DOSSIER DE PROJET

« VATMINTON »



**Titre professionnel**

**NIVEAU III**

Développeur Web et Web Mobile

**CHIV Vathana**

**LA MANU Soissons – Promotion n°3**

SOMMAIRE

1. Résumé du projet 3
2. Cahier des charges : 1. Descriptif 4

2. outils de développement 5

3. rôle utilisateur et fonctionnalités associées 6

1. Liste des compétences couvertes par le projet : 7
2. Spécifications techniques 8
3. Réalisations (sécurité et web mobile)

Partie Front-end

1. Maquetter une application 9
2. Réaliser une interface utilisateur web

et adaptable 11

1. Développer une interface utilisateur web dynamique 12

Partie Back-end

1. Créer une base de données 15
2. Développer les composants d’accès aux données 20
3. Développer la partie Back-End

d’une application web et web mobile 21

VI. Présentation du CRUD “product” 23

VII. Description de la veille 32

VIII. Description d’une recherche anglophone 35

IX. Remerciements 37

X. Conclusion 37

I. Résumé du projet

Passionné d’informatique et de badminton, mon projet portera sur la vente d’accessoires de badminton en ligne dont son identité sera « Vatminton » caractérisée par le logo d’un badiste en plein smash avec un « C » inversé pour faire référence à la première lettre du mot « compétiteurs».

Mon site sera aussi bien ciblé pour des particuliers, des professionnels que des collectivités. L’objectif sera de le rendre « responsive ». La création du site par un « en-tête» sera composée d’une barre de navigation avec trois catégories (« boutique », « compétition », « entraînement »). J’y intégrerai également un champ de  recherche pour les produits ainsi que deux onglets supplémentaires (« mon compte », « mon panier ») accessible après la connexion de l’utilisateur.

La page d’accueil sera animée par l’actualité hebdomadaire, la meilleure offre promotionnelle du moment, le classement des meilleurs joueurs mondiaux.

Je conclurai que je mettrai au profit du site toutes mes connaissances de langages de programmation étudiés en cours de formation de « LA MANU » tels que HTML, CSS, JavaScript, jQuery, PHP,MySQL.

II. Cahier des charges

1. Descriptif du projet :

Il y a encore quelques années, les articles de sport (vêtements, équipements, …) se vendaient par tradition en point de vente physique ou lors d’événements. Aujourd’hui, l’e-commerce prend de plus en plus d’importance sur le marché des produits sportifs. Certains acteurs du retail sportif vont même jusqu’à vendre exclusivement en ligne !

Cette évolution est principalement due aux nouvelles habitudes de consommation des clients, à la multiplication du nombre d’écrans par foyer (ordinateurs, mobiles, tablettes, …), ainsi qu’aux améliorations logistiques.

Mon site web ciblera aussi bien des particuliers, des professionnels que des collectivités.

1. Leurs outils de développement :

* Lignes de commandes (cd, mkdir, mv, …)
* Trello (outils de gestion de projets)
* Git, Github (outils de programmation)
* Atom, Netbeans (éditeurs de texte)

1. Rôle utilisateur & Fonctionnalités associées :

a) Visiteurs

* Un utilisateur non enregistré sera considéré comme simple visiteur. Il n’aura donc qu’un droit de consultation sur l’ensemble des données (actualités, articles de la boutique, …). S’il désire faire un achat, il devra se connecter en tant que client.

b) Client

* Création d’un compte client :

Un utilisateur est considéré comme client dès lors qu’il dispose d’un compte utilisateur enregistré, ce qui lui permettra d’effectuer ses achats sur le site.

* Se connecter à son compte utilisateur :

A l’aide de son adresse mail et de son mot de passe, l’utilisateur pourra se connecter au site en tant que client.

* Lire et modifier les informations de son profil :

Un client aura la possibilité de voir les différentes informations qu’il aura saisies dans le formulaire d’inscription et de les modifier à tout moment.

* Suppression du compte utilisateur :

Un utilisateur aura la possibilité de supprimer son propre compte.

* Déconnexion :

Un utilisateur pourra se déconnecter à tout moment.

c) Administrateur

* Administration de la liste des produits :  
  L’administrateur a la possibilité de créer, de modifier et de supprimer les produits destinés à être mis en vente.
* Modification des comptes utilisateurs :  
  Il pourra à sa guise modifier le rôle de n’importe quel utilisateur sur le site, c’est à dire le faire passer de simple utilisateur à administrateur afin qu’il puisse bénéficier de toutes les fonctionnalités.  
  Suppression de compte :  
  Il pourra en outre supprimer n’importe quel compte utilisateur.

III. Liste des compétences

|  |  |  |  |
| --- | --- | --- | --- |
| N° Fiche AT | Activités Types | N° Fiche CP | Compétences Professionnelles |
| 1 | Développer la partie Front-End d’une Application Web ou Web Mobile en intégrant les recommandations de sécurité | 1 | Maquetter une application |
| 2 | Réaliser une interface utilisateur web statique et adaptable |
| 3 | Développer une interface utilisateur web dynamique |
| 4 | Réaliser une interface utilisateur avec une solution de gestion de contenu ou E-commerce |
| 2 | Développer la partie Back-End d’une Application Web ou Web Mobile en intégrant les recommandations de sécurité | 5 | Créer une base de données |
| 6 | Développer les composants d’accès aux données |
| 7 | Développer la partie Back-End d’une application web et web mobile |
| 8 | Élaborer et mettre en œuvre des composants dans une application de gestion de contenu ou E-commerce |

IV. Spécifications techniques

Logiciels

|  |  |
| --- | --- |
| HTML 5 | HyperText Markup Language  Langage de base pour la création d’un site internet, il sert à structurer sa page web, c’est-à-dire le contenu, les images, … |
| CSS 3 | Cascading Style Sheets  Langage permettant de mettre en forme sa page web, c’est à dire de la styliser. |
| JavaScript | Langage de programmation de scripts très utilisé pour rendre ses pages web interactives. |
| PHP 7 | HyperText PreProcessor  Langage de programmation utilisé notamment pour faire la liaison entre l’utilisateur et la base de données. |
| MySql | Système de gestion de base des données relationnelles (SGBDR). Permet la réalisation de requêtes pour dialoguer avec la base de données. |

Frameworks

|  |  |
| --- | --- |
| Bootstrap 4 | Framework CSS permettant de faciliter le développement de sites et d’applications réactives et responsives (adaptables à tout moment). |
| JQuery | Framework JavaScript permettant de rendre son site dynamique, gérer des évènements, animations, … |

Outils

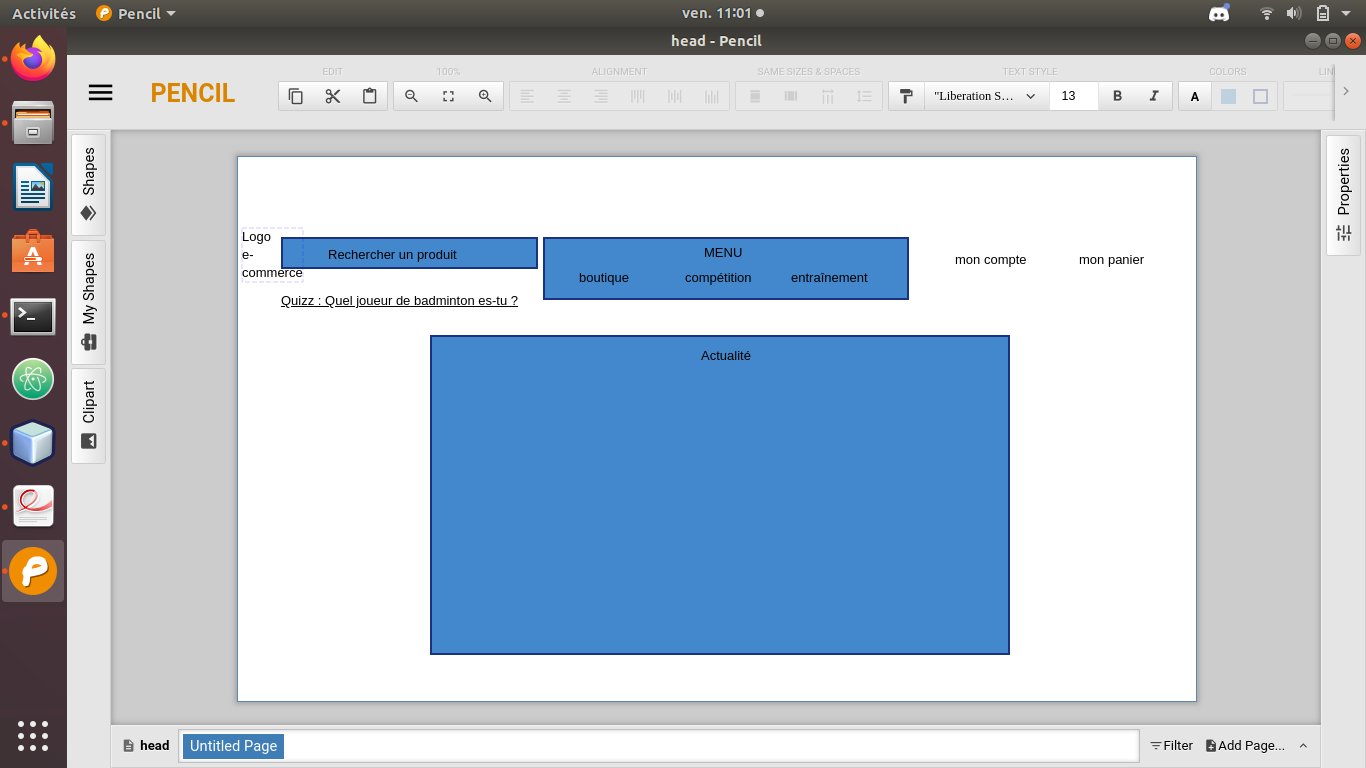
|  |  |
| --- | --- |
| Netbeans | Environnement de développement intégré (IDE). |
| PhpMyAdmin | Système de gestion de base de données (SGBD) pour gérer la base de données ainsi que la visualiser. |
| Jmerise | Logiciel de modélisation permettant de schématiser une base de données, ses différentes relations et de générer son script SQL. |
| Pencil | Logiciel de maquettage. |
| Git | Logiciel de gestion de versions décentralisées, permettant la sauvegarde progressive de son avancée et de récupérer en cas de perte. |

V. Réalisations (sécurité et web mobiles)

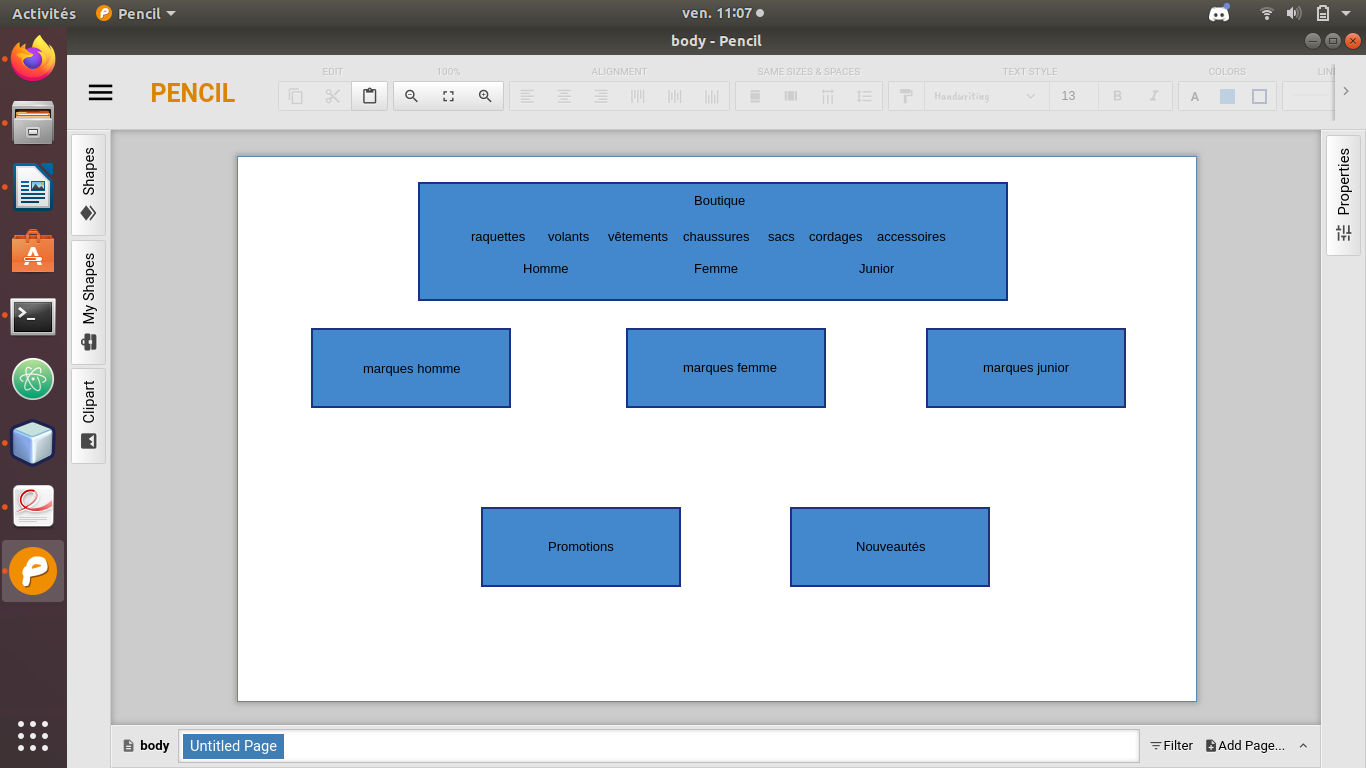
1. Partie Front-end

1. Maquetter une application

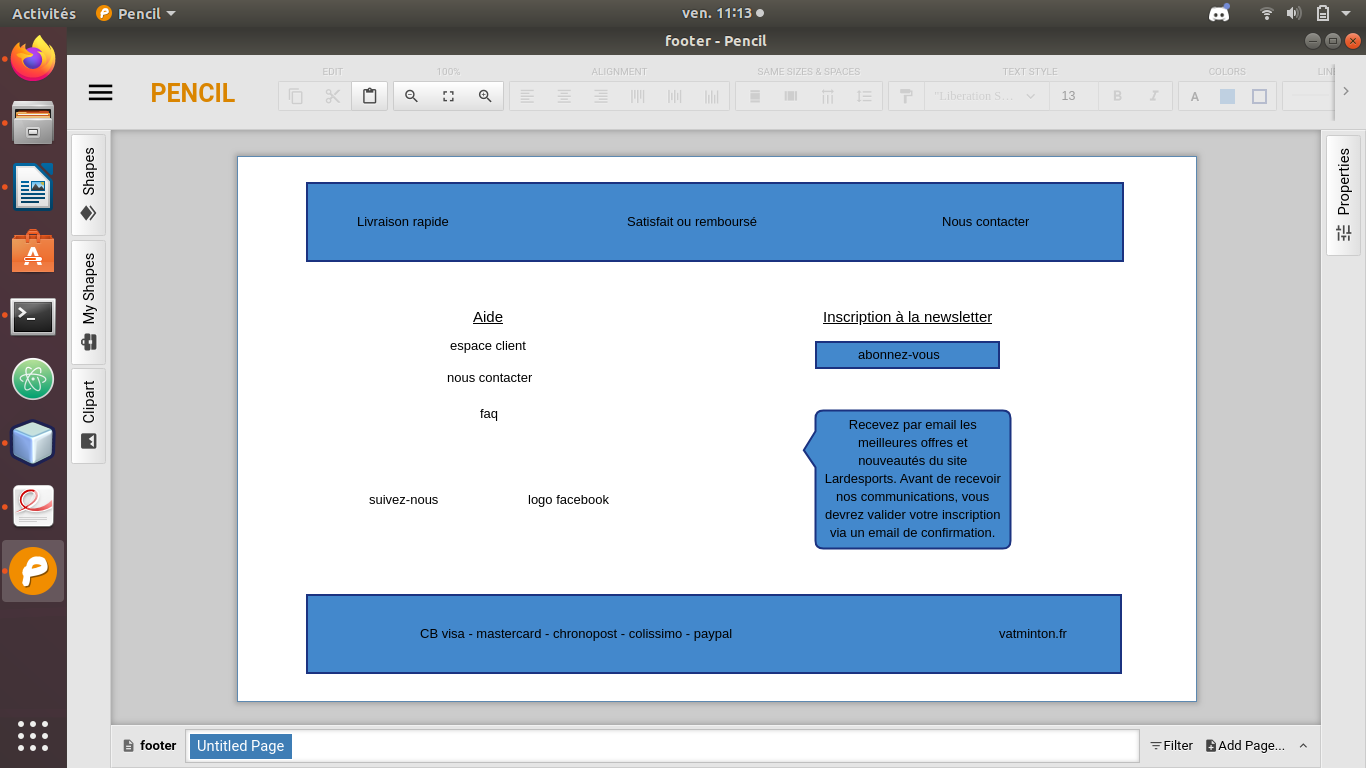
Après une mûre réflexion sur l’aspect visuel de mon site, j’ai réalisé un maquettage de mon projet sous le logiciel Pencil.



Vue de l’en-tête de ma page



Vue du corps de ma page



Vue du pieds de ma page

2. Réaliser une interface utilisateur web statique et adaptable, WEB MOBILE (responsive)

Le responsive Web design est une approche de conception Web qui vise à l’élaboration de sites offrant une expérience de lecture et de navigation optimales pour l’utilisateur quelle que soit sa gamme d’appareil (téléphones mobiles, tablettes, liseuses, moniteurs d’ordinateur de bureau).

Une expérience utilisateur “Responsive” réussie implique un minimum de redimensionnement (zoom), de recadrage, et de défilements multidirectionnels de pages.

Voici un échantillon de mon script :

<?php

<div **class**="row d-flex justify-content-center my-4">

<div **class**="col-lg-12">

<div **class**="news text-center my-3 p-0">Actualité n°1</div>

</div>

</div>

<!-- 1er article -->

<div **class**="row lee">

<div **class**="col-md-6 d-flex justify-content-center py-3">

<img src="../assets/img/leeChongWei.jpg" **class**="wei" alt="" /></div>

<div **class**="col-md-5">

<h1 **class**="display text-center">Atteint d’un cancer, Lee Chong Wei ne participera pas aux Mondiaux</h1>

<p **class**="lead text-center">Atteint d’un cancer depuis l’été dernier, la star du badminton Lee Chong Wei ne participera pas aux Mondiaux en Suisse au mois

…

<p **class**="lead mt-2">Tombé à la 113e place mondiale, Lee Chong Wei a repris l’entraînement au mois de janvier, mais a repoussé plusieurs fois depuis son retour à la compétition.</p>

</div>

</div>

?>

Quelques explications :

on retrouvera le plus fréquemment des règles pour :

* agrandir la taille du texte
* agrandir la taille des contrôles et zones cliquables (pour une utilisation au doigt)
* faire passer le contenu sur une seule colonne
* masquer ou afficher des éléments spécifiques
* ajuster les dimensions et marges

Voici une partie de mon css :

<?php

@media only screen and (max-width: 600px) {

img {

width:100%;

display: block;

float: none;

margin-left: auto;

margin-right: auto;

}

body {

width:100%;

background-color: #FFFFFF;

display: block;

float: none;

margin-left: auto;

margin-right: auto;

}

#carousel{

width: 85%;

display: block;

float: none;

margin-left: auto;

margin-right: auto;

}

}

?>

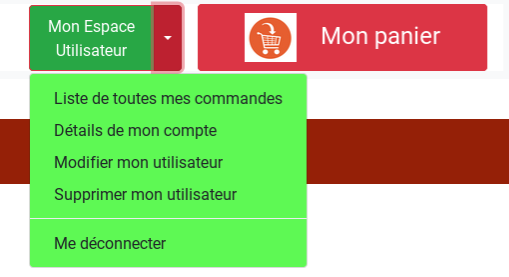
3. Développer une interface utilisateur web dynamique

La barre de navigation est un élément essentiel du site puisqu’elle est présente sur chaque page.

Le contenu de ma page web dynamique peut donc varier en fonction d'informations (role utilisateur, entreprise ou administrateur, panier, formulaire rempli par l'utilisateur, etc.) qui ne sont connues qu'au moment de sa consultation. À l'inverse, le contenu d'une page web statique est *a priori* identique à chaque consultation.

Les 3 “navBar” différentes de mon site :

**Barre de navigation de l’utilisateur**



**Barre de navigation du fournisseur**



**Mode administrateur avec tous les droits et ses contenus**





Explications : J’ai créé un fichier “header.php” et “footer.php” afin de pouvoir ensuite l’inclure dans mes pages, ce qui évite de remettre le même code (allègement de la taille de mes fichiers). Pour l’intégrer, j’utilise la fonction PHP “include”.

Comme les images ci-dessus, pour que mon application soit fonctionnelle, j’ai codé de telle sorte que chaque “role” à des fonctions différentes dès leur connexion au site.

Les 3 rôles sont :

* Administrateur
* Utilisateur
* Fournisseur

Un utilisateur par exemple ne pourra pas *ajouter un produit* ou un fournisseur ne pourra pas *ajouter dans son panier*.

Partie Back-end

4. Créer une base de données

Pour réaliser mon projet, j’ai réfléchi à la structure de ma base de données, aux tables et aux champs dont j’aurais besoin.

MCD : Modèle conceptuel de Données

Pour concevoir ma base de données, j’ai utilisé l’outil de modélisation JMerise.

J’ai tout d’abord créé les différentes tables et ajouté les relations entre elles.

Ma base de données comporte 8 tables, auxquelles j’ai ajouté un préfixe pour augmenter la sécurisation des données.

En effet, sachant que ma base de données va contenir la majeure partie des informations du site, ces failles peuvent avoir pour but de :

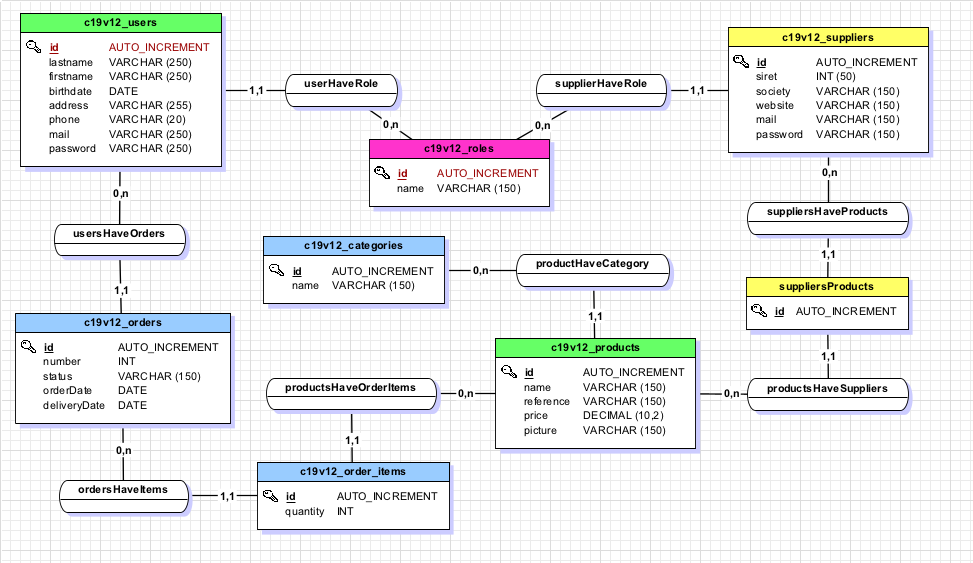
* Détruire le contenu du site
* Faire “planter” le site
* Ajouter des commentaires de spams
* Extraire des informations (ex : récupérer des adresses mails)
* Ajouter du contenu sur le site (ex : liens vers des sites illégaux)

Le site sera composé des tables suivantes :

* “c19v12\_users” stocke les informations des utilisateurs
* “c19v12\_roles” contient les différents rôles qui peuvent être attribués
* “c19v12\_suppliers” stocke les informations des fournisseurs
* “c19v12\_products” stocke les informations des produits
* “c19v12\_suppliersProducts” stocke les informations des produits des fournisseurs
* “c19v12\_catégories” stocke les informations du nom de catégorie
* “c19v12\_orders” stocke les informations de commande
* “c19v12\_order\_items” stocke les quantités de produits

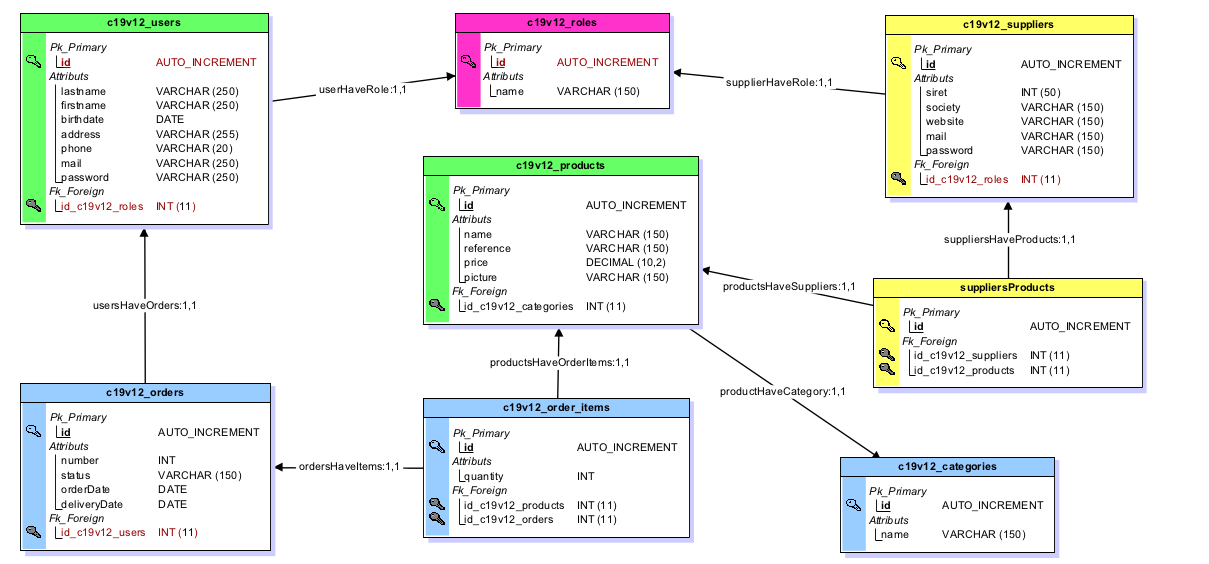
Mes tables auront des relations entre elles :

* Un utilisateur, comme le fournisseur et l’administrateur aura obligatoirement un rôle “role”
* Un utilisateur ne pourra pas ajouter, modifier ou supprimer un produit
* Un fournisseur n’aura pas accès au panier de la boutique pour y faire ces achats car l’espace est dédié à l’utilisateur
* Un produit appartient à une catégorie
* Un utilisateur aura accès au panier avec toutes les commandes en cours. Il pourra d’ailleurs accéder à toutes les coordonnées le concernant et les modifier, ou supprimer
* Un fournisseur aura un accès dédié pour ajouter un produit. Il pourra d’ailleurs accéder à toutes les informations de son entreprise et les modifier, ou supprimer
* Un utilisateur connecté pourra faire des commandes de vente en ligne
* Un panier peut avoir une ou plusieurs commandes du même utilisateur
* Un utilisateur pourra passer autant de commande qu’il le souhaite
* Un fournisseur pourra ajouter autant de produit qu’il le souhaite



MLD : Modèle logique de Données

Grâce à ces relations, il est possible de générer le MLD sur lequel vont apparaître les clés étrangères, permettant de lier les tables entre elles.



A partir de JMerise, j’ai généré un script SQL pour créer ma base de données que j’ai intégré à partir de la **console de linux** et par la suite avec **PhpMyAdmin**.

<?php

#------------------------------------------------------------

# Script MySQL.

#------------------------------------------------------------

#------------------------------------------------------------

# Table: c19v12\_roles

#------------------------------------------------------------

CREATE TABLE c19v12\_roles(

id Int Auto\_increment NOT NULL ,

name Varchar (150) NOT NULL

,CONSTRAINT c19v12\_roles\_PK PRIMARY KEY (id)

)ENGINE=InnoDB;

#------------------------------------------------------------

# Table: c19v12\_suppliers

#------------------------------------------------------------

CREATE TABLE c19v12\_suppliers(

id Int Auto\_increment NOT NULL ,

siret Int NOT NULL ,

society Varchar (150) NOT NULL ,

website Varchar (150) ,

mail Varchar (150) NOT NULL ,

password Varchar (150) NOT NULL ,

id\_c19v12\_roles Int NOT NULL

,CONSTRAINT c19v12\_suppliers\_PK PRIMARY KEY (id)

,CONSTRAINT c19v12\_suppliers\_c19v12\_roles\_FK FOREIGN KEY (id\_c19v12\_roles) REFERENCES c19v12\_roles(id)

)ENGINE=InnoDB;

#------------------------------------------------------------

# Table: c19v12\_categories

#------------------------------------------------------------

CREATE TABLE c19v12\_categories(

id Int Auto\_increment NOT NULL ,

name Varchar (150) NOT NULL

,CONSTRAINT c19v12\_categories\_PK PRIMARY KEY (id)

)ENGINE=InnoDB;

#------------------------------------------------------------

# Table: c19v12\_products

#------------------------------------------------------------

CREATE TABLE c19v12\_products(

id Int Auto\_increment NOT NULL ,

name Varchar (150) NOT NULL ,

reference Varchar (150) NOT NULL ,

price Decimal (10,2) NOT NULL ,

picture Varchar (150) ,

id\_c19v12\_categories Int NOT NULL

,CONSTRAINT c19v12\_products\_PK PRIMARY KEY (id)

,CONSTRAINT c19v12\_products\_c19v12\_categories\_FK FOREIGN KEY (id\_c19v12\_categories) REFERENCES c19v12\_categories(id)

)ENGINE=InnoDB;

#------------------------------------------------------------

# Table: c19v12\_users

#------------------------------------------------------------

CREATE TABLE c19v12\_users(

id Int Auto\_increment NOT NULL ,

lastname Varchar (250) NOT NULL ,

firstname Varchar (250) NOT NULL ,

birthdate Date NOT NULL ,

address Varchar (255) NOT NULL ,

phone Varchar (20) NOT NULL ,

mail Varchar (250) NOT NULL ,

password Varchar (250) NOT NULL ,

id\_c19v12\_roles Int NOT NULL

,CONSTRAINT c19v12\_users\_PK PRIMARY KEY (id)

,CONSTRAINT c19v12\_users\_c19v12\_roles\_FK FOREIGN KEY (id\_c19v12\_roles) REFERENCES c19v12\_roles(id)

)ENGINE=InnoDB;

#------------------------------------------------------------

# Table: c19v12\_orders

#------------------------------------------------------------

CREATE TABLE c19v12\_orders(

id Int Auto\_increment NOT NULL ,

number Int NOT NULL ,

status Varchar (150) NOT NULL ,

orderDate Date NOT NULL ,

deliveryDate Date NOT NULL ,

id\_c19v12\_users Int NOT NULL

,CONSTRAINT c19v12\_orders\_PK PRIMARY KEY (id)

,CONSTRAINT c19v12\_orders\_c19v12\_users\_FK FOREIGN KEY (id\_c19v12\_users) REFERENCES c19v12\_users(id)

)ENGINE=InnoDB;

#------------------------------------------------------------

# Table: suppliersProducts

#------------------------------------------------------------

CREATE TABLE suppliersProducts(

id Int Auto\_increment NOT NULL ,

id\_c19v12\_suppliers Int NOT NULL ,

id\_c19v12\_products Int NOT NULL

,CONSTRAINT suppliersProducts\_PK PRIMARY KEY (id)

,CONSTRAINT suppliersProducts\_c19v12\_suppliers\_FK FOREIGN KEY (id\_c19v12\_suppliers) REFERENCES c19v12\_suppliers(id)

,CONSTRAINT suppliersProducts\_c19v12\_products0\_FK FOREIGN KEY (id\_c19v12\_products) REFERENCES c19v12\_products(id)

)ENGINE=InnoDB;

#------------------------------------------------------------

# Table: c19v12\_order\_items

#------------------------------------------------------------

CREATE TABLE c19v12\_order\_items(

id Int Auto\_increment NOT NULL ,

quantity Int NOT NULL ,

id\_c19v12\_products Int NOT NULL ,

id\_c19v12\_orders Int NOT NULL

,CONSTRAINT c19v12\_order\_items\_PK PRIMARY KEY (id)

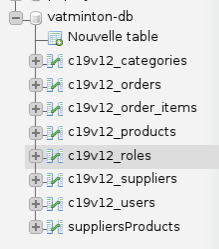
,CONSTRAINT c19v12\_order\_items\_c19v12\_products\_FK FOREIGN KEY (id\_c19v12\_products) REFERENCES c19v12\_products(id)

,CONSTRAINT c19v12\_order\_items\_c19v12\_orders0\_FK FOREIGN KEY (id\_c19v12\_orders) REFERENCES c19v12\_orders(id)

)ENGINE=InnoDB;

?>

Voici la structure de ma base de données dans phpMyAdmin :



5. Développer les composants d’accès aux données

Pour une question pratique et économiser la taille des fichiers, j’ai placé mon en-tête et mon pied de page dans 2 fichiers distincts : “header.php” et “footer.php”. Il me suffit ensuite d’inclure ces 2 fichiers dans chacune de mes pages.

voici mon code :

<?php

session\_start();

**require\_once** '../init/credentials.php';

**require\_once** '../models/database.php';

**...**

**include** 'header.php';

?>

…

**<?php include** 'footer.php'; ?>

Pour accéder aux données de ma base de données, je crée la méthode **“getInstance()”** dans mon modèle “database”.

La connexion à ma base de données s’effectue de la façon suivante :

<?php

**class** Database {

// Objet de connexion PDO

**public** $db;

// Objet auto-instancié utilisé avec les méthodes de transaction SQL

**private** **static** $instance;

**private** $host = DB\_HOST;

**private** $database = DB\_NAME;

**private** $username = DB\_USER;

**private** $password = DB\_PWD;

/\*\*

\* Initialise une connection à la base de données

\*\*/

**public** **function** \_\_construct() {

**try** {

// -- IMPORTANT --

// Les tables doivent être du type innoDB pour prendre en charge les transactions.

// Les tables de type MyISAM ne gèrent pas les transactions.

$**this**->db = **new** **PDO**('mysql:host=' . $**this**->host .';dbname=' . $**this**->database . ';charset=utf8',

$**this**->username,

$**this**->password);

// En mode ERRMODE\_EXCEPTION, si un échec survient, le script est

// interrompu, la connexion fermée, et mysql effectue un roll back

// sur la transaction

$**this**->db->setAttribute( **PDO**::ATTR\_ERRMODE, **PDO**::ERRMODE\_EXCEPTION );

} **catch** (**Exception** $e) {

**die**('Error : ' . $e->getMessage());

}

}

/\*\*

\* Méthode retournant une instance de classe PDO

\* @return object

\*/

**public** **static** **function** getInstance() {

**if** ( is\_null( **self**::$instance ) ) {

**self**::$instance = **NEW** Database();

}

**return** **self**::$instance;

}

/\*\*

\* Fermeture automatique de la connexion lorsqu'une instance de classe est détruite

\*/

**public** **function** \_\_destruct() {

$**this**->db = NULL;

}

}

?>

Explications : La méthode getInstance permet de récupérer l’instance de connexion à ma base de données. Elle permet de créer une instance unique de mon objet.

6. Développer la partie Back-end d’une application web et web mobile

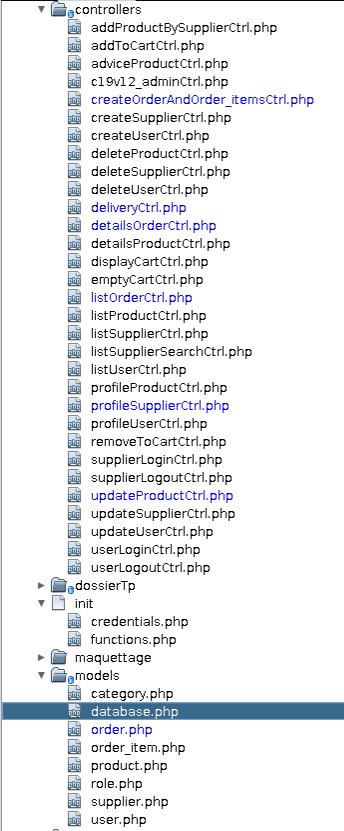
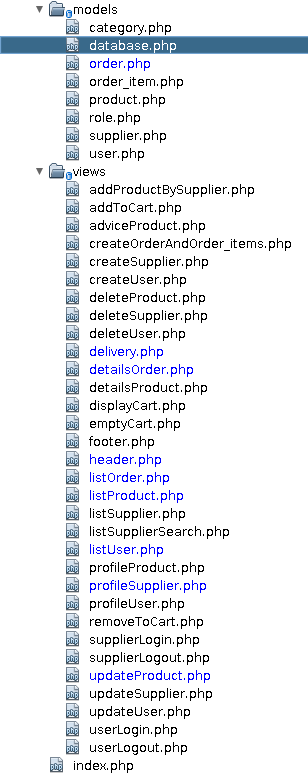
Pour développer la partie Back-end de mon application web, j’ai choisi de structurer mon code avec l’architecture (MVC Modèle Vue Contrôleur) de PHP.

-> Un modèle (Model) qui contient les requêtes qui vont me permettre de récupérer les données à afficher (en PHP)

-> Une vue (view) qui correspond à l’interface graphique (en HTML.CSS)

-> Un contrôleur (Controller) qui permet de faire dialoguer la vue et le modèle (en PHP)

Voici l’architecture de mes dossiers dans netbeans :



VI. Présentation CRUD “product”

Qu’est ce que le CRUD ?

L'acronyme informatique anglais **CRUD** (pour *create*, *read*, *update*, *delete*) (parfois appelé SCRUD avec un "S" pour *search*) désigne les quatre opérations de base pour la persistance des données, en particulier le stockage d'informations en base de données.

Soit :

* **c**reate : créer
* **r**ead : lire
* **u**pdate : mettre à jour
* **d**elete : supprimer

Plus généralement, il désigne les opérations permettant la gestion d'une collection d'éléments.

Ce terme est aussi un jeu de mot en anglais sur l'adjectif **crude** (en français **brut** ou **rudimentaire**).

1. **C**reate : product

1.La première étape consiste à créer une table de base de données qui sera manipulée en utilisant l’accès CRUD. Pour cela, il faut soumettre la requête suivante dans votre base de données MySQL :  
  
<?php  
**public** **function** addProductBySupplier() {  
 //Soumission de la requête au serveur de la base de données  
 $results = $**this**->pdo->db->prepare('INSERT INTO `c19v12\_products` (`name`, `reference`, `price`, `picture`, `id\_c19v12\_categories`)  
 VALUES (:name, :reference, :price, :picture, :id\_c19v12\_categories)');  
 // Associe une valeur à chaque marqueur nominatif de la requête $results->bindValue(':name', $**this**->name, **PDO**::PARAM\_STR); $results->bindValue(':reference', $**this**->reference, **PDO**::PARAM\_STR); $results->bindValue(':price', $**this**->price, **PDO**::PARAM\_STR); $results->bindValue(':picture', $**this**->picture, **PDO**::PARAM\_STR); $results->bindValue(':id\_c19v12\_categories', $**this**->id\_c19v12\_categories, **PDO**::PARAM\_INT);  
// Exécution de la requête  
**try** {  
**return** $results->execute(); // Méthode execute() toujours avec prepare() sinon query()  
}   
**catch** (**Exception** $e) {  
**die**('échec de la connexion :' . $e->getMessage());  
}  
}  
?>

La table est utilisée pour collecter des informations sur l’utilisateur : le nom, la référence, le prix, l’imagetable. Chaque entrée reçoit automatiquement une **clé primaire** (*AUTO\_INCREMENT PRIMARY KEY*), c’est-à-dire son propre ID unique.  
  
2. Il est ensuite nécessaire de configurer la connexion et de supprimer la base de données. Il faut créer un fichier PHP sous le nom database.php, il est ensuite nécessaire d’insérer le script suivant avec la classe « Database » pour gérer les connexions de base de données :  
  
« Database » pour gérer les connexions de base de données :

<?php

**class** Product **extends** Database {

**public** $pdo;

//attributs

//(seront utilisés lorsque l'on récupèrera des données à partir du formulaire)

**public** $id;

**public** $name;

**public** $reference;

**public** $price;

**public** $picture;

**public** $id\_c19v12\_categories;

**public** $formErrors = **array**();

**public** **function** \_\_construct() {

$**this**->pdo = Database::getInstance(); // Création de l'instance de connexion

}

**public** **function** \_\_destruct() {

$**this**->pdo->db = NULL;

}

/\*

\* Méthode démarrant une transaction SQL

\*/

**public** **function** beginTransaction() {

**return** $**this**->pdo->db->beginTransaction();

}

/\* \* Méthode appliquant la transaction courante et rendant ses modifications permanentes \*/  
**public** **function** commit() {

**return** $**this**->pdo->db->commit();

}

/\*

\* Méthode faisant un rollback de la transaction courante et annulant ses modifications

\*/

**public** **function** rollBack() {

**return** $**this**->pdo->db->rollBack();

}

/\*

\* Method retournant l'id de la dernière rangée insérée en base de données

\*/

**public** **function** lastInsertId() {

**return** $**this**->pdo->db->lastInsertId();

}

?>

Afin de pouvoir utiliser ici la classe définie pour l’accès à la base de données avec PDO, il faut **spécifier exactement les valeurs pour les quatre entrées $database** (nom de la base de donnée utilisée), **$host** (nom de l’hôte sur lequel la base de données est en cours d’exécution), **$username** (nom d’utilisateur) et **$password** (mot de passe de l’utilisateur).

Dans ce script, la classe de base de données comporte trois fonctions : *\_\_construct()*, le **constructeur de classe** permet que la classe rappelle aux utilisateurs que l’initialisation (c’est-à-dire l’assignation d’une valeur initiale ou de départ) n’est pas autorisée. Ensuite *connect()* est la fonction principale de la classe qui régule la connexion enfin au contraire *disconnect()* est utilisée pour mettre fin à la connexion.

<?php  
// Création d'une instance de classe Product. Instance de classe = OBJET $product = **new** Product();  
// Récupération des données du formulaire  
// Affectation de chaque champ de formulaire à l'attribut auquel il est associé  
$product->name = **isset**($\_POST['name']) ? htmlspecialchars($\_POST['name']) : '';  
$product->price = **isset**($\_POST['price']) ? htmlspecialchars($\_POST['price']) : '';  
$product->reference = **isset**($\_POST['reference']) ? htmlspecialchars($\_POST['reference']) : ''; $product->id\_c19v12\_categories = **isset**($\_POST['category']) && $\_POST['category'] > 0 ? $\_POST['category'] : ''; // input name = category  
//Toutes regex pour le formulaire  
$regexNameProduct = '/^[a-zA-Z0-9À-ÿ\' -]+$/';  
$regexReference = '/^[A-Z]{1,5}[0-9]{1,5}$/';  
$regexPrice = '/^\d+(\d{3})\*(\.\d{1,2})?$/';  
…  
// Validation de picture  
**if** (**isset**($\_FILES['picture']) && $\_FILES['picture']['error'] == 0) { $filesInfo = basename($\_FILES['picture']['name']);  
$extension\_upload = $filesInfo['extension'];  
$extension\_allowed = **array**('jpg','png','bmp');  
$product->picture = $filesInfo;  
**if** (in\_array($extension\_upload, $extension\_allowed)) {  
**echo** 'Le fichier transmis est bien un fichier jpg';  
} **else** {  
**echo** '/!\ vous devez sélectionner un fichier JPG /!\ ';  
}  
}  
// Insertion du product dans la base de données  
**if** (**empty**($product->formErrors)) {  
$success = $product->addProductBySupplier();  
?>

La page *createUser.php* est désormais activée, permettant ainsi d’entrer les informations utilisateur. Le script garantit que toutes les **données entrées et les messages d’erreur** sont enregistrés et que les messages d’erreur correspondants apparaissent bien quand une **entrée est incorrecte**, enfin le script assure aussi la **transmission des données à la base de données.**

1. **R**ead : product

La procédure Read lit les enregistrements d’une table sur la base de clé primaire indiqué en paramètre d’entrée. Mes 2 fichiers permettant de lire dans mon CRUD User les enregistrements d’une table sont “listUser.php” et “profilUser.php”.  
  
Ma vue “listProduct.php” permet d’afficher la liste des produits. Mon code :  
  
<?php

session\_start();

**require\_once** '../init/functions.php';

**require\_once** '../init/credentials.php';

**require\_once** '../models/database.php';

**require\_once** '../models/product.php';

**require\_once** '../models/category.php';

**require\_once** '../controllers/listProductCtrl.php';

//require\_once '../controllers/listProductSearchCtrl.php';

**require\_once** 'header.php';

?>

<**div** id="detailsProduct" class="h1 text-center mt-4">LISTE DES PRODUITS

</**div**>

<**div** class="row d-flex justify-content-center mt-5">

<**input** type="hidden" name="id" value="<?= **isset**($\_GET['id']) ? $\_GET['id'] : '' ?>" />

<?php **foreach** ($productList **as** $product) { ?>

<**div** class="col-md-4 d-flex justify-content-center mt-3">

<**div** class="card text-black text-center bg-secondary mb-3" style="**max-width**: 20rem;">

<**div** class="card-header"><?= $product->id ?><?= $product->productName ?></**div**>

<**div** class="card-body">

<**div**><**img** src="../assets/img/<?= $product->picture ?>" class="picture" alt="vatminton" /></**div**>

<**h4** class="card-title">référence : <?= $product->reference ?></**h4**>

<**p** class="card">Catégorie de produit : <?= $product->categoryName ?></**p**>

<**p** class="card">Prix : <?= $product->price ?> €</**p**>

<**a** class="px-3" href="detailsProduct.php?id=<?= $product->id ?>">Détails</**a**>

<?php **if** (isUser() || isAdmin()) { ?>

<**a** class="px-3" href="addToCart.php?id=<?= $product->id ?>">Ajouter à mon panier</**a**>

<?php } ?>

</**div**>

</**div**>

</**div**>

<?php }

?>

</**div**>

Mon contrôleur “listProductCtrl.php” permet de faire toutes les vérifications nécessaires avant d’envoyer les données à la vue. Mon code :

<?php  
$product = **new** Product();  
$productList = $product->getProductList();  
?>

1. **U**pdate : product

La procédure Update utilise la commande sql UPDATE sur la table en se basant bien sûr la clé primaire spécifié dans la clause WHERE. Tout comme la procédure Create il possède un paramètre pour chaque colonne dans la table. La requête *“prepare” évite l’injection de code SQL.*Voici mon script permettant de modifier le profil de l’utilisateur et de le mettre à jour dans la base de données “product.php” :  
  
<?php

**public** **function** updateProfilProduct() {

//Soumission de la requête au serveur de la base de données

$results = $**this**->pdo->db->prepare('UPDATE `c19v12\_products` SET `name` = :name, `reference` = :reference, `price` = :price, `picture` = :picture

WHERE `id` = :id');

// :mail est un marqueur nominatif, pour éviter l'injection de code SQL

// Associe une valeur à chaque marqueur nominatif de la requête

$results->bindValue(':id', $**this**->id, **PDO**::PARAM\_INT);

$results->bindValue(':name', $**this**->name, **PDO**::PARAM\_STR); //dans On nomme ce qu'on veut dans bindvalue

$results->bindValue(':reference', $**this**->reference, **PDO**::PARAM\_STR);

$results->bindValue(':price', $**this**->price, **PDO**::PARAM\_STR);

$results->bindValue(':picture', $**this**->picture, **PDO**::PARAM\_STR);

$results->bindValue(':name', $**this**->name, **PDO**::PARAM\_STR);

// Exécution de la requête

**try** {

**return** $results->execute(); // Méthode execute() toujours avec prepare() sinon query()

// execute retourne true ou false

} **catch** (**Exception** $e) {

**die**('échec de la connexion :' . $e->getMessage());

}

}

?>

Ma vue “update.php” permet de modifier un produit. Mon code :

<?php

session\_start();

**require\_once** '../init/functions.php';

**require\_once** '../init/credentials.php';

**require\_once** '../models/database.php';

**require\_once** '../models/product.php';

**require\_once** '../models/category.php';

**require\_once** '../models/role.php';

**require\_once** '../controllers/updateProductCtrl.php';

**include** 'header.php';

?>

<**div** id="formulary" class="h1 text-center my-4">Espace fournisseur</**div**>

<**h2** class="text-center mb-4">Modifier un produit</**h2**>

<**div** class="container">

<**div** class="row">

<**div** id="mainForm" class="col-md-12">

<**div** class="row justify-content-center">

<**div** class="col-md-12">

<**div** class="formA p-4 mt-2">

<**form** method="POST" action="" enctype="multipart/form-data">

<!-- Nom -->

<**div** class="form-row">

<**div** class="form-group col-md-6 text-center">

<**label** for="name">Nom : </**label**>  
 <**input** type="text" class="form-control text-center text-info" name="name" placeholder="" id="name" value="<?= **isset**($detailsProduct->productName) ? $detailsProduct->productName : '' ?>" />

<**small** class="text-warning"><?= **isset**($profilProduct->formErrors['name']) ? $profilProduct->formErrors['name'] : '' ?></**small**>

</**div**>

<!-- Price -->

<**div** class="form-group col-md-6 text-center">

<**label** for="price">Prix : </**label**>

<**input** type="text" class="form-control text-center text-info" name="price" id="price" placeholder="" value="<?= **isset**($detailsProduct->price) ? $detailsProduct->price : '' ?>" />

<**small** class="text-warning"><?= **isset**($profilProduct->formErrors['price']) ? $profilProduct->formErrors['price'] : '' ?></**small**>

</**div**>

<!-- Reference -->

<**div** class="form-group col-md-8 text-center">

<**label** for="reference">Référence du produit : </**label**>

<**input** type="text" class="form-control text-center text-info" name="reference" id="reference" placeholder="" value="<?= **isset**($detailsProduct->reference) ? $detailsProduct->reference : '' ?>">

<**small** class="text-warning"><?= **isset**($profilProduct->formErrors['reference']) ? $profilProduct->formErrors['reference'] : '' ?></**small**>

</**div**>

<!-- Categories -->

<**div** class="form-group col-md-4 text-center">

<**label** for="categories">Catégorie : </**label**>

<**select** id="categories" name="category" class="form-control"> <!-- le name="category" est lié à $category dans le controller -->

<**option** value="<?= $detailsProduct->id\_c19v12\_categories ?>" selected><?= $detailsProduct->categoryName ?></**option**>

<?php **foreach** ($categoryList **as** $item) { ?>

<**option** class="text-white" value="<?= $item->id ?>"><?= ucfirst($item->name) ?></**option**> <!-- uppercase first letter -->

<?php } ?>

</**select**>

<**small** class="text-warning"><?= **isset**($profilProduct->formErrors['categories']) ? $profilProduct->formErrors['categories'] : '' ?></**small**>

</**div**>

<**div** class="row">

<**div** class="form-group col-md-12 text-center">

<**input** type="file" name="picture" />

</**div**>

<**input** type="submit" name="submit" value="Enregistrer" />

</**div**>

</**div**>

</**form**>

</**div**>

</**div**>

</**div**>

</**div**>

</**div**>

Mon contrôleur “updateProductCtrl.php” permet de faire toutes les vérifications nécessaires avant d’envoyer les données à la vue. Mon code :

<?php

$product = **new** Product();

$category = **new** Category();

$product->id = **isset**($\_GET['id']) ? $\_GET['id'] : '';

$detailsProduct = $product->detailsProduct();

$categoryList = $category->getCategories();

**if** (**isset**($\_POST['submit'])) {

// Récupération des données du formulaire

// Affectation de chaque champ de formulaire à l'attribut auquel il est associé

$product->name = **isset**($\_POST['name']) ? htmlspecialchars($\_POST['name']) : '';

$product->price = **isset**($\_POST['price']) ? htmlspecialchars($\_POST['price']) : '';

$product->reference = **isset**($\_POST['reference']) ? htmlspecialchars($\_POST['reference']) : '';

$product->id\_c19v12\_categories = **isset**($\_POST['category']) && $\_POST['category'] > 0 ? $\_POST['category'] : ''; // input name = category

//Toutes regex pour le formulaire

$regexNameProduct = '/^[a-zA-Z0-9À-ÿ\' -]+$/';

$regexReference = '/^[A-Z]{1,5}[0-9]{1,5}$/';

$regexPrice = '/^\d+(\d{3})\*(\.\d{1,2})?$/';

// Validation du nom du produit

**if** (**empty**($product->name)) {

$product->formErrors['name'] = 'Ce champ est vide';

} **elseif** (!preg\_match($regexNameProduct, $product->name)) {

$product->formErrors['name'] = 'Merci de rentrer un nom de produit valide';

} **elseif** (strlen($product->name) < 1 || strlen($product->name) > 26) {

$product->formErrors['name'] = 'Le nom doit comporter entre 2 et 25 caractères'; }

// Validation des catégories (select)

**if** (**isset**($\_POST['categories'])) {

**if** (**empty**($product->categories)) {

$product->formErrors['categories'] = 'Ce champ est vide';

} **else** {

$product->formErrors['categories'] = 'champ est vide';

}

}

// Validation du prix

**if** (**empty**($product->price)) {

$product->formErrors['price'] = 'Ce champ est vide';

} **elseif** (!preg\_match($regexPrice, $product->price)) {

$product->formErrors['price'] = 'Merci de rentrer un prix valide';

}

// Validation de la référence

**if** (**empty**($product->reference)) {

$product->formErrors['reference'] = 'Ce champ est vide';

} **elseif** (!preg\_match($regexReference, $product->reference)) {

$product->formErrors['reference'] = 'Merci de rentrer un numéro de référence valide';

}

// Validation de picture

**if** (**isset**($\_FILES['picture']) && $\_FILES['picture']['error'] == 0) {

$filesInfo = basename($\_FILES['picture']['name']);

$extension\_upload = $filesInfo['extension'];

$extension\_allowed = **array**('jpg','png','bmp');

$product->picture = $filesInfo;

**if** (in\_array($extension\_upload, $extension\_allowed)) {

**echo** 'Le fichier transmis est bien un fichier jpg, png, bmp';

} **else** {

**echo** '/!\ vous devez sélectionner un fichier JPG, PNG, BMP /!\ ';

}

}

// Insertion du product dans la base de données

**if** (**empty**($profilProduct->formErrors)) {

$success = $product->updateProfilProduct();

header('Location: ../../views/listProduct.php');

//header('Location: ../../views/detailsProduct.php?id=' . $product->id);

**exit**();

}

}

?>

1. **D**elete : user

Cette procédure supprime une ligne spécifiée dans la clause WHERE.

Mon code dans le modèle product.php :

<?php

**public** **function** deleteProduct() {

//Définition de la requête SQL

$results = $**this**->pdo->db->prepare('DELETE FROM `c19v12\_products` WHERE `id` = :id');

// :mail est un marqueur nominatif, pour éviter l'injection de code SQL

// Associe une valeur à chaque marqueur nominatif de la requête

$results->bindValue(':id', $**this**->id, **PDO**::PARAM\_INT);

//Soumission de la requête au serveur de la base de données

**try** {

**return** $results->execute();

} **catch** (**Exception** $e) {

**die**('échec de la connexion :' . $e->getMessage());

}

}

?>

Ma vue “delete.php” permet de modifier un produit. Mon code :

<?php

session\_start();

**require\_once** '../init/functions.php';

**require\_once** '../init/credentials.php';

**require\_once** '../models/database.php';

**require\_once** '../models/product.php';

**require\_once** '../controllers/deleteProductCtrl.php';

**require\_once** '../views/header.php';

?>

<**div** class="row">

<**div** class="col-md-12 text-center my-5">Êtes-vous sûr de vouloir supprimer le produit ?</**div**>

</**div**>

<**div** class="row">

<**div** class="col-md-6 text-center">

<**a** class="btn bg-success mx-5" href="../views/detailsProduct.php">Annuler</**a**>

</**div**>

<**div** class="col-md-6 text-center">

<**a** href="deleteProduct.php?id=<?= $product->id ?>&confirm=true" class="btn bg-danger mx-5">Confirmer</**a**>

</**div**>

</**div**>

Mon contrôleur “deleteProductCtrl.php” permet de faire toutes les vérifications nécessaires avant d’envoyer les données à la vue. Mon code :

<?php

**if** (**isset**($\_GET['id']) && $\_GET['id'] > 0 && $\_GET['confirm'] == true) {

$product = **new** Product();

$product->id = **isset**($\_GET['id']) ? htmlspecialchars($\_GET['id']) : '';

$success = $product->deleteProduct();

**if** ($success) {

$\_SESSION['message'] = 'Le produit a bien été supprimé';

} **else** {

$\_SESSION['message'] = 'Le produit n\'a pas été supprimé';

}

header('Location: ../views/listProduct.php');

**exit**();

} **else** {

$product = **new** Product();

$product->id = **isset**($\_GET['id']) ? htmlspecialchars($\_GET['id']) : '';

}

?>

VII. Description de la veille

Les failles de sécurité

1. Le htmlspecialchars

Cette fonction de PHP convertit les caractères spéciaux en entités HTML et permet ainsi permet l’injection de code HTML et JS.

Certains caractères ont des significations spéciales en HTML, et doivent être remplacés par des entités HTML pour conserver leurs significations. Cette fonction retourne une chaîne de caractères avec ces modifications.

Si la chaîne en entrée passée à cette fonction et le document final partagent le même jeu de caractères, cette fonction est suffisante pour préparer l’entrée pour une inclusion dans la plupart des contextes d’un document HTML.

1. La REGEX

La REGEX (expression régulière) et un fichier php que j’ai nommé “functions”. Voici quelques lignes de mes REGEX :

<?php

$regexName = '/^[a-zA-ZÀ-ÿ\' -]+$/';

$regexAddress = '/[0-9]{1,3}(**?:**(**?:**[,. ]){1,2}[-a-zA-ZÀ-ÿ]+)+$/';

$regexBirth = '/^(19|20)[0-9]{2}-[0-9]{2}-[0-9]{2}$/';

$regexPhone = '/^[0-9]{2}(-|\ ?)[0-9]{2}(-|\ ?)[0-9]{2}(-|\ ?)[0-9]{2}(-|\ ?)[0-9]{2}$/';

$regexMail = '/^[^\W][a-zA-Z0-9]+(.[a-zA-Z0-9]+)@[a-zA-Z0-9]+(.[a-zA-Z0-9]+).[a-zA-Z]{2,4}$/';

$regexPassword = '/^(**?=**.\*[a-z])(**?=**.\*[A-Z])(**?=**.\*[0-9])(**?=**.\*\W).{3,}$/';

?>

Explications : La regex ou expression régulière est une chaîne de caractères, qui décrit, selon une syntaxe précise, un ensemble de chaînes de caractères possibles. Une regex évite par exemple qu’un numéro de téléphone “alphanumérique” ne soit accepté car un numéro de téléphone n’est composé que de chiffres.

La partie validation des champs dans mon formulaire vient renforcer la sécurité du site. Vous remarquerez que ce code complète la regex créée en amont (code ci-dessus *$regexMail).* Voici mon code de validation de l’adresse email :

<?php

// Validation du mail

**if** (**empty**($supplier->mail)) {

$supplier->formErrors['mail'] = 'Ce champ est vide';

} **elseif** (!preg\_match($regexMail, $supplier->mail)) {

$supplier->formErrors['mail'] = 'Merci de rentrer un mail valide';

} **elseif** (strlen($supplier->mail) > 100) {

$supplier->formErrors['mail'] = 'Le mail est trop long';

} **elseif** (!$supplier->hasUniqueMail()) {

// Verifie si l'adresse mail existe déjà en base de données

$supplier->formErrors['mail'] = 'Mail existant, veuillez saisir une adresse mail différente';

}

?>

1. les fonctions
2. “function debug($data), debugPrint($data) et debugSession()”

* Ma fonction debug($data) m’a permis de faire un var\_dump pour identifier une erreur dans mon code et ainsi trouver une solution. Les codes de retour pour ces exceptions ont des noms au format SQL\_ERROR\_XXX dont voici le script :

<?php

**public** **function** \_\_construct() {

**try** {

// -- IMPORTANT --

// Les tables doivent être du type innoDB pour prendre en charge les transactions.

// Les tables de type MyISAM ne gèrent pas les transactions.

$**this**->db = **new** **PDO**('mysql:host=' . $**this**->host .';dbname=' . $**this**->database . ';charset=utf8',

$**this**->username,

$**this**->password);

// En mode ERRMODE\_EXCEPTION, si un échec survient, le script est

// interrompu, la connexion fermée, et mysql effectue un roll back

// sur la transaction

$**this**->db->setAttribute( **PDO**::ATTR\_ERRMODE, **PDO**::ERRMODE\_EXCEPTION );

} **catch** (**Exception** $e) {

**die**('Error : ' . $e->getMessage());

}

}

?>

* Ma fonction debugPrint(), à la différence de la fonction debug() elle permet d’avoir des informations sommaires.

La balise <pre> représente du texte préformaté, généralement écrit avec une police à chasse fixe. Le texte est affiché tel quel, les espaces utilisés dans le document HTML seront retranscrits (mets à la ligne).

* ”function debugSession()”

Cette fonction permet d’afficher l’id du *“role”.*

1. les 3 fonctions : isAdmin(), isUser() et isSupplier()

Ses 3 fonctions me permettent d’afficher certains éléments sur une page pour un *“role”* et non pour les autres. Elles permettent également de **limiter les données à chaque *“role”****.*

1. isAdmin()

*<?php*

*// Méthode certifiant que l'utilisateur est un administrateur*

***function*** *isAdmin() {*

***if*** *(****isset****($\_SESSION['user\_role']) && $\_SESSION['user\_role'] == 1) {*

***return*** *TRUE;*

*}* ***else*** *{*

***return*** *FALSE;*

*}*

*?>*

1. isUser()

<?php

// Méthode certifiant que l'utilisateur est un client enregistré

**function** isUser() {

**if** (**isset**($\_SESSION['user\_role']) && $\_SESSION['user\_role'] == 2) {

**return** TRUE;

} **else** {

**return** FALSE;

}

}

?>

1. isSupplier()

<?php

// Méthode certifiant que l'utilisateur est un fournisseur

**function** isSupplier() {

**if** (**isset**($\_SESSION['user\_role']) && $\_SESSION['user\_role'] == 3) {

**return** TRUE;

} **else** {

**return** FALSE;

}

}

?>

D) Le “hashage” de mots de passe via la fonction PHP password\_hash()

Le “hashage” de mot de passe est l’une des pratiques de sécurité les plus basiques qui doit être effectuée. Sans cela, chaque mot de passe stocké peut être volé si le support de stockage est compromis. Ce mot de passe peut alors être immédiatement utilisé pour accéder frauduleusement non seulement à votre application mais aussi sur d’autres applications si l’utilisateur utilise le même mot de passe ailleurs.

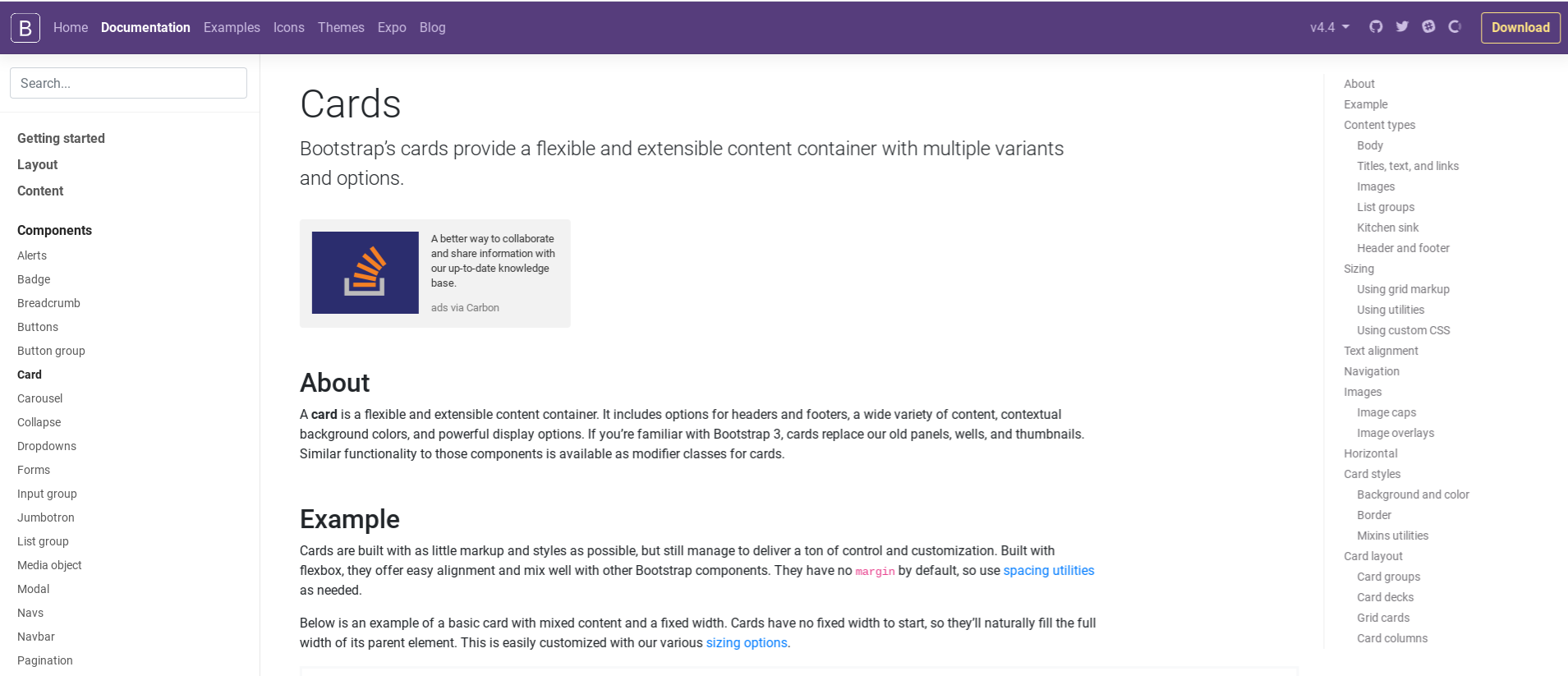
En appliquant un “hashage” sur le mot de passe avant de le stocker, vous rendez la tâche d’un attaquant très difficile pour connaître le mot de passe d’origine, et vous avez toujours la possibilité de comparer le mot de passe “hashé” à une chaîne reçue.

La fonction PHP password\_hash() va créer une chaîne de caractères aléatoire, et c’est généralement la façon la plus sécurisée et la plus simple.

VIII. Description d’une recherche anglophone

Au cours de mes différentes recherches, j’ai été amené à consulter différents sites ou tuto entièrement en anglais et notamment le site <https://getbootstrap.com/docs/4.4/components/card/> pour mes différentes vues (views) concernant l’affichage de la d’utilisateur, d’un fournisseur ou d’un produit.

Voici le modèle de “card” que j’ai utilisé :



Traduction :

“Cartes

Les [cartes](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/cartes) [de](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/de) [Bootstrap](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/Bootstrap) [offrent](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/offrent) [un](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/un) [conteneur](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/conteneur) [de](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/de) [conten](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/conten)u [flexible](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/flexible) [et](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/et) [extensible](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/extensible) [avec](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/avec) [de](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/de) [multiples](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/multiples) [variantes](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/variantes) [et](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/et) [options](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/options).

[À](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/%C3%80) [propos](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/propos) [d'une](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/d'une) [carte](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/carte) [est](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/est) [un](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/un) [conteneur](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/conteneur) [de](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/de) [conten](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/conten)u [flexible](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/flexible) [et](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/et) [extensible](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/extensible). [Il](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/Il) [comprend](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/comprend) [des](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/des) [options](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/options) [pour](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/pour) [les](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/les) [en-têtes](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/en-t%C3%AAtes) [et](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/et) [les](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/les) [pieds](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/pieds) [de](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/de) [page](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/page), u[ne](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/ne) [grande](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/grande) [variété](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/vari%C3%A9t%C3%A9) [de](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/de) [contenus](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/contenus), [des](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/des) [couleurs](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/couleurs) [de](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/de) [fond](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/fond) [contextuelles](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/contextuelles) [et](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/et) [de](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/de) [puissantes](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/puissantes) [options](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/options) [d'affichage](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/d'affichage). [Si](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/Si) [vous](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/vous) [connaissez](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/connaissez) b[ien](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/ien) [Bootstrap](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/Bootstrap) [3,](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/3,) [les](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/les) cartes [remplacent](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/remplacent) [nos](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/nos) [vieux](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/vieux) [panneaux](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/panneaux), [puits](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/puits) [et](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/et) [vignettes](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/vignettes). [Des](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/Des) [fonctionnalités](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/fonctionnalit%C3%A9s) [similaires](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/similaires) [à](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/%C3%A0) [ces](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/ces) [composants](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/composants) [sont](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/sont) [disponibles](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/disponibles) [sous](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/sous) [forme](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/forme) [de](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/de) [classes](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/classes) [de](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/de) [modification](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/modification) [pour](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/pour) [les](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/les) [cartes](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/cartes).

Exemples

[Les](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/Les) [cartes](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/cartes) [d'exemple](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/d'exemple) [sont](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/sont) [construites](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/construites) [avec](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/avec) [le](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/le) [moins](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/moins) [de](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/de) b[alisage](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/alisage) [et](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/et) [de](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/de) [styles](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/styles) [possible](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/possible), [mais](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/mais) [parviennent](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/parviennent) [à](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/%C3%A0) [offrir](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/offrir) u[ne](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/ne) [tonne](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/tonne) [de](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/de) [contrôle](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/contr%C3%B4le) [et](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/et) [de](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/de) [personnalisation](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/personnalisation). [Construits](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/Construits) [avec](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/avec) [la](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/la) [flexbox](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/flexbox), [ils](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/ils) [offrent](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/offrent) [un](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/un) [alignement](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/alignement) [facile](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/facile) [et](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/et) [se](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/se) [mélangent](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/m%C3%A9langent) b[ien](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/ien) [avec](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/avec) [les](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/les) [autres](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/autres) [composants](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/composants) [du](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/du) [Bootstrap](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/Bootstrap). [Ils](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/Ils) [n'ont](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/n'ont) [pas](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/pas) [de](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/de) [marge](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/marge) [par](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/par) [défaut](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/d%C3%A9faut), [donc](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/donc) u[tilisez](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/tilisez) [les](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/les) u[tilitaires](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/tilitaires) [d'espacement](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/d'espacement) [si](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/si) [nécessaire](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/n%C3%A9cessaire).

[Vous](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/Vous) [trouverez](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/trouverez) [ci-dessous](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/ci-dessous) [un](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/un) [exemple](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/exemple) [de](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/de) [carte](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/carte) [de](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/de) b[ase](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/ase) [avec](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/avec) [un](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/un) [conten](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/conten)u [mixte](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/mixte) [et](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/et) u[ne](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/ne) [largeur](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/largeur) [fixe](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/fixe). [Les](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/Les) [cartes](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/cartes) [n'ont](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/n'ont) [pas](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/pas) [de](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/de) [largeur](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/largeur) [fixe](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/fixe) [au](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/au) [départ](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/d%C3%A9part), [elles](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/elles) [rempliront](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/rempliront) [donc](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/donc) [naturellement](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/naturellement) [toute](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/toute) [la](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/la) [largeur](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/largeur) [de](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/de) [son](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/son) [élément](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/%C3%A9l%C3%A9ment) [parent](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/parent). [Il](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/Il) [est](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/est) [facile](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/facile) [à](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/%C3%A0) [personnaliser](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/personnaliser) [grâce](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/gr%C3%A2ce) [à](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/%C3%A0) [nos](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/nos) [différentes](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/diff%C3%A9rentes) [options](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/options) [de](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/de) [taille](https://fr.pons.com/traduction/fran%C3%A7ais-anglais/taille).”

IX. Remerciements

Je referme ce merveilleux chapitre en adressant mes remerciements aux formateurs Audrey DERLINCOURT et Guy LOUIS pour leur aide et leurs précieux conseils lors de cette formation.

Je remercie également toute l’équipe de l’école de laManu pour tout le travail effectué en amont, particulièrement les dirigeants et encadrants du campus de SOISSONS et NOYON.

Enfin merci à tous mes camarades apprenants pour la joie, la bonne ambiance, les bons moments et surtout l’esprit d’entraide.

IX. Conclusion

Ce projet a été pour moi un réel défi et une expérience très enrichissante car ce fût mon premier projet de E-Commerce.

Cela m’a permis de mettre en pratique l’enseignement reçu durant ses six mois de formation et d’approfondir davantage mes connaissances en développement web puis de me conforter dans mon choix de carrière.

Comme la plupart des projets qui nous tiennent à coeur, le temps nous manque… Les idées et les besoins du clients sont infinis. Cependant, je continuerai à développer mon site en y intégrant d’autres fonctionnalités, comme par exemple la mise en place d’un captcha pour améliorer la sécurité de mon site.