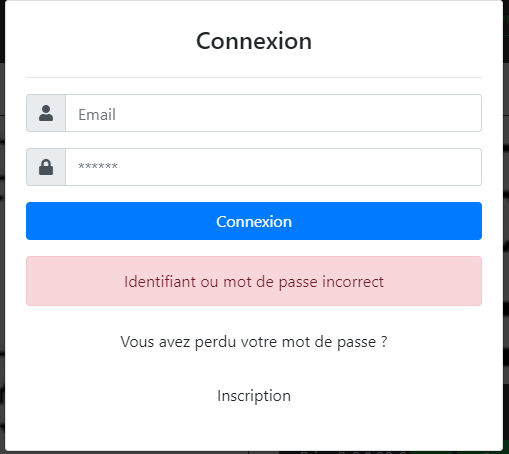
L’affichage de notre portail de connexion s’active par un modal qui est un élément animé du Framework Bootstrap et qui s’active de cette manière :



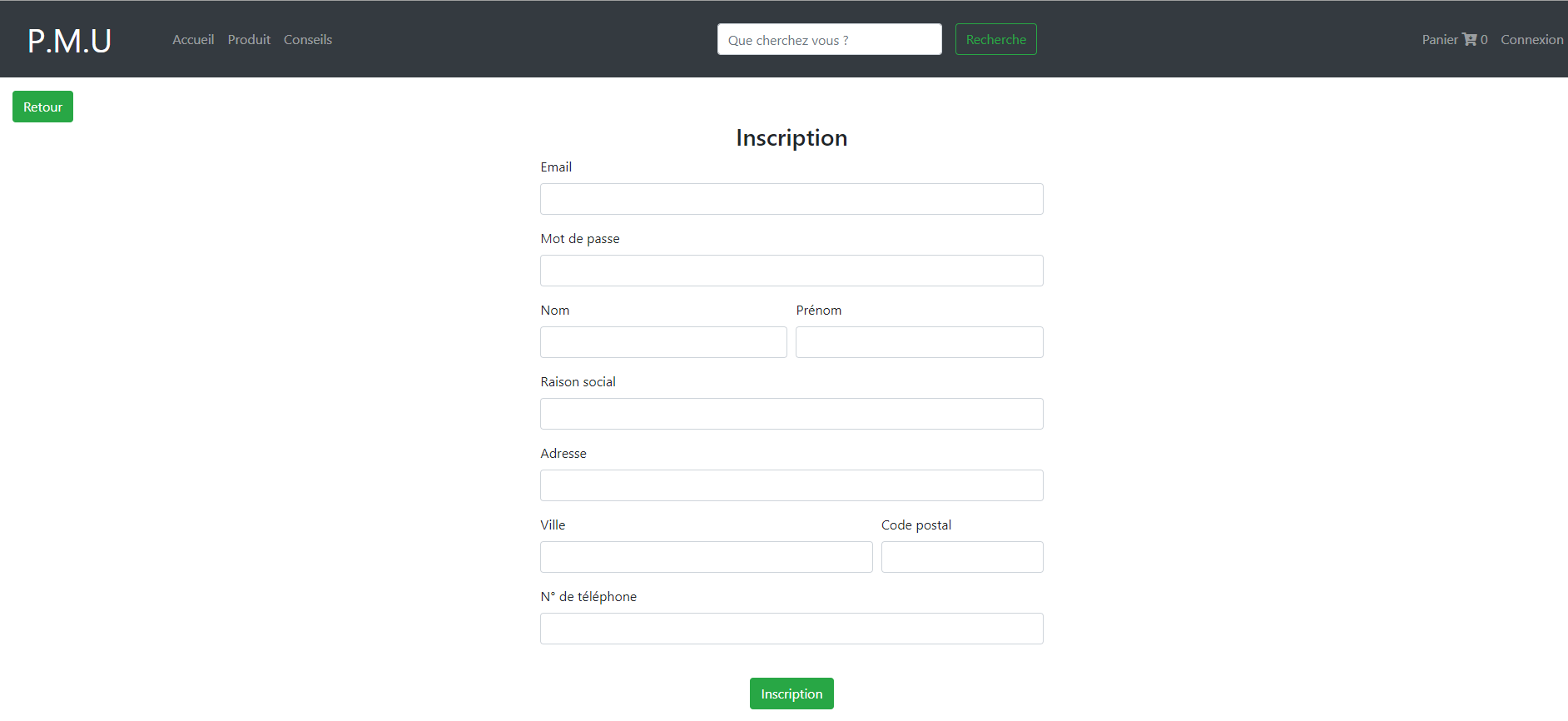
Selon si vous êtes connecté ou non vous ne verrez pas la même chose et cela est géré avec mon code PHP au-dessus, lorsque vous cliquez sur « Connexion » le portail de connexion apparaît et si vous regarder du côté des classes il y a un élément qui s’appelle data-target c’est lui qui va appeler la div viser par un ID qui est ici #exempleModal et qui vise le code suivant :



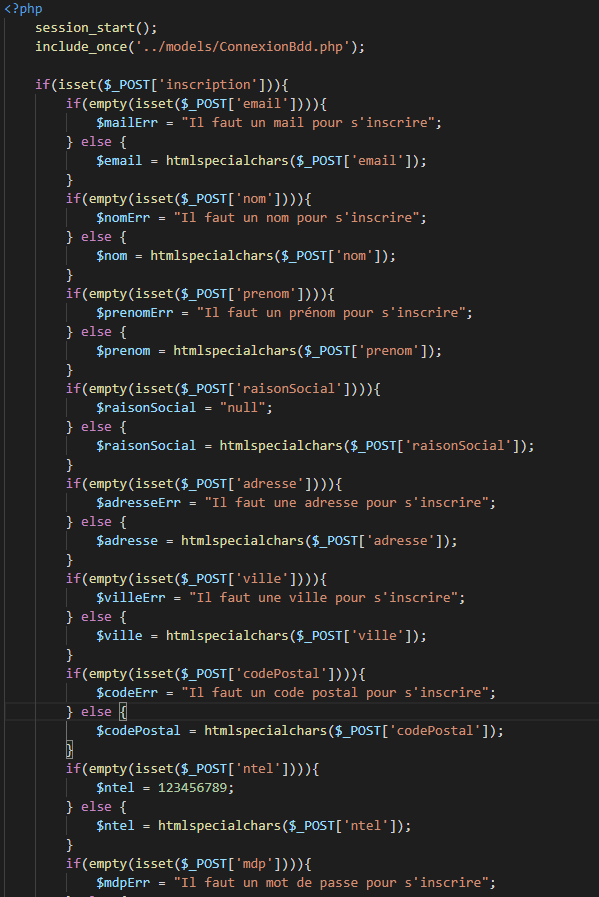
Cette section une fois appelée nous permettra de nous connecter en remplissant les inputs, une alerte en cas d’erreur d’identifiant à été mise en place comme vous pouvez le constater à la fin et qui s’affiche comme ceci :

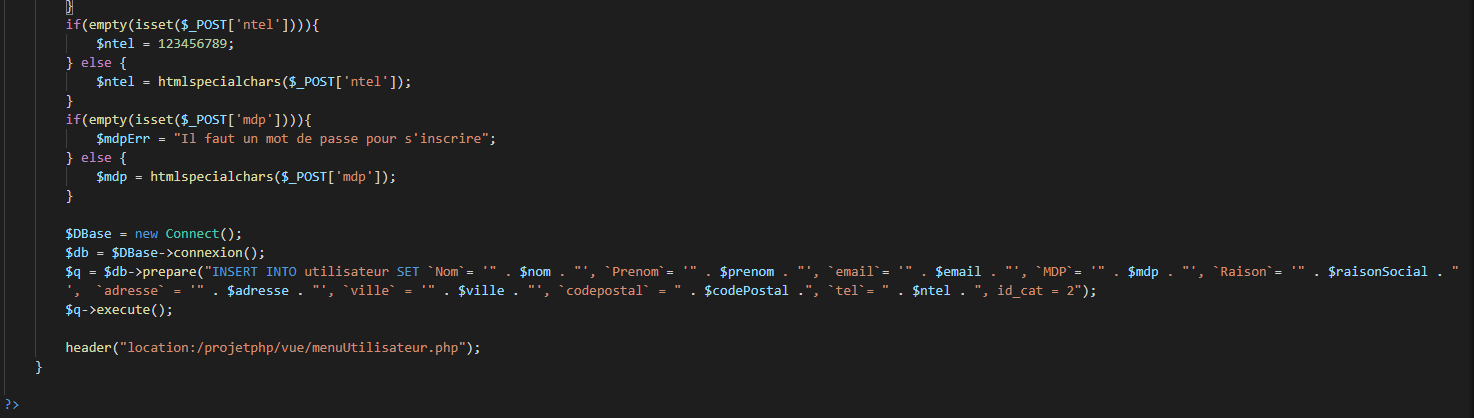


Les inscriptions fonctionnent il suffit de cliquer dessus pour arriver sur la page suivante :

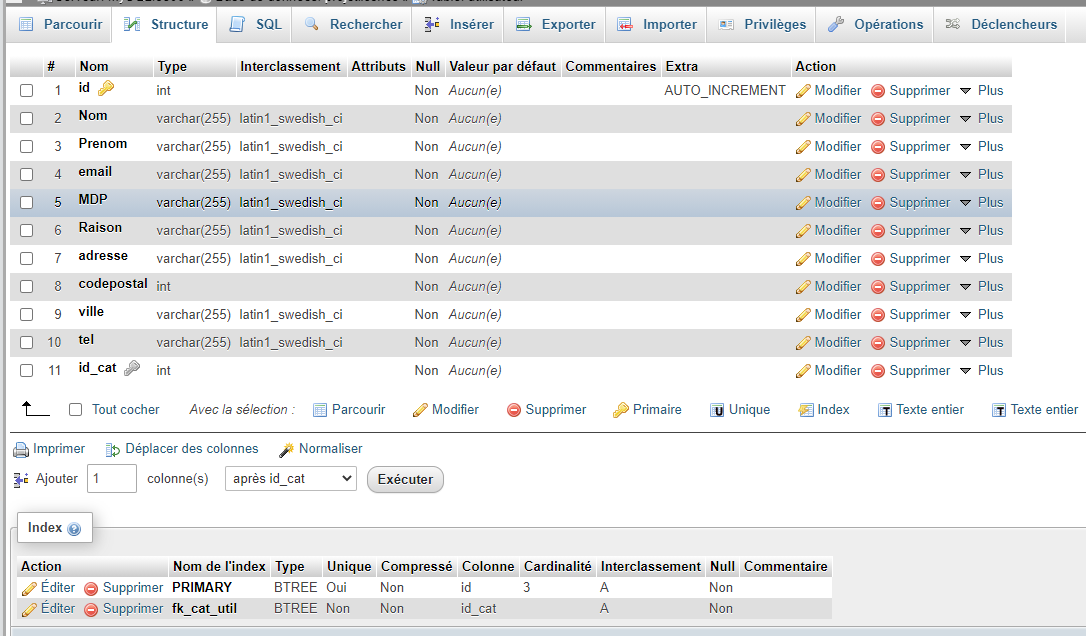


Une fois les inputs remplies vous pouvez cliquer sur inscriptions en cas de non complétion de l’inscription vous êtes éjecté sur la page principale, ensuite le code PHP qui gère l’inscription :

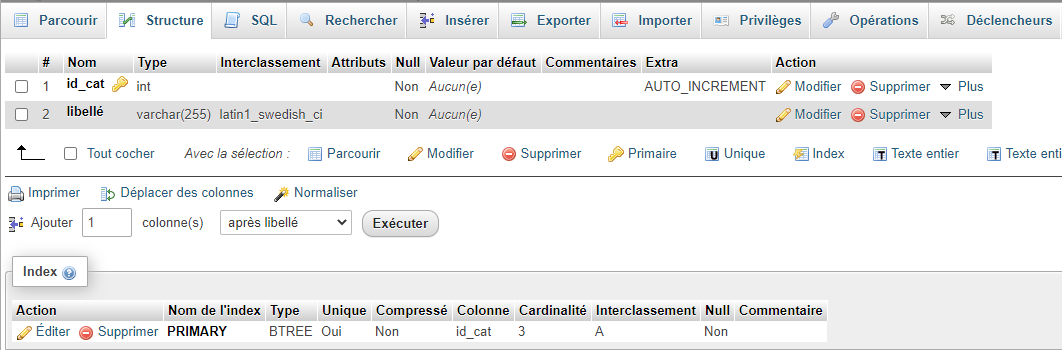




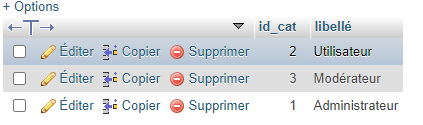
Une fois inscrit vous pouvez vous connectez voici la table de la base de données pour les utilisateurs :



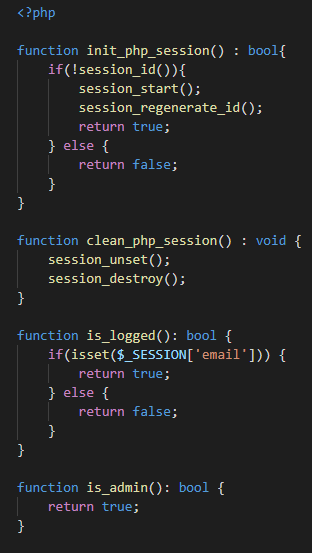
Il y a une clé étrangère vers la table catégorie utilisateur qui gère les droits utilisateurs et dont la structure est la suivante :

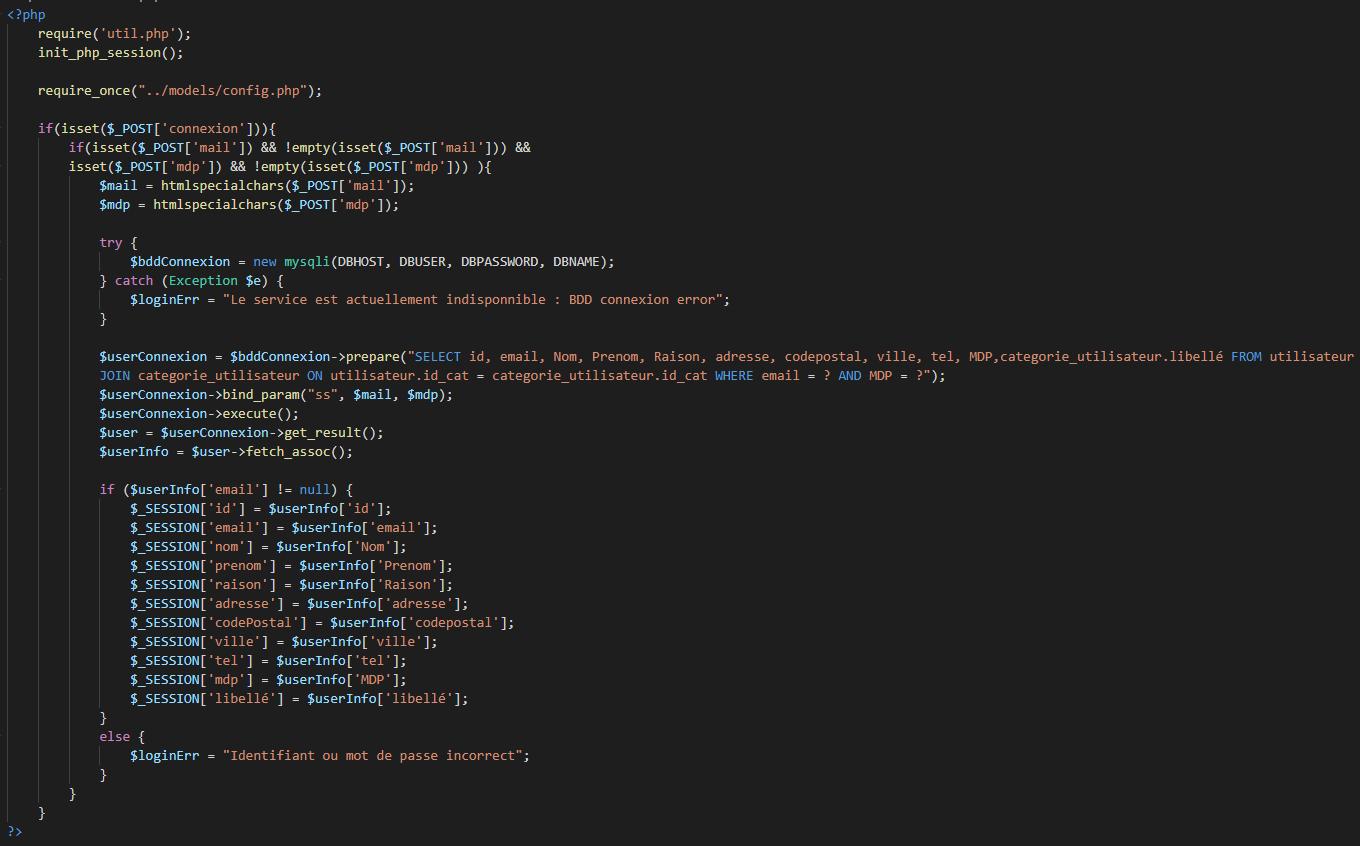


Voici les différents grades utilisateurs :



Voici le code PHP qui gère la connexion :



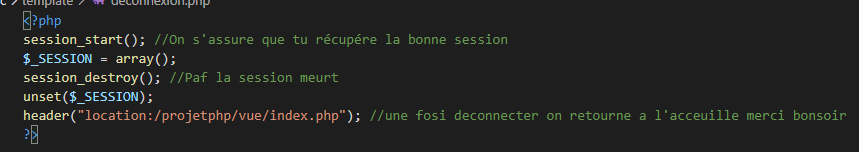


Il notifie une erreur au client s’il se trompe à la fin, et comme vous pouvez le constater nous n’utilisons pas de classe objet utilisateur car les super globale de PHP sont amplement suffisantes pour stocker le profil et les données importantes du client en toute sécurité.

Une fois connecté, le site se modifie comme ceci :



L’adresse mail du client apparait et un bouton power permet de se déconnecter voici le code :



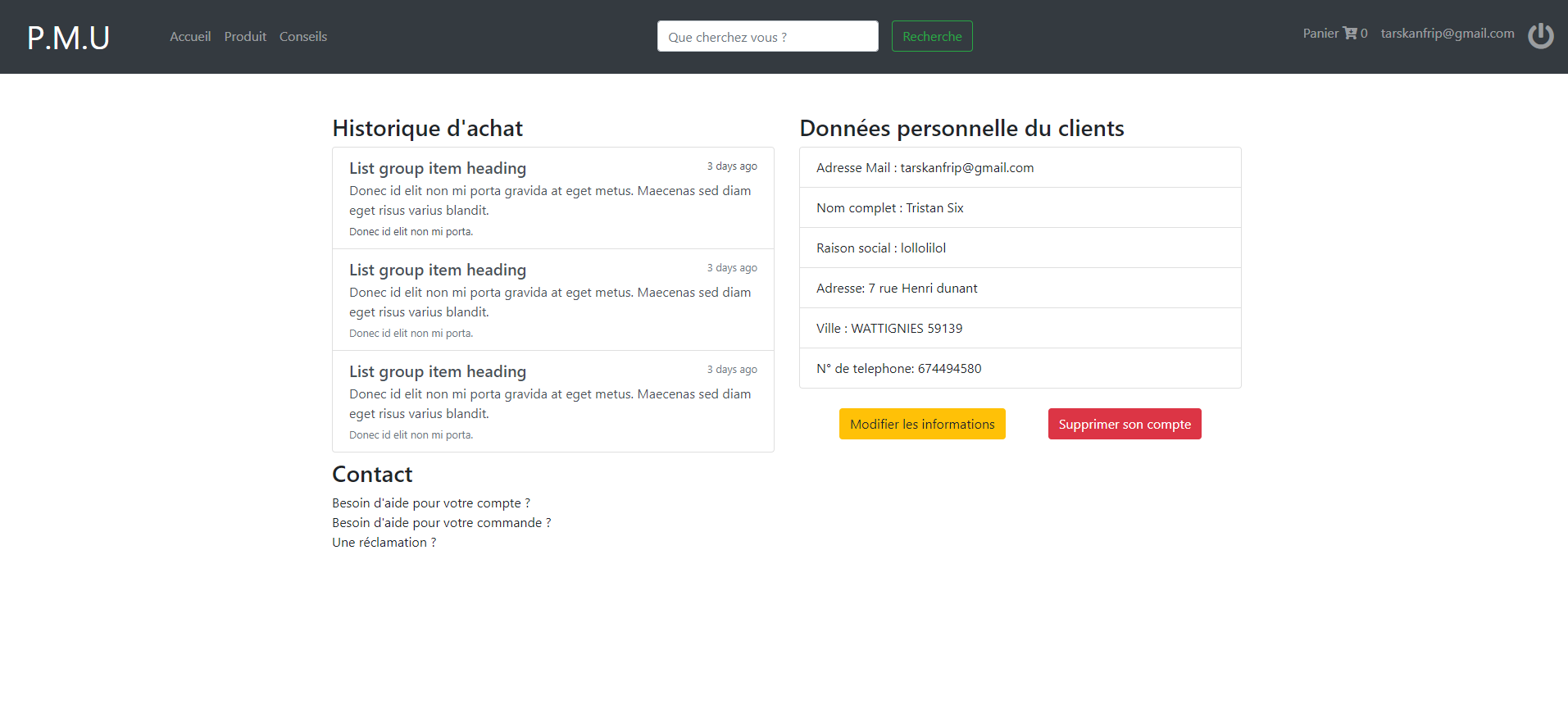
Notre code commenter parle de lui-même on s’assure que l’on vise la bonne session puis on la détruit pour sortir le client du circuit utilisateur et il retourne à la page d’accueil

# Chapitre utilisateur

Lorsque l’utilisateur est connecté, il peut aller consulter ses informations en cliquant sur son adresse mail affichée de la manière suivante :

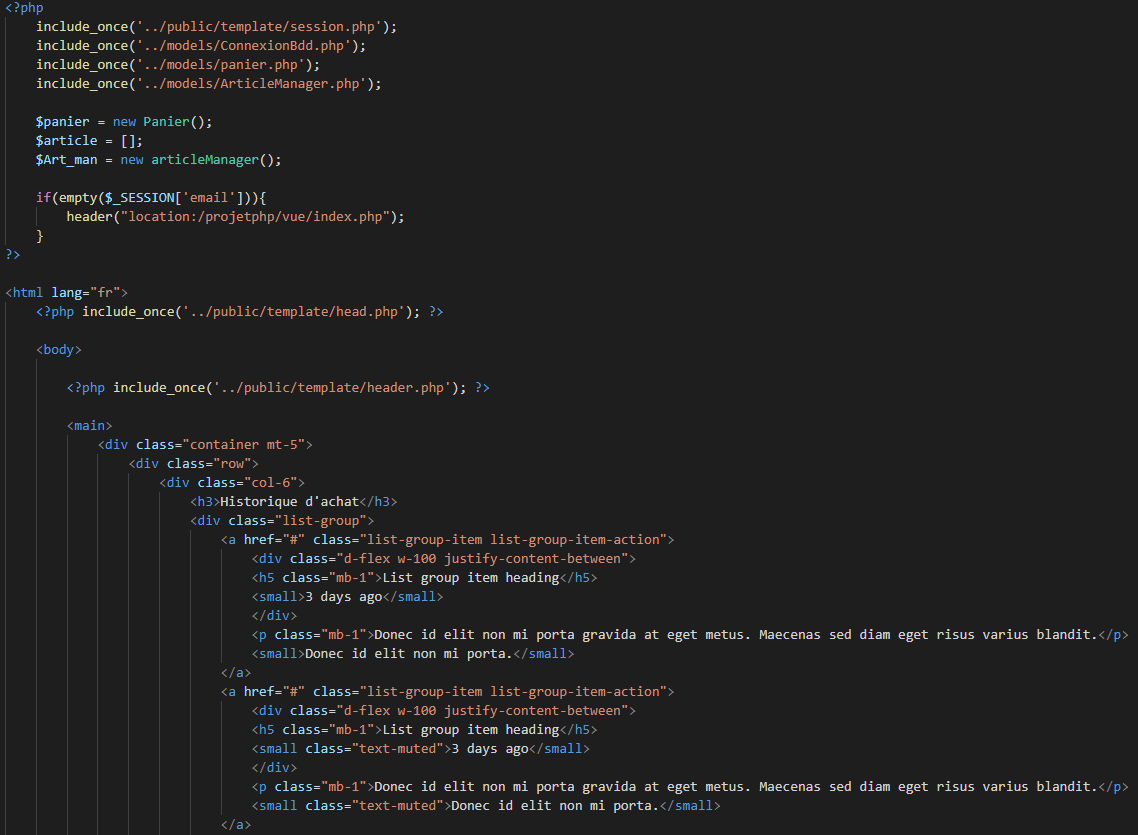


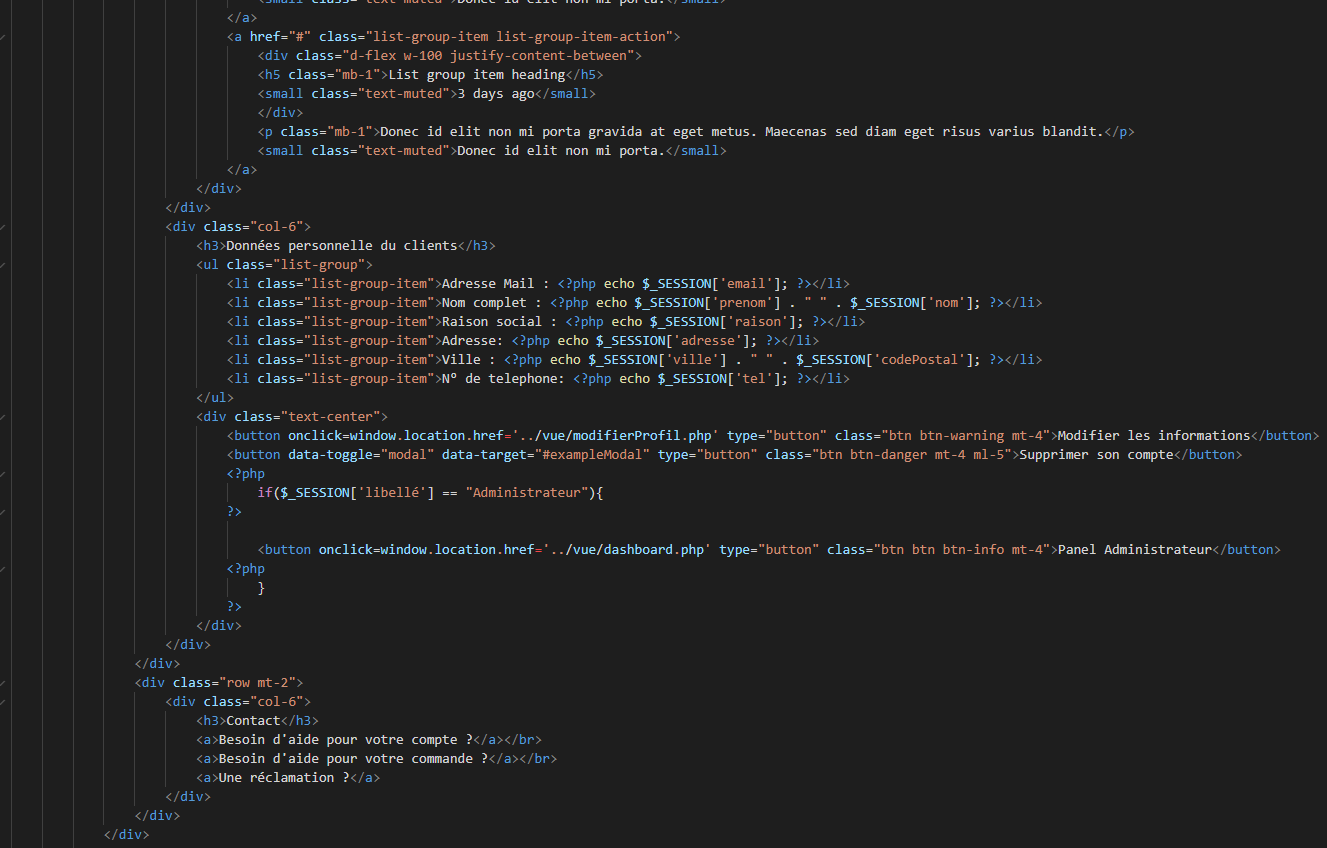
Une fois ceci fait, vous allez arriver dans la page de gestion utilisateur qui se présente comme ceci :

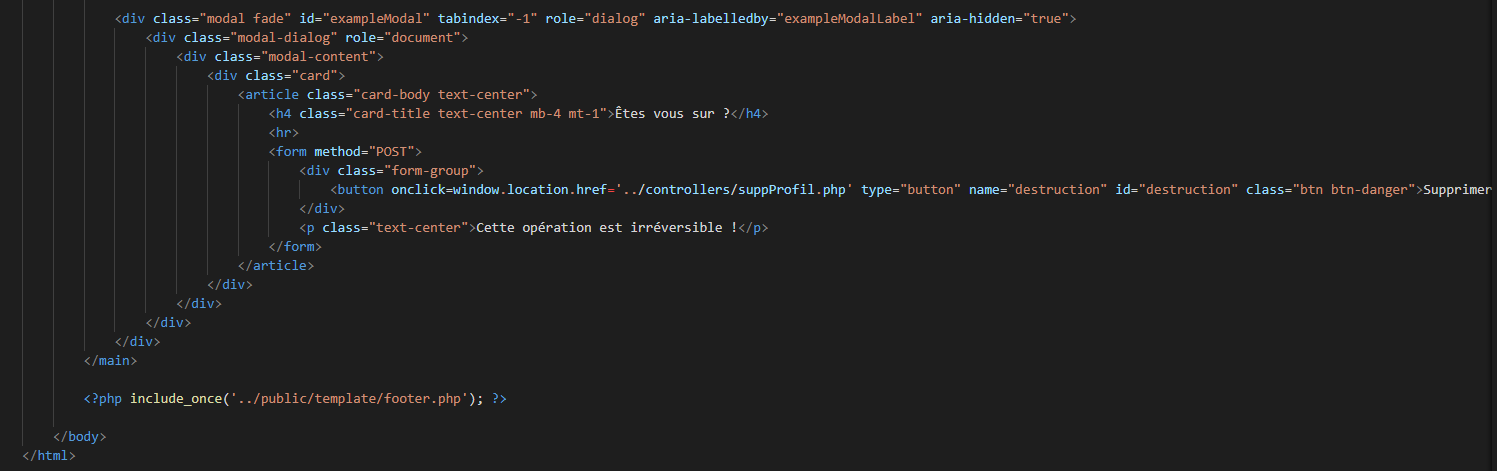


Comme vous pouvez le voir toutes les informations personnelles du client apparaissent, ainsi que ses achats et des options de contacte en cas de soucis.

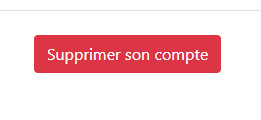
Voici le code de la page :

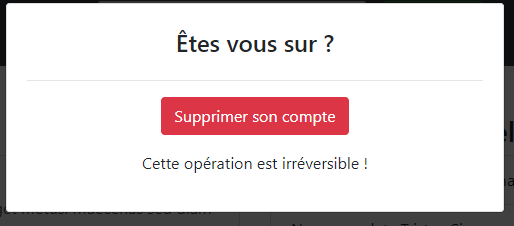




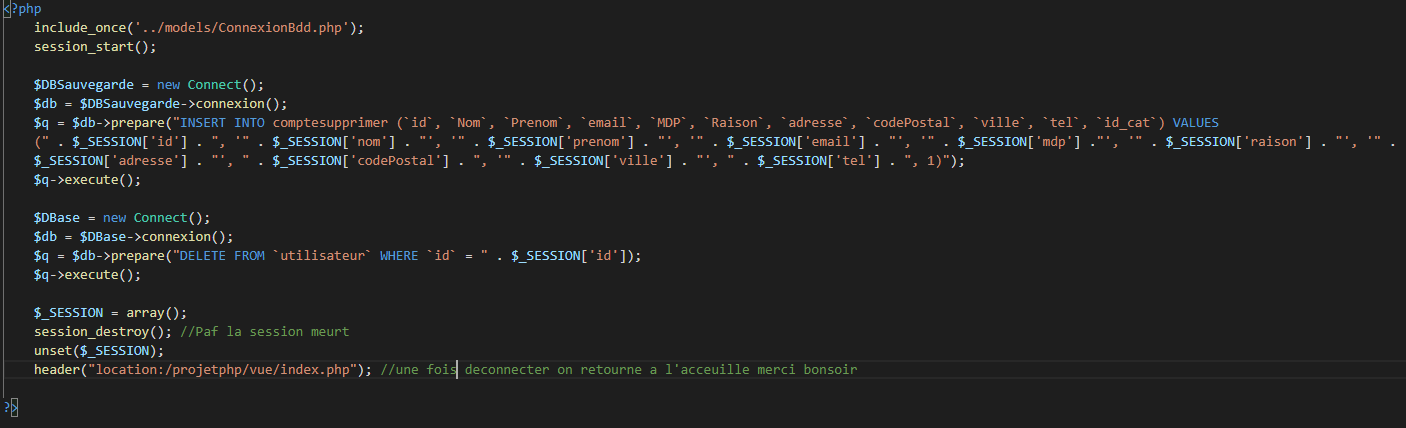


Lorsque vous voulez supprimer votre compte, vous n’avez qu’à cliquer sur le bouton supprimer qui fera apparaitre un modal, vous demandant de confirmer votre action comme ceci :





Une fois que vous cliquez dessus votre compte est supprimé et sauvegardé dans une table spéciale compteSupprimer le code suivant s’exécute :



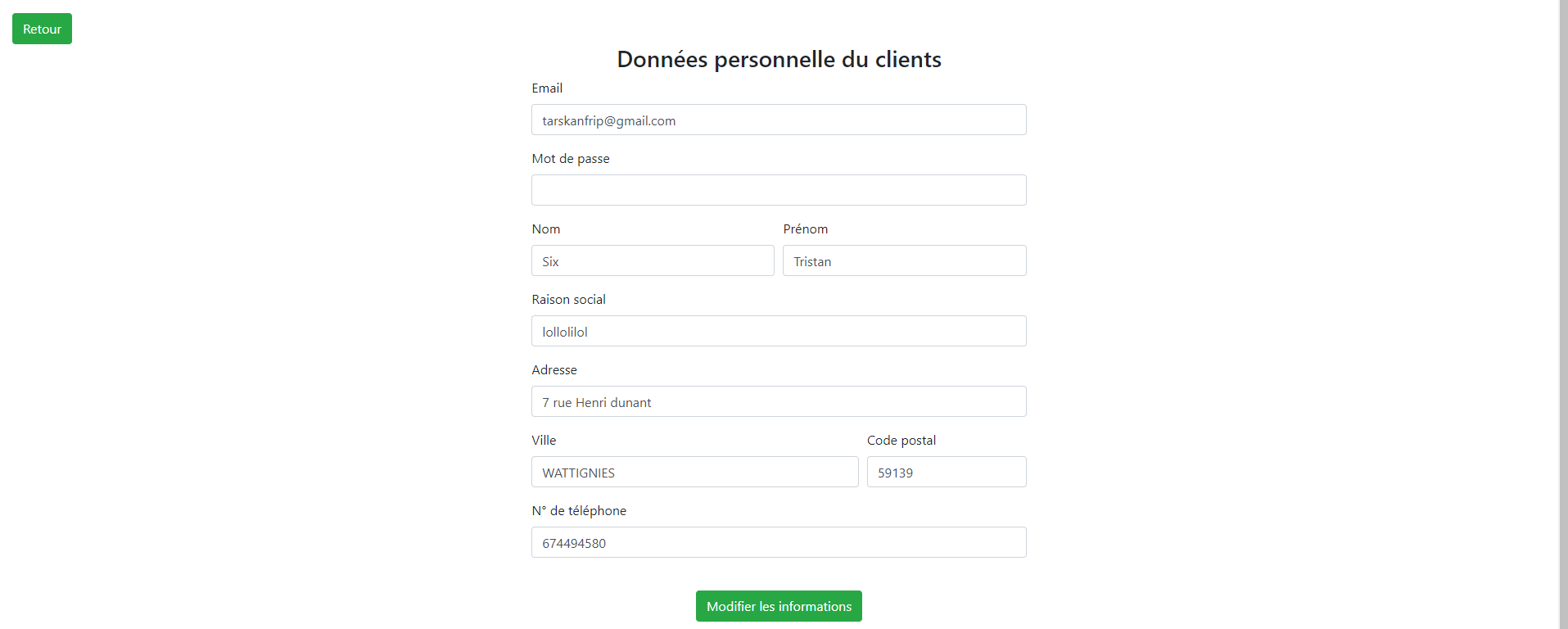
Le compte est sauvegardé à la fin de l’opération, voici la structure de la table qui est en tout point similaire à la table utilisateur :



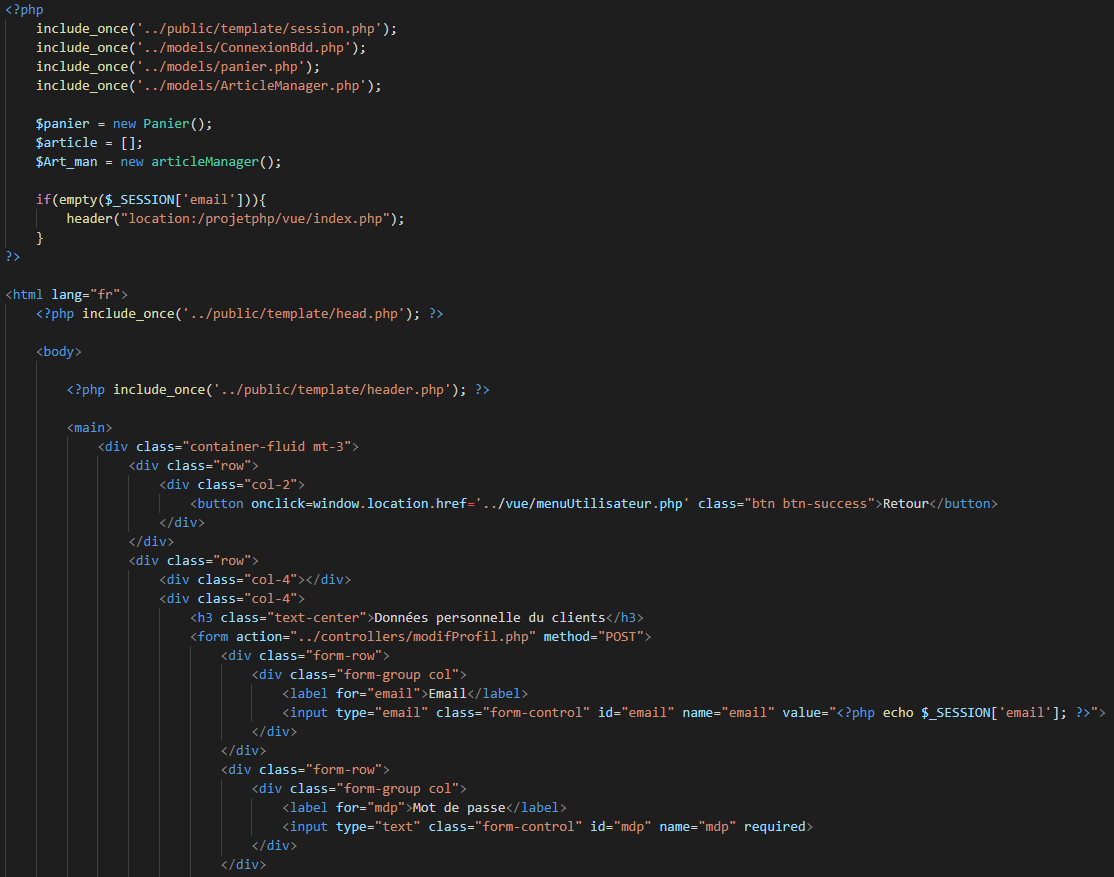
Passons à la seconde option, la modification de données :

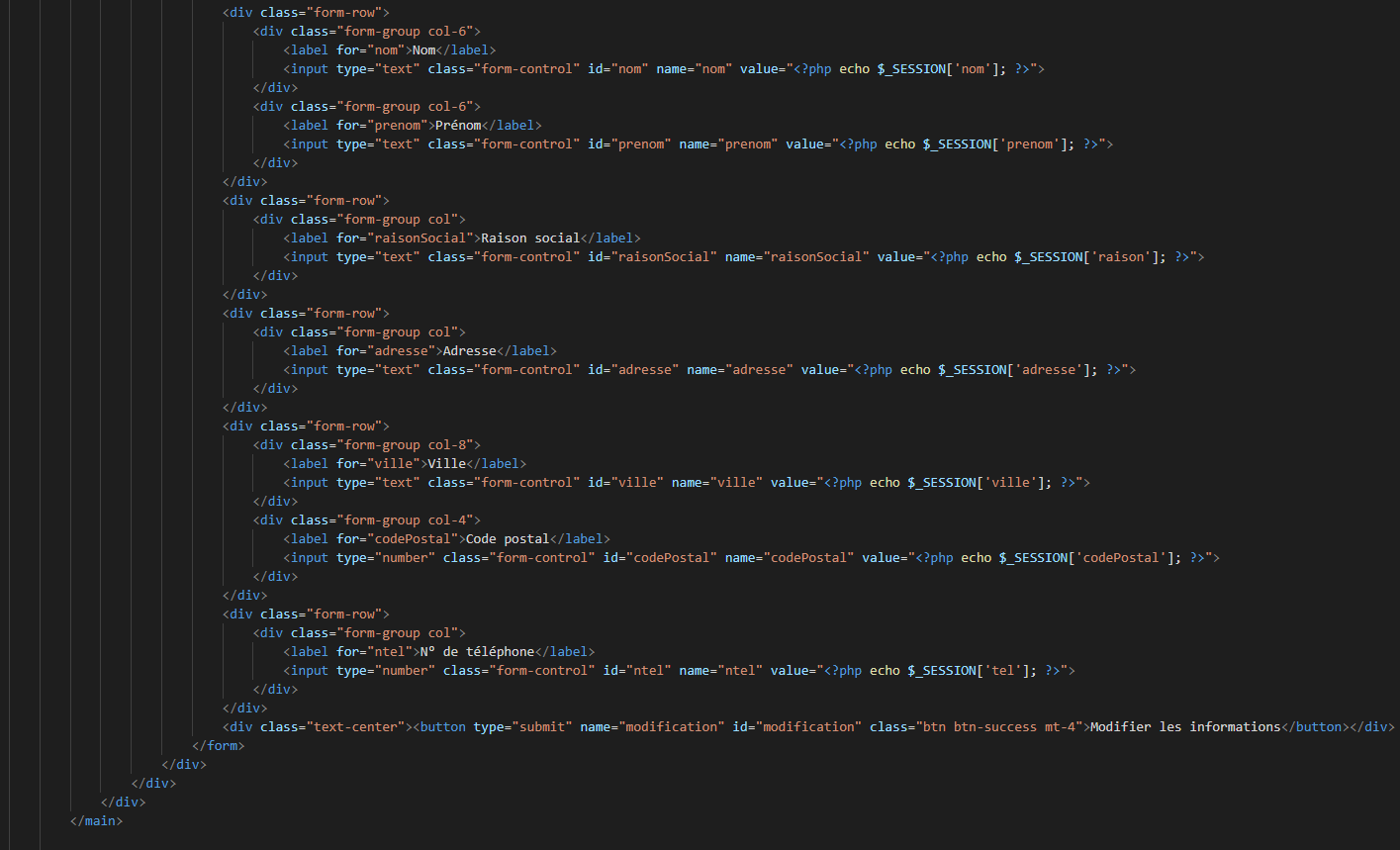


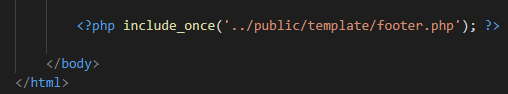
Une fois que vous avez cliqué sur le bouton vous êtes dirigé sur la page suivante :



Dont le code est le suivant :

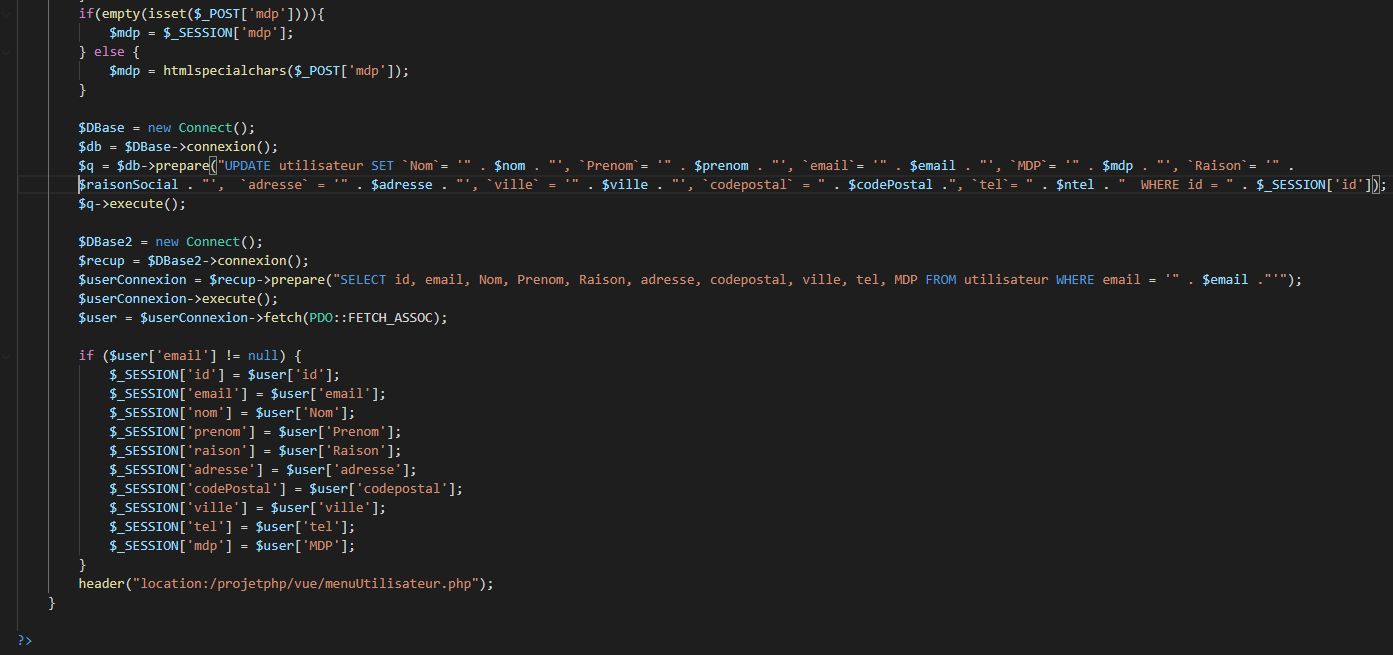






On y retrouve toutes nos données sauf notre mot de passe pour des raisons de sécurité, vous pouvez à partir de là, modifier vos données personnelles voici le code pour la modification :



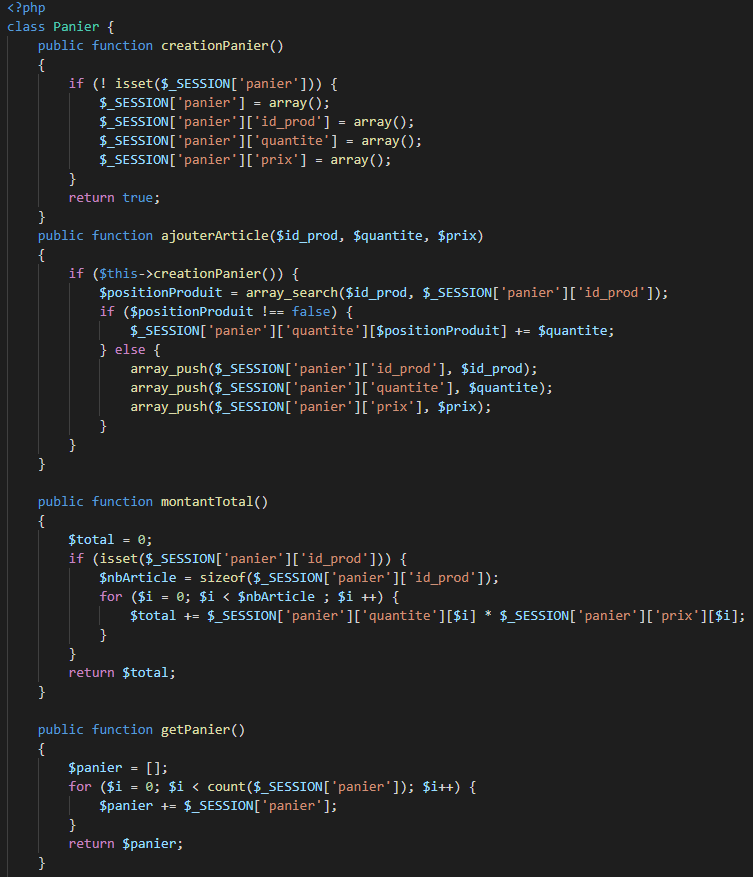


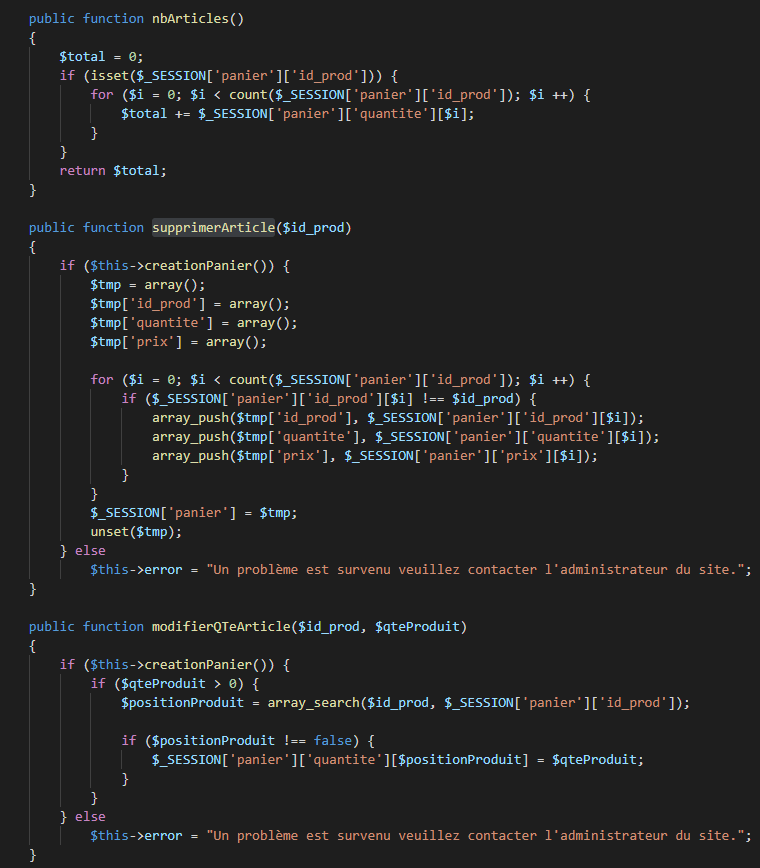
Comme vous pouvez le voir depuis plusieurs formulaires on utilise htmlspecialchars qui est une autre sécurité qui nous protège des injection SQL dans notre base de données.

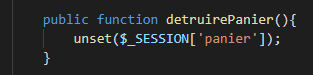
Une fois vos données modifiées, vous êtes renvoyé sur votre page utilisateur pour constater vos changements d’information.

# Chapitre Panier d’achat

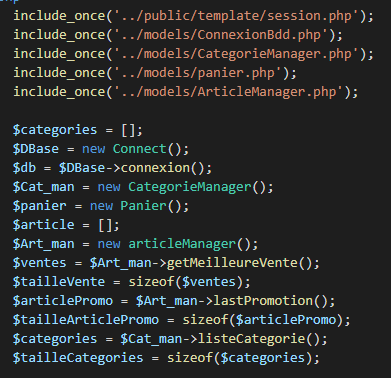
Le cas du panier est spécial car il utilise une classe et à la fois les variables super globales voici le code de la classe Panier (qui nous a été fournis par le professeur) avec une fonction mineure ajouter de « destruction de panier » crée par moi-même :



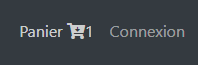




On l’appelle à chaque page pour être sûr qu’il soit bien initialisé comme ceci :



De ce fait, on peut prendre des articles dans notre panier connecté ou non exemple ci-dessous :



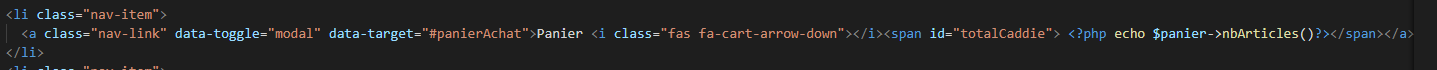
Le panier se présente au client comme ceci, il suffit juste de cliquer sur « panier » ou le caddie blanc et il apparait comme ci-dessous :

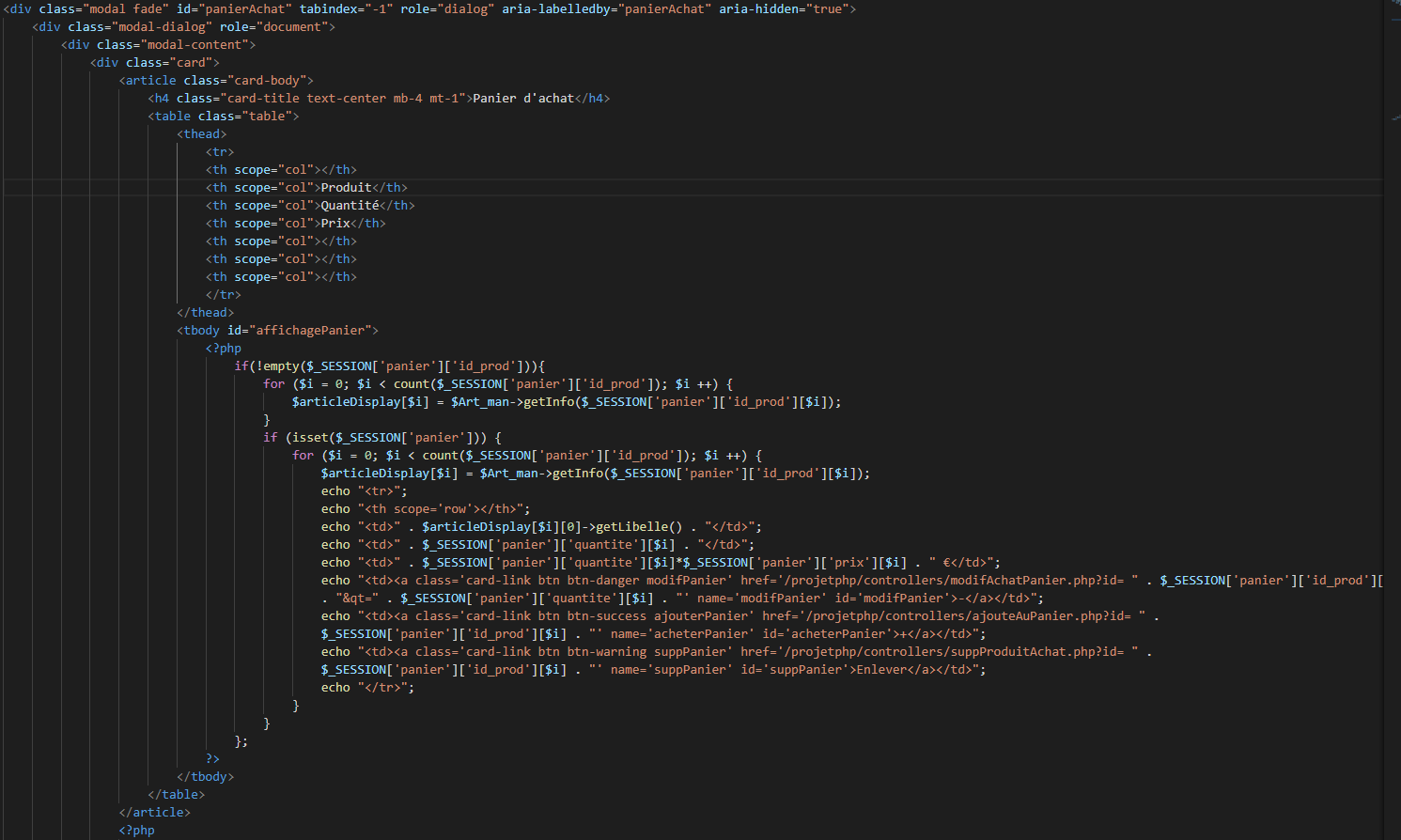


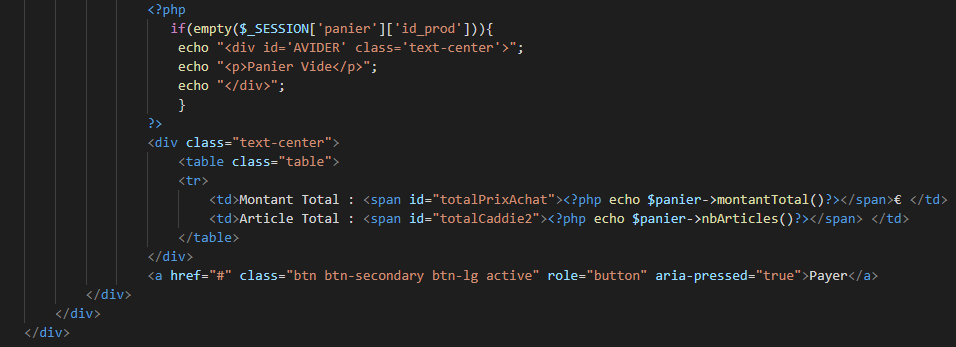
S’il y a un article et ensuite sans article :



Le code HTML pour le panier :







Afin de rendre le panier dynamique pour le client du code javascript a été écrit en conséquence :

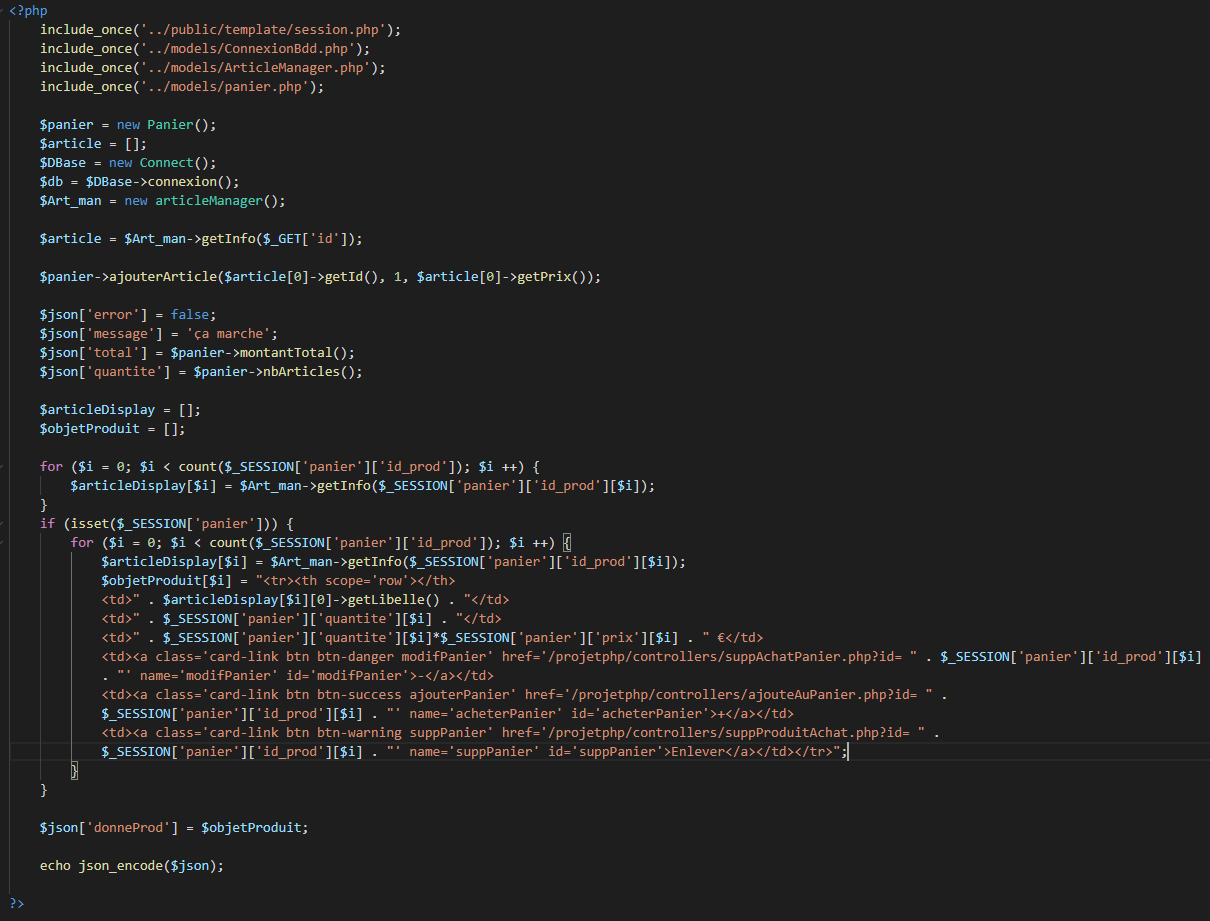


Raccordée à ce fichier qui recharge le panier d’achat avec les bonnes valeurs souhaitées :



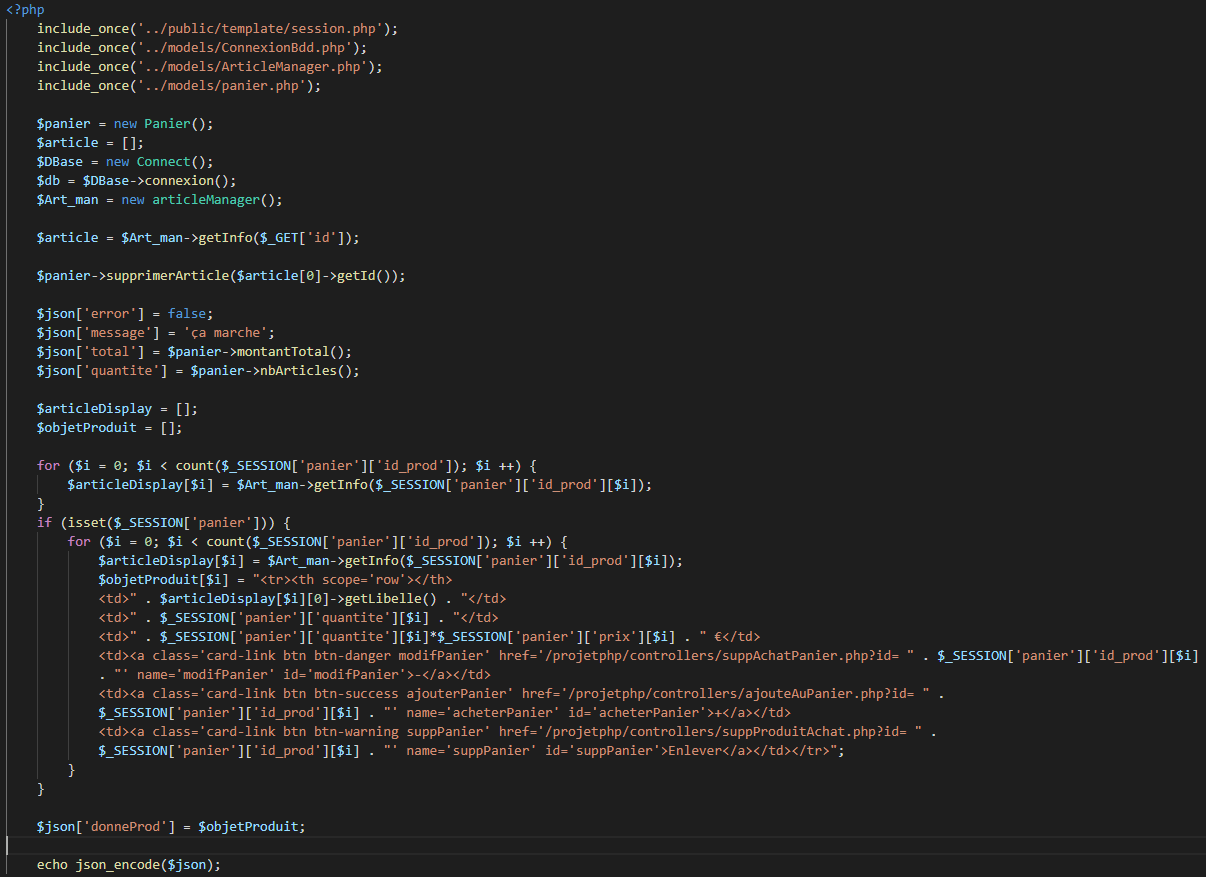


Raccordée à ce fichier qui recharge le panier d’achat avec les bonnes valeurs souhaitées :





Raccordée à ce fichier qui recharge le panier d’achat avec les bonnes valeurs souhaitées :



Ces fichiers sont raccordés dans le panier aux trois boutons précédemment montrés dans une image juste au-dessus de présentation du panier et la fonction ajouter panier est aussi dans le bouton acheter en dessous des produits.

# Réalisation technique :

Les langages utilisés :

HTML5 : Langage de balisage conçu pour représenter les pages web. C’est un langage permettant d’écrire de l’hypertexte, d’où son nom. HTML permet également de structurer sémantiquement et logiquement et de mettre en forme le contenu des pages, d’inclure des ressources multimédias dont des images, des formulaires de saisie et des programmes informatiques. Il permet de créer des documents. Il est souvent utilisé conjointement avec le langage de programmation JavaScript et des feuilles de style en cascade (CSS).

CSS3 : Afin de manipuler la présentation, nous avons utilisé des feuilles de style CSS.

BOOTSTRAP : Bootstrap est un ensemble de fichiers CSS et JavaScript fonctionnant ensemble et qu’on va pouvoir utiliser pour créer des designs complexes de manière relativement simple et rapide de plus cette solution est gratuite donc privilégier. Il nous permet d’économiser un temps précieux lors de la réalisation des interfaces de nos pages.

PHP7 : Pour le côté serveur, nous avons choisi le langage PHP, ceci pour plusieurs raisons :

Tout d’abord, le PHP gère très bien les requêtes SQL. Ce qui est important pour manipuler les résultats de requêtes SQL.

Enfin, le PHP est un des langages coté serveur le plus utilisé, il existe une grande communauté d’utilisateurs de PH, ce qui en cas de problème peut vite être résolue par des tuto sur YouTube ou des forums.

SQL : L'association PHP/MySQL est la plus utilisée. C'est pourquoi nous avons choisi le serveur de base de données MySQL contenue dans l’application 3 en 1 qu’est WAMP qui offre une grande flexibilité. Il intègre de nombreuses fonctionnalités et assure une bonne sécurité des données.

JAVASCRIPT : Javascript permet de créer des sites internet dynamiques. Javascript permet donc d’optimiser au maximum l’expérience de l’utilisateur par l’insertion de script dans nos pages PHP, comme recharger des requêtes pour actualiser nos éléments sans recharger la page.

# Organisation

Nous nous sommes répartie le travail par secteur chacun s’occuper d’un morceau du site telle que le style, la base de données ou encore l’utilisateur. Nous regroupions notre travail grâce a un dêpot GIT.

# Résultat

Les résultats sont atteints sauf pour les commandes, les factures et les paiements.

Les exigences non fonctionnelles qui assurent la qualité et la performance du site ont toutes été développées. L’ergonomie du site est satisfaisante pour les clients.

# Améliorations envisagées

Ajouter un système de gestion des factures

Ajouter les paiements des clients

La sauvegarde des factures

Un système de fidélité