**Introduction générale**

**CHAPITRE 1 : CADRE GENERAL DU PROJET ET SPECIFICATIONS DES BESOINS**

**1.Introduction**

**2. Présentation du projet**

**3. Choix des méthodes, conception et modélisation**

**4. Spécification des besoins**

**4.1. Besoins fonctionnels :**

**4.2. Besoins non fonctionnels :**

**CHAPITRE 1 : MODELISATION CONCEPTUELLE (UML)**

**1.Introduction**

**2.Diagramme de cas d’utilisation**

**3.Diagramme de classe**

**3.Diagramme de séquences**

**CHAPITRE 3 : REALISATION TECHNIQUE**

**1.Introduction**

**2.Framwork LARAVEL 10**

**3.Environnement de développement IDE**

**4. Serveur local**

**5.Description de développement et test d’interfaçage**

**Conclusion générale**

**Introduction générale**

Le e-commerce ou commerce électronique regroupe l'ensemble des transactions commerciales s'opérant à distance par le biais d'interfaces électroniques et digitales. Il est devenu le principal canal de la vente ce qui explique le remplacement du terme de "vente par correspondance" par celui de "vente à distance". Il englobe essentiellement les transactions commerciales s'effectuant sur Internet à partir des différents types de terminaux sur des sites e-commerce ou applications mobiles marchandes.

Ce type d’applications représente un système mondial, offrant aux commerçants un pont pour accéder à toutes les informations, produits et services liés à un portail unique. Un site de vente en ligne permet aux commerçants de profiter des salons virtuels disponibles, qui sont mis à jour quotidiennement sans aucune restriction, ce qui leur permettra de ne jamais manquer de favoris, donc il n'y a pas de distance de travail géographique ou de distance par rapport au calendrier de travail équitable ou à la disponibilité du transport. En revanche, ces sites offrent à l'entreprise l'opportu plus large.

Dans ce contexte et dans le cadre de mon projet de fin d’études, je vise la mise en place d’une application web B2C que je dénoterai "Shop", une boutique en ligne de fruits et légumes organiques.

Le présent rapport permet de détailler les différentes étapes que j’ai suivi lors du développement de projet. Il se compose de trois chapitres. Dans le premier chapitre, je commence par présenter le cadre général du projet, saisir les besoins fonctionnels et non fonctionnels, ainsi que le cycle de développement adopté pour la mise en œuvre de mon projet. Le deuxième chapitre est consacré à la modélisation conceptuelle globale sur laquelle se base le projet.

Le troisième chapitre, présentent l’étude technique tout en déterminant les technologies et outils utilisés, et démonstration de code et l’interfaçage du l’application par des captures significatives.

nité de profiter de cet espace et de présenter ses produits à un groupe de commerçants

CHAPITRE 1 : CONTEXTE ET BESOINS DU PROJET

1. Introduction

Dans ce premier chapitre, nous abordons l'introduction du projet dans son ensemble ainsi que la spécification des besoins. Nous commençons par définir la portée et les objectifs du projet, ainsi que les exigences initiales.

1. Présentation du projet

Le projet vise à développer une application Web permettant aux commerçants de présenter leurs produits et leurs offres commerciales aux clients. Cette application offrira également un accès administrateur pour la gestion des contenus. Du côté des utilisateurs, ces derniers auront la possibilité de parcourir les produits, remplir leur panier et effectuer des achats en ligne. Ce projet est réalisé dans le cadre de l'épreuve E5 du BTS SIO (Services Informatiques aux Organisations), option SLAM (Solutions Logicielles et Applications Métiers).

1. Méthodologie, Conception et Modélisation

Dans le cadre de ce projet, la méthodologie de conception orientée objet a été choisie. Cette approche consiste à modéliser les objets du système et leurs interactions. Pour ce faire, nous utilisons le formalisme UML (Unified Modeling Language) qui est reconnu comme un outil performant pour la modélisation des systèmes informatiques.

1. Spécification des besoins

4.1. Besoins fonctionnels :

La conception de l'application web implique deux aspects principaux : la création des pages accessibles au grand public (Front Office) et la mise en place de l'espace d'administration (Back Office).

Pour les fonctionnalités accessibles aux clients, nous avons :

* Inscription
* Consultation des produits et des offres
* Ajout d'articles au panier
* Authentification
* Passage à la caisse

Pour les fonctionnalités du Back Office, nous avons :

* Inscription
* Gestion et visualisation des catégories de produits
* Gestion et visualisation des produits

4.2. Besoins non fonctionnels :

Les aspects non fonctionnels sont cruciaux pour assurer la performance et la satisfaction de l'utilisateur. Voici les exigences à prendre en compte :

* Fiabilité : L'application doit fonctionner de manière stable et sans erreur, garantissant ainsi une expérience utilisateur satisfaisante.
* Gestion des erreurs : Les messages d'erreur doivent être clairs et précis afin d'orienter l'utilisateur en cas de problème, assurant ainsi une utilisation fluide de l'application.
* Ergonomie des interfaces : L'interface utilisateur doit être intuitive et conviviale, facilitant la navigation et l'utilisation de l'application sans effort supplémentaire de la part de l'utilisateur.
* Sécurité : L'application doit garantir la sécurité des données personnelles et des transactions, en mettant en place des mesures robustes de protection contre les menaces potentielles.

CHAPITRE 2 : MODÉLISATION CONCEPTUELLE (UML)

1. Introduction

Dans ce chapitre, nous abordons la modélisation conceptuelle des principaux éléments de notre application web. Nous nous concentrons notamment sur le diagramme de cas d'utilisation, le diagramme de classes et le diagramme de séquence.

1. Diagramme de cas d'utilisation

Dans cette section, nous présentons le diagramme de cas d'utilisation, qui met en évidence les fonctionnalités globales de l'application ainsi que les différents acteurs qui interagissent avec celle-ci.

Une image contenant texte, motif, habits, tissu

Description générée automatiquement

1. Diagramme de classe

Dans cette partie, nous décrivons le diagramme de classe qui définit la structure de l'application en identifiant les différentes classes à développer ainsi que leurs méthodes associées.

**Une image contenant texte, diagramme, Plan, Dessin technique

Description générée automatiquement**

* + Diagramme de classe de l’application shop

**4.Diagramme de séquences**

1. Je présente ci-dessous un digramme de séquence d’une fonction sur le site qui permet à l’administrateur d’ajouter un produit à la vente, passant par la base de données, ainsi qu’une capture de l’interface d’ajout de produit.

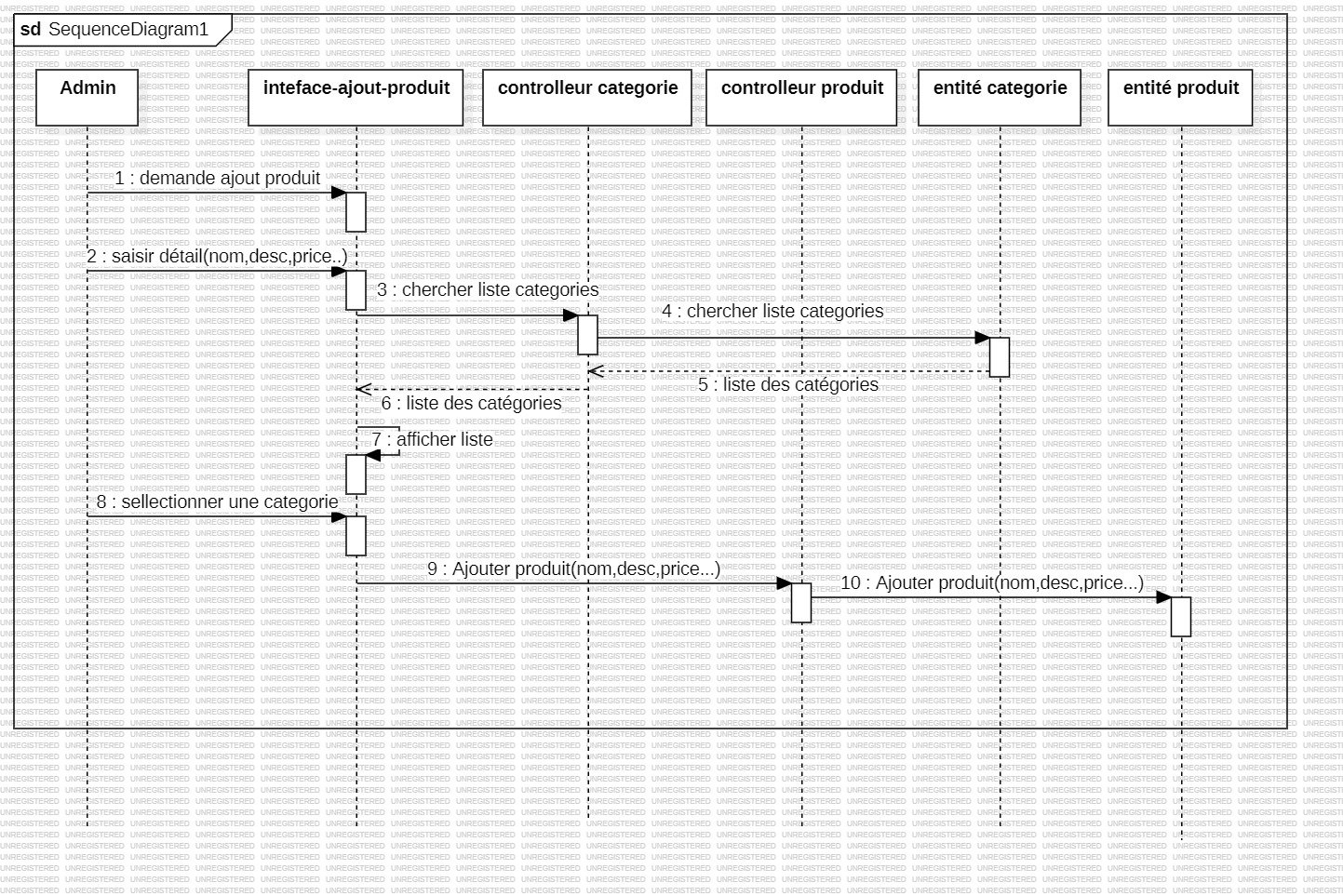


Diagramme de séquence : Ajout de produit

Une image contenant texte, capture d’écran, nombre, logiciel

Description générée automatiquement

**CHAPITRE 3 : REALISATION TECHNIQUE**

**1.Introduction**

Dans cette partie de projet, je vais m’intéresser à l’aspect techniques de l’application, à savoir les technologies utilisés tel que, les Framework, les SGBD , ainsi l’environnement de développement choisi tout au long du la réalisation techniques du projet. D’une autre part, je vais montrer les étapes techniques principales du la réalisation, accompagnés par des figures montrant le code et les interfaces visuelles du l’application.

**2.Framwork LARAVEL 10**

Dans le cadre de travail sur un Framework, j’ai opté pour le Framework Laravel de version 10. Ce dernier est un Framework PHP open-source réputé pour sa simplicité, sa rapidité de développement et sa robustesse. Il offre une structure organisée pour construire des applications web complexes grâce à ses nombreuses fonctionnalités intégrées telles que l'ORM Eloquent, la gestion des sessions, la sécurité avancée, ainsi que son système de routage et de templating efficace. En combinant des outils modernes et des conventions de développement bien définies, Laravel facilite la création d'applications web évolutives et maintenables.

**3.Environnement de développement IDE**

Comme, environnement de développement, je vais utiliser Visual Studio Code au long de projet, quant a lui, est un éditeur de code source gratuit et open-source développé par Microsoft. Il offre une multitude de fonctionnalités telles que la coloration syntaxique, l'achèvement automatique, le débogage intégré, la gestion des extensions, et la prise en charge de multiples langages de programmation.

**4. Serveur local**

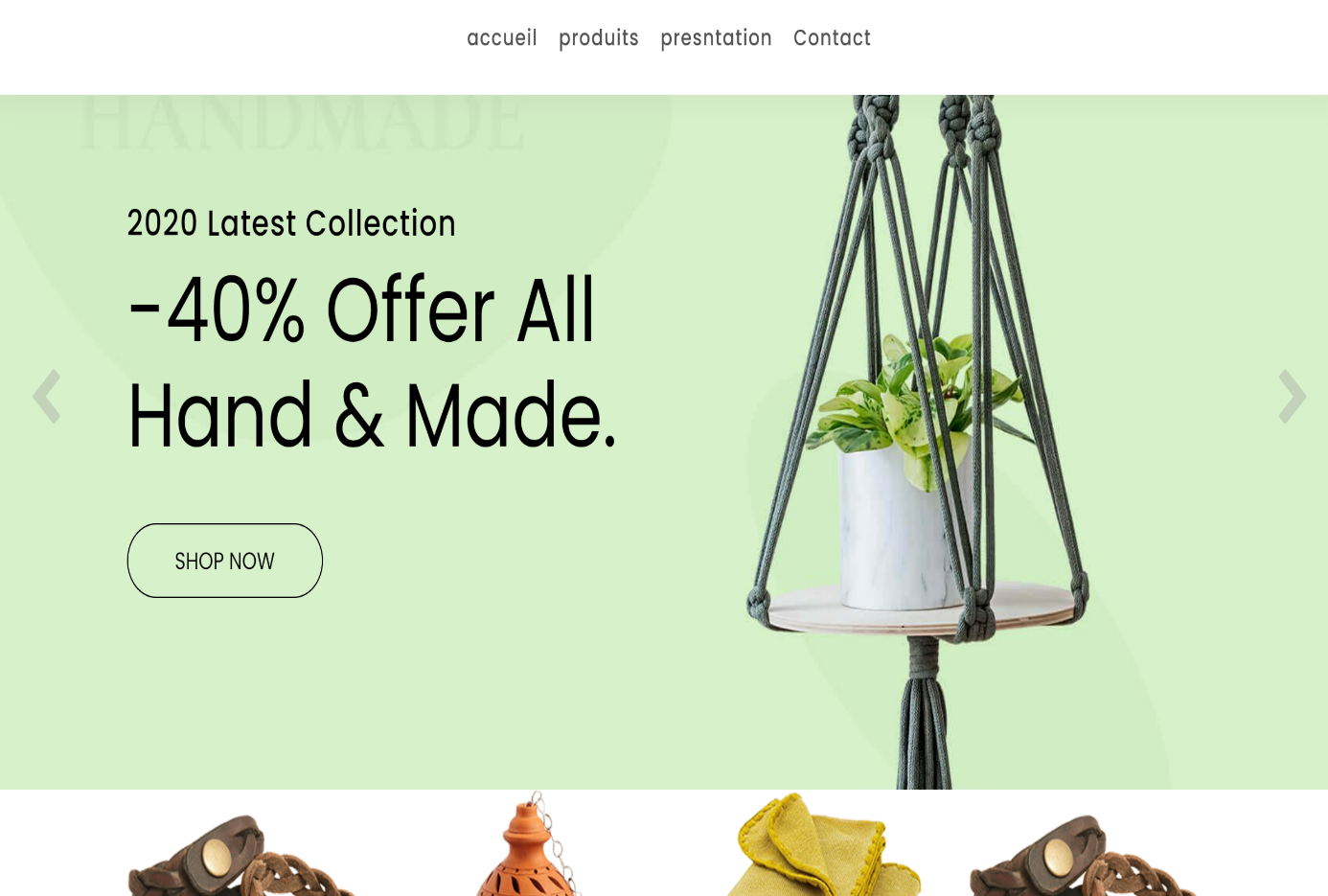
Afin de pouvoir exécuter le projet au niveau local, je vais basé sur l’outil Laragon pour plusieurs raison, premièrement il simplifier le processus de configuration et de gestion des serveurs web locaux, deuxièmement, il offre une interface conviviale permettant de démarrer rapidement des projets PHP avec des fonctionnalités telles que l'installation automatique de composer et l'intégration de bases de données

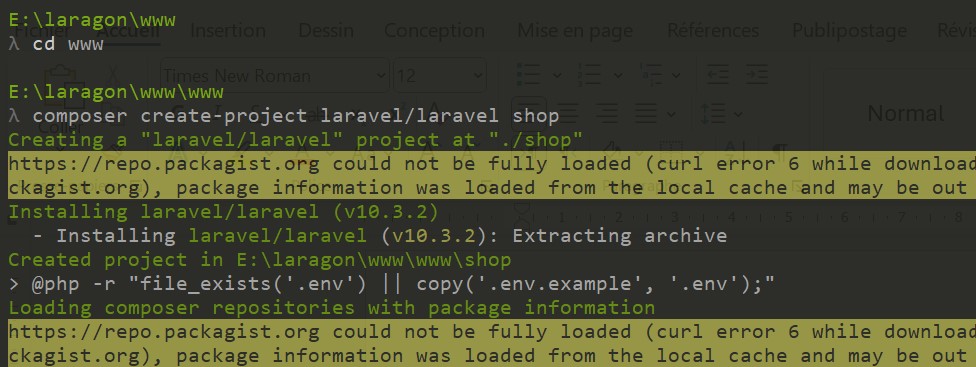
MySQL et MariaDB, ainsi que des outils pour la gestion des hôtes virtuels et des environnements de développement isolés.

**5.Description de développement et test d’interfaçage**

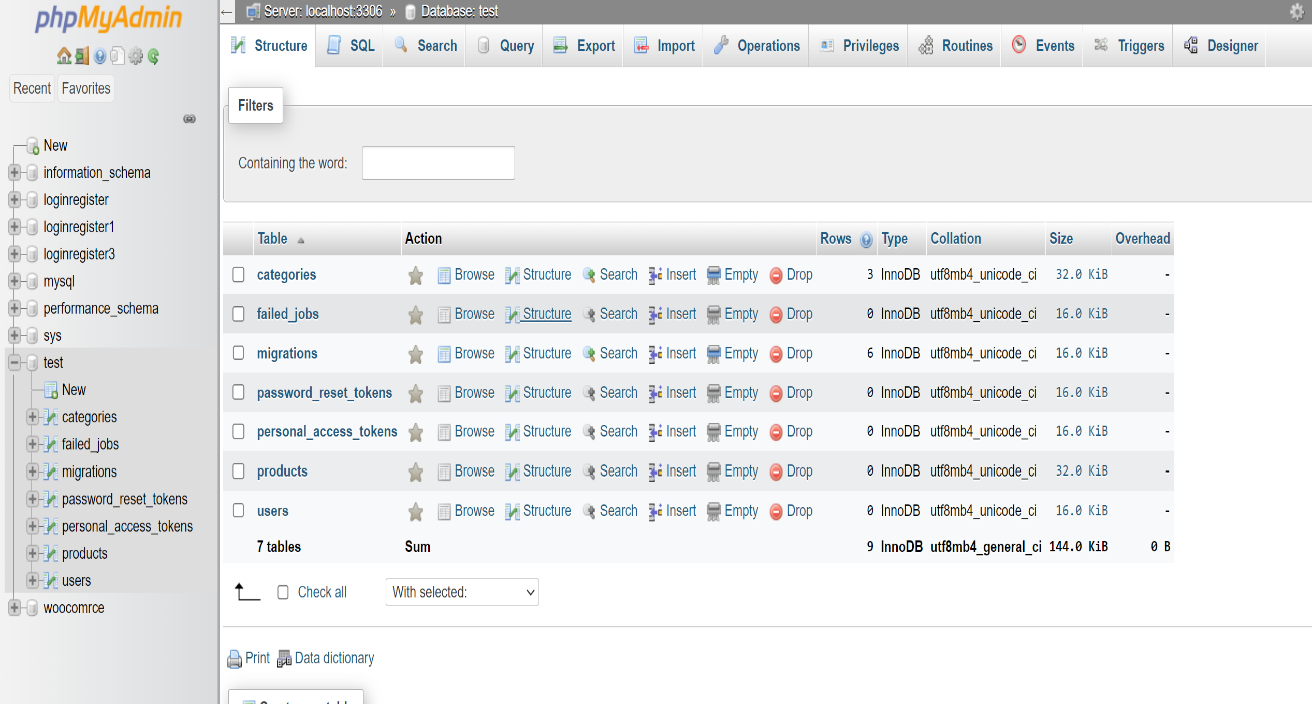
Dans cette section, je vais expliquer globalement le processus de développement et tout en démontrant des extraits de code et leurs résultat fonctionnels et visuels.

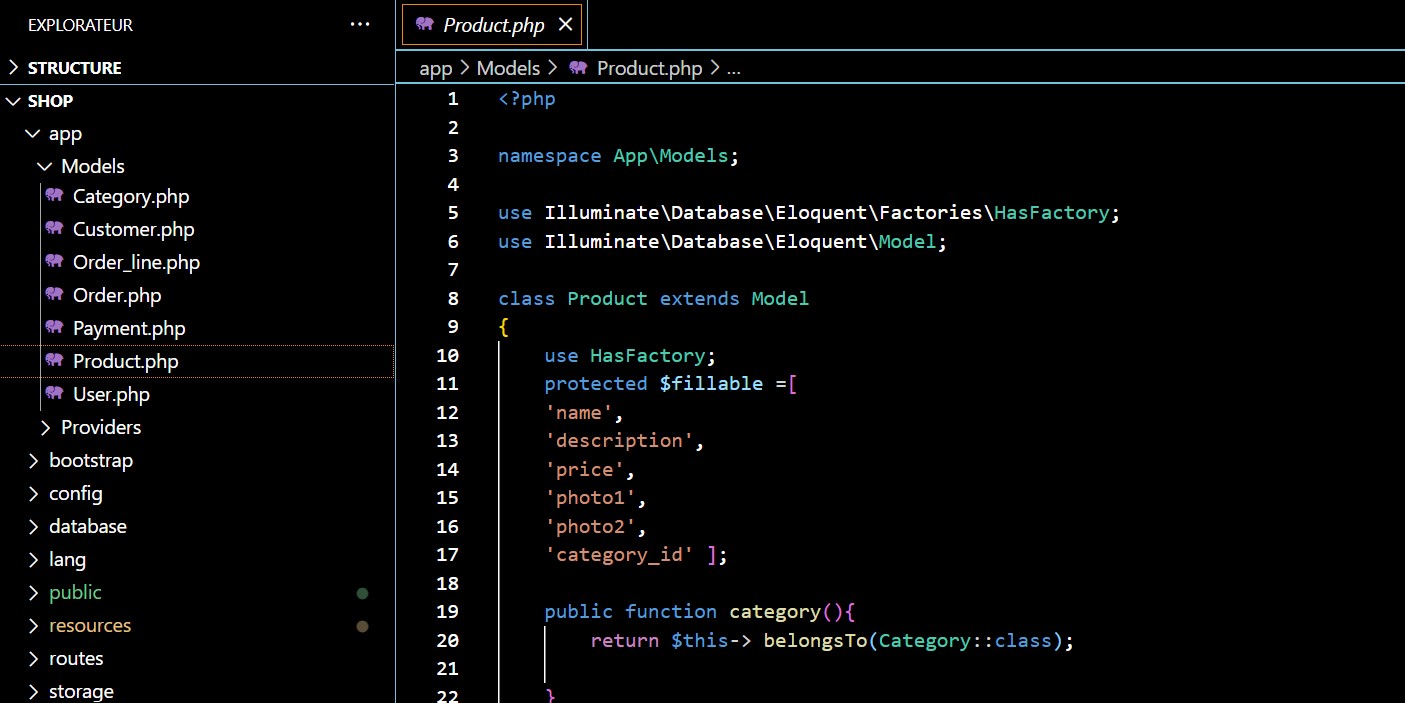
Après installer Laravel et mettre à jour Composer, j’ai ouvert l’invite de commande, dans le dossier www de Laragon, j’ai exécuté :

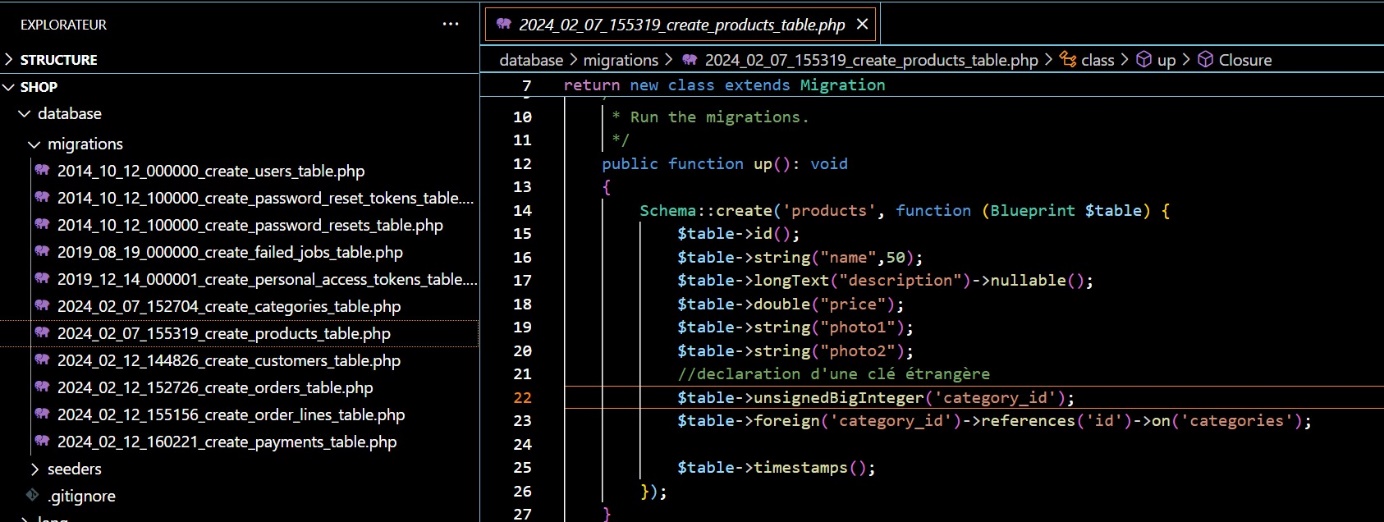




Pour la gestion de la base de données, je vais commencer par créer une base de données nommée "testlaravel" sur PhpMyAdmin, fourni par Laragon. Cette base de données servira à stocker toutes les informations nécessaires pour le fonctionnement de notre application.







Ensuite, je vais créer les contrôleurs correspondants à mes modèles, à savoir le **categoriesController** et le **productsController**. En utilisant des commandes simples, je pourrai générer ces contrôleurs avec des méthodes CRUD préremplies, ainsi que les routes correspondantes de type ressources.

php artisan make:controller CategoriesController --resource --model=Category

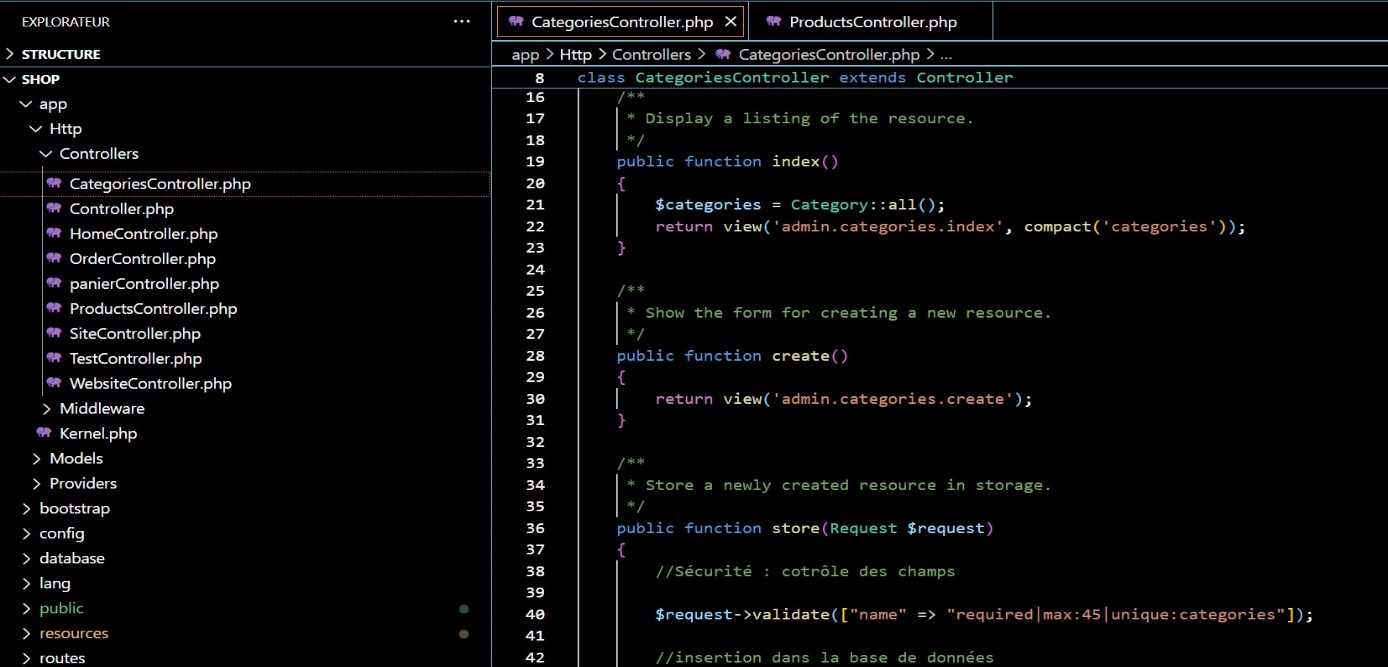
Cette commande va créer un contrôleur nommé **CategoriesController** avec les méthodes CRUD préremplies pour le modèle **Category**, ainsi que les routes correspondantes.

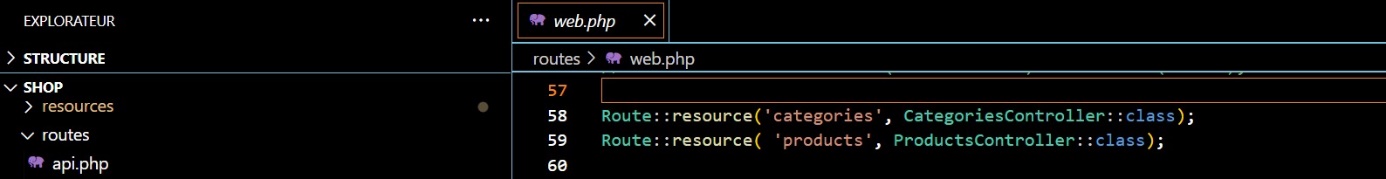
Pour le **productsController**, la commande serait similaire :

php artisan make:controller ProductsController --resource --model=Product

Cela va générer un contrôleur nommé **ProductsController** avec les méthodes CRUD préremplies pour le modèle **Product**, ainsi que les routes correspondantes.

Une fois ces contrôleurs créés, je pourrai les utiliser pour gérer les opérations de création, lecture, mise à jour et suppression des catégories et des produits dans mon application.



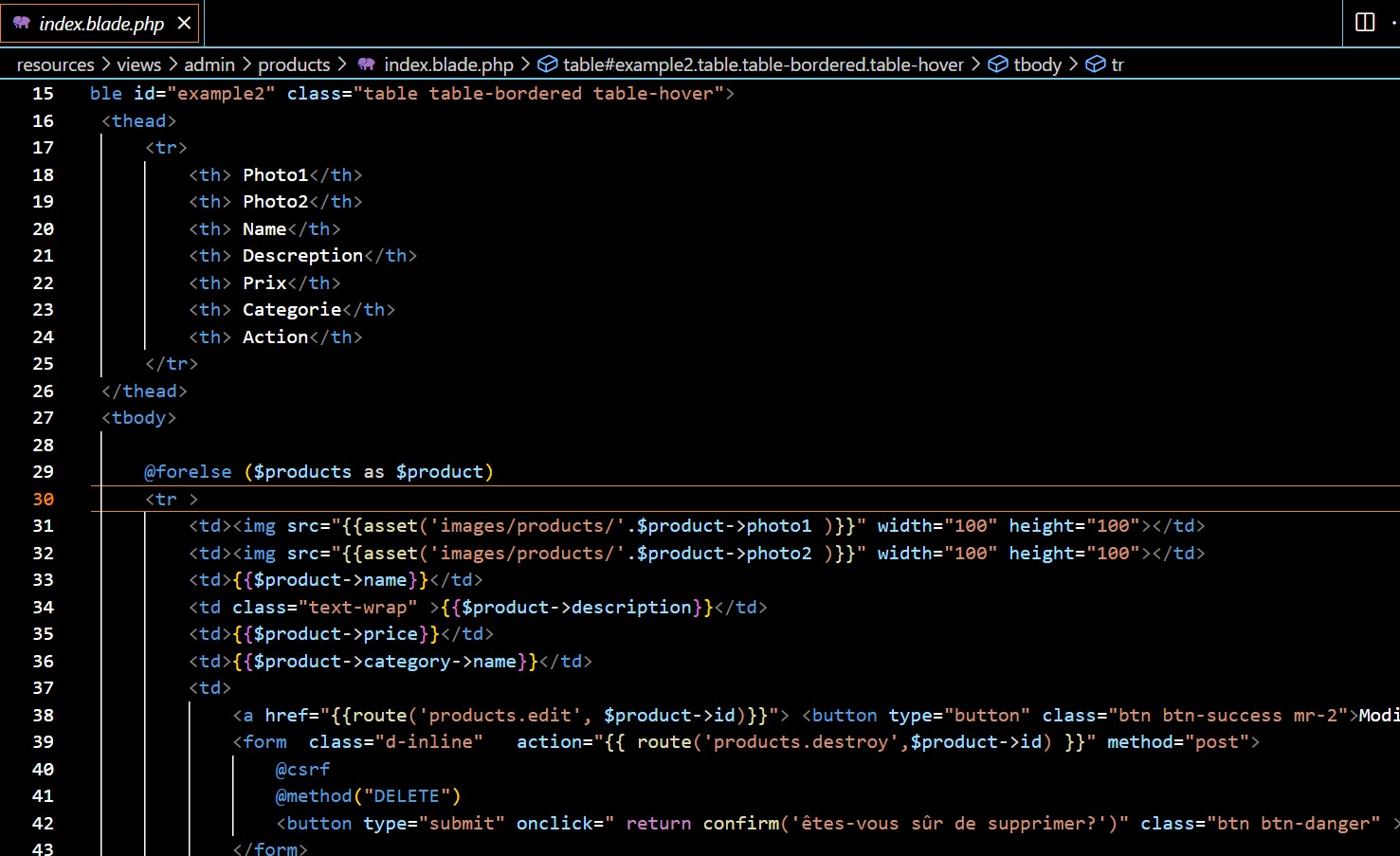


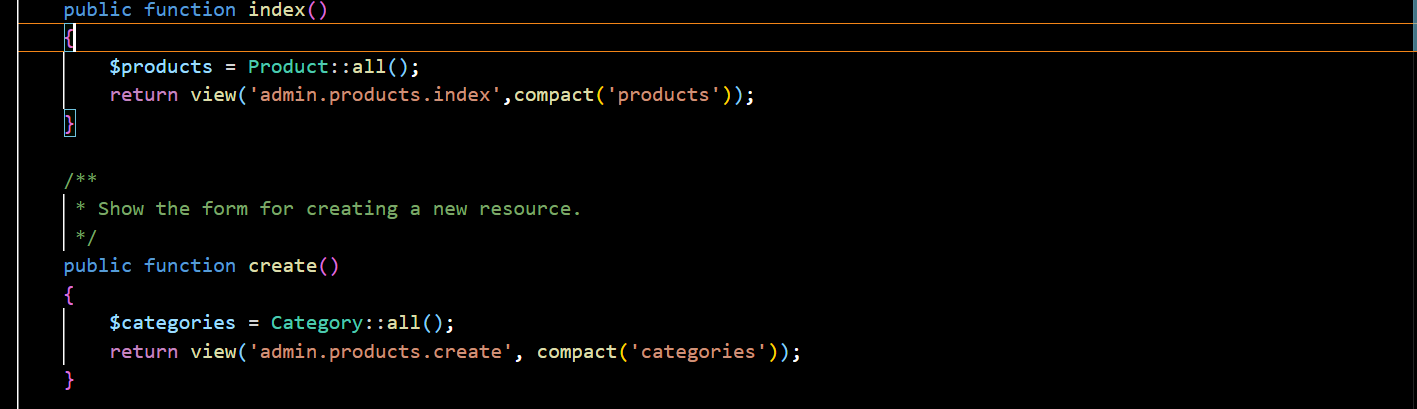
Après avoir accompli toutes les tâches préalables avec succès, je vais désormais débuter la création de la section administration de mon application. Pour commencer, je vais importer un template admin pour bénéficier d'une interface utilisateur déjà élaborée et fonctionnelle.

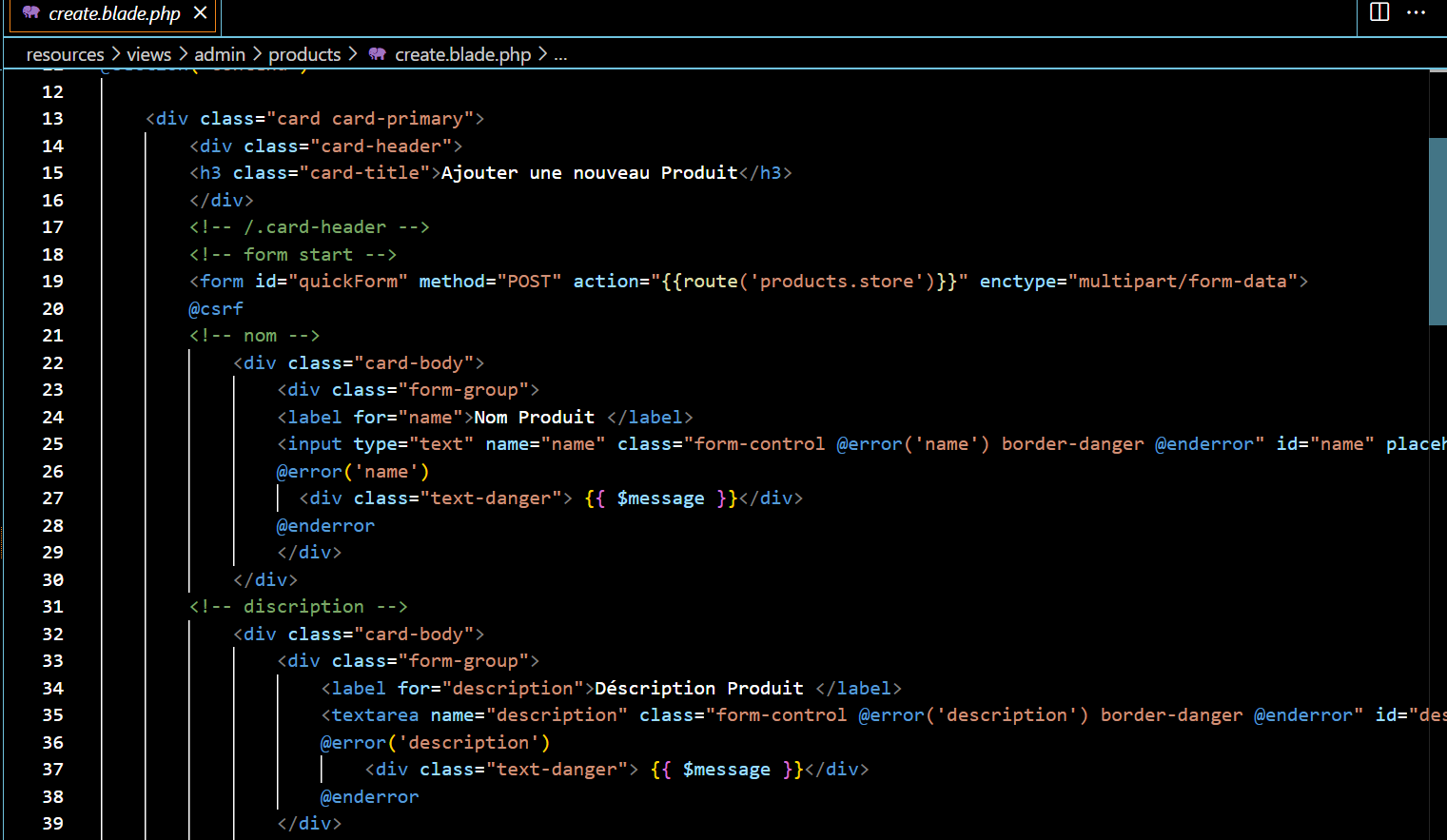


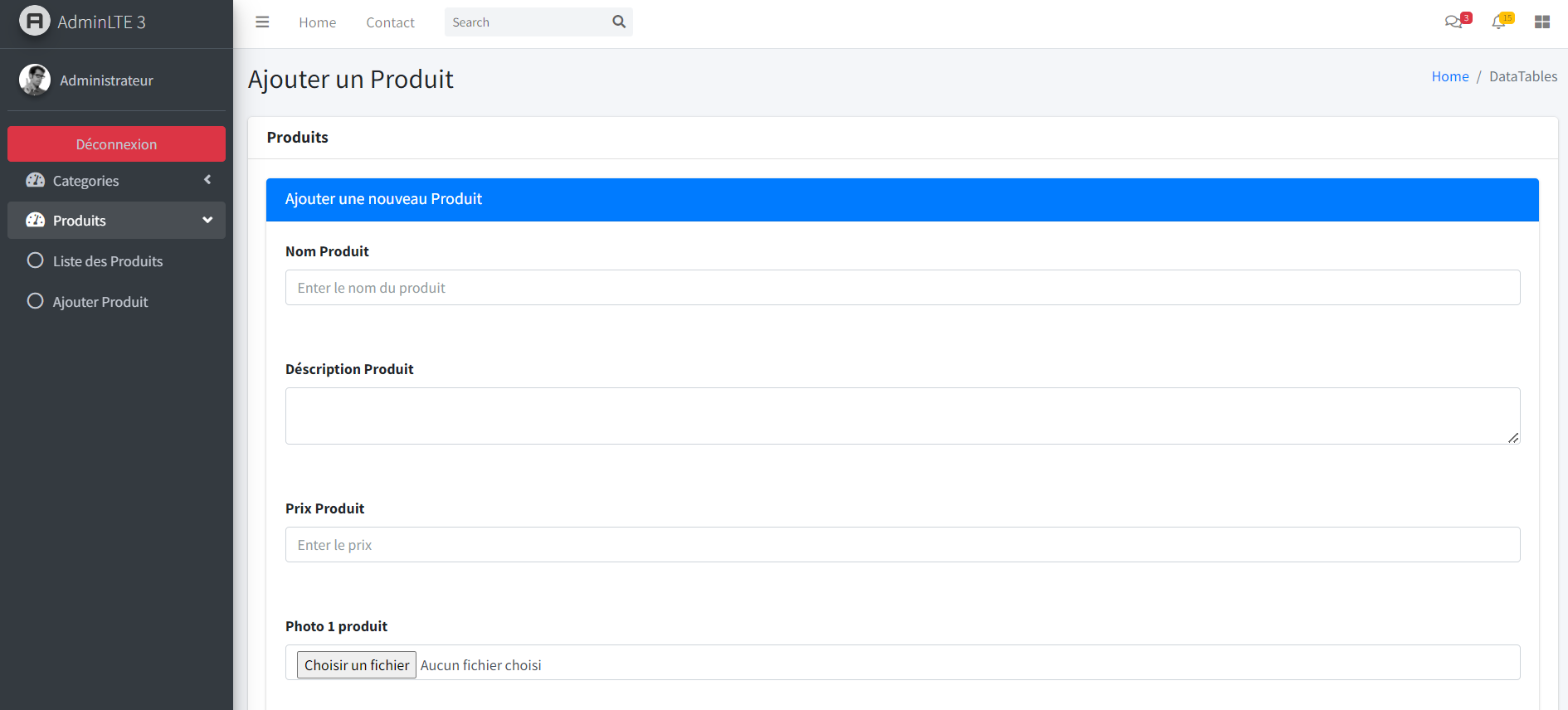
Pour créer la vue index destinée à la gestion des produits dans la partie admin de mon application, voici une approche concise

améliorer son controlleur qui va afficher la liste des produits et contenir des lien pour supprimer et modifier



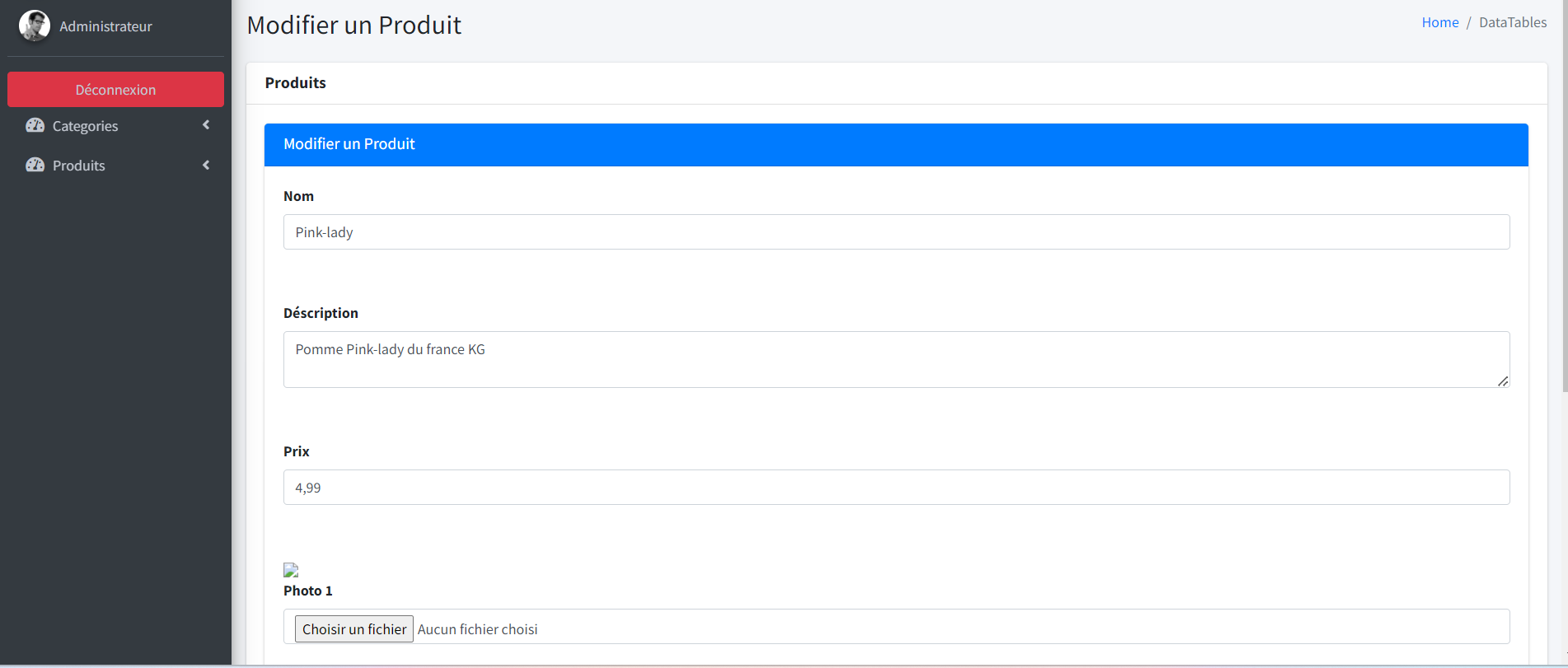






la vue ajouter nommé create.blade.php :





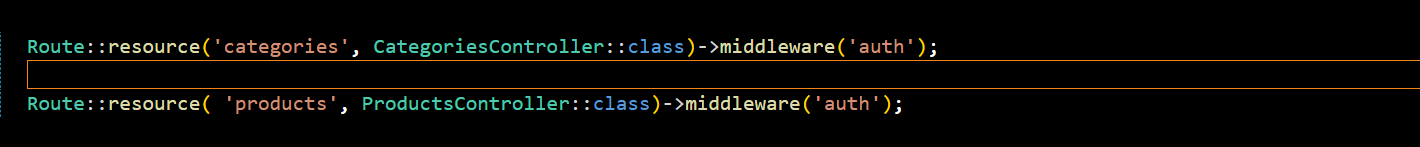
la vues modifier nommé edit.blade.php

Pour sécuriser la partie admin de mon application en demandant l'inscription à l'administrateur avant d'y accéder, je vais commencer par ajouter le package d'authentification au projet Laravel. Voici comment je vais procéder avec une approche alternative :

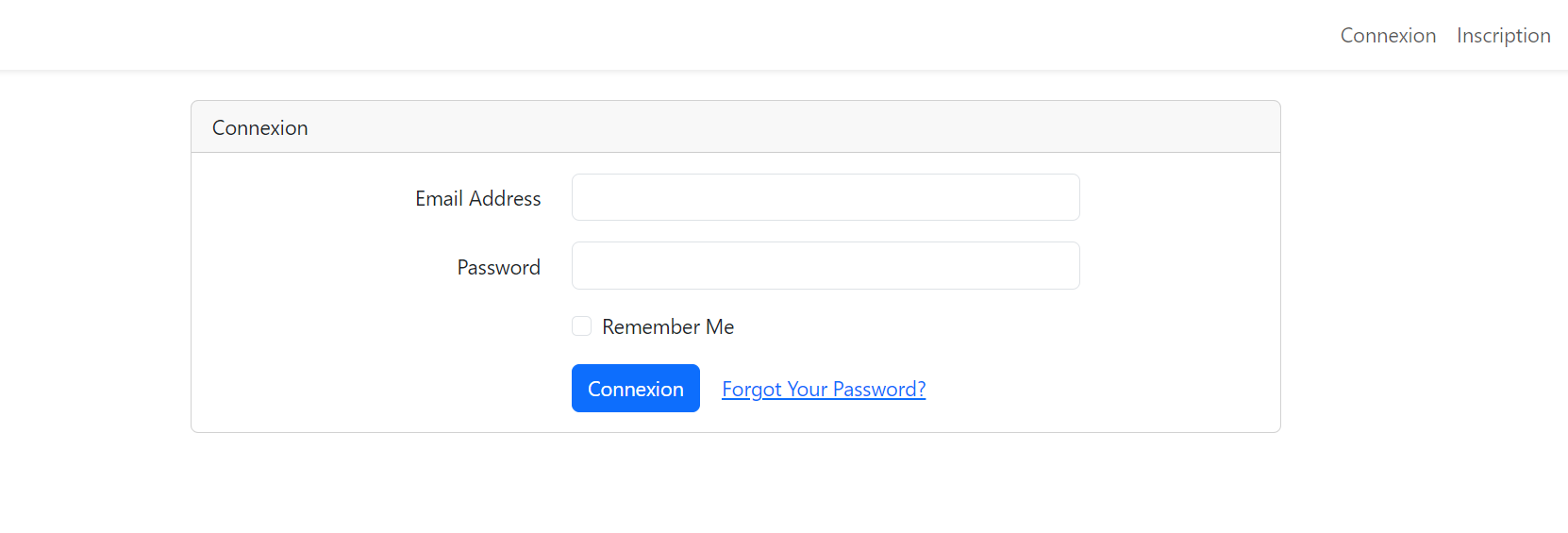
1. **Installation du package d'authentification :** Je vais ouvrir une fenêtre de terminal et exécuter la commande suivante à la racine de mon projet Laravel :
2. **Configuration du package :** Après avoir installé le package, je vais utiliser la commande artisan pour générer les vues et les routes nécessaires à l'authentification :
3. **Exécution des migrations :** Ensuite, je vais exécuter les migrations pour créer la table des utilisateurs et d'autres tables nécessaires à l'authentification :
4. **Protection des routes admin avec un middleware :** Je vais créer un middleware personnalisé pour vérifier si l'utilisateur est authentifié en tant qu'administrateur avant de lui permettre l'accès à la partie admin de l'application. Pour cela, je vais exécuter la commande artisan suivante :
5. **Implémentation de la logique dans le middleware :** J'ouvrirai le fichier créé dans le répertoire **app/Http/Middleware/AdminMiddleware.php** et j'ajouterai la logique nécessaire pour vérifier si l'utilisateur est authentifié et s'il possède les permissions d'administrateur. Si l'utilisateur n'est pas authentifié ou s'il n'est pas un administrateur, je le redirigerai vers la page de connexion.
6. **Enregistrement du middleware :** Enfin, je vais enregistrer le middleware dans le kernel HTTP en ajoutant son alias dans le tableau **$routeMiddleware** du fichier **app/Http/Kernel.php**.

En suivant ces étapes, je pourrai sécuriser la partie admin de mon application en demandant l'inscription à l'administrateur avant d'y accéder.

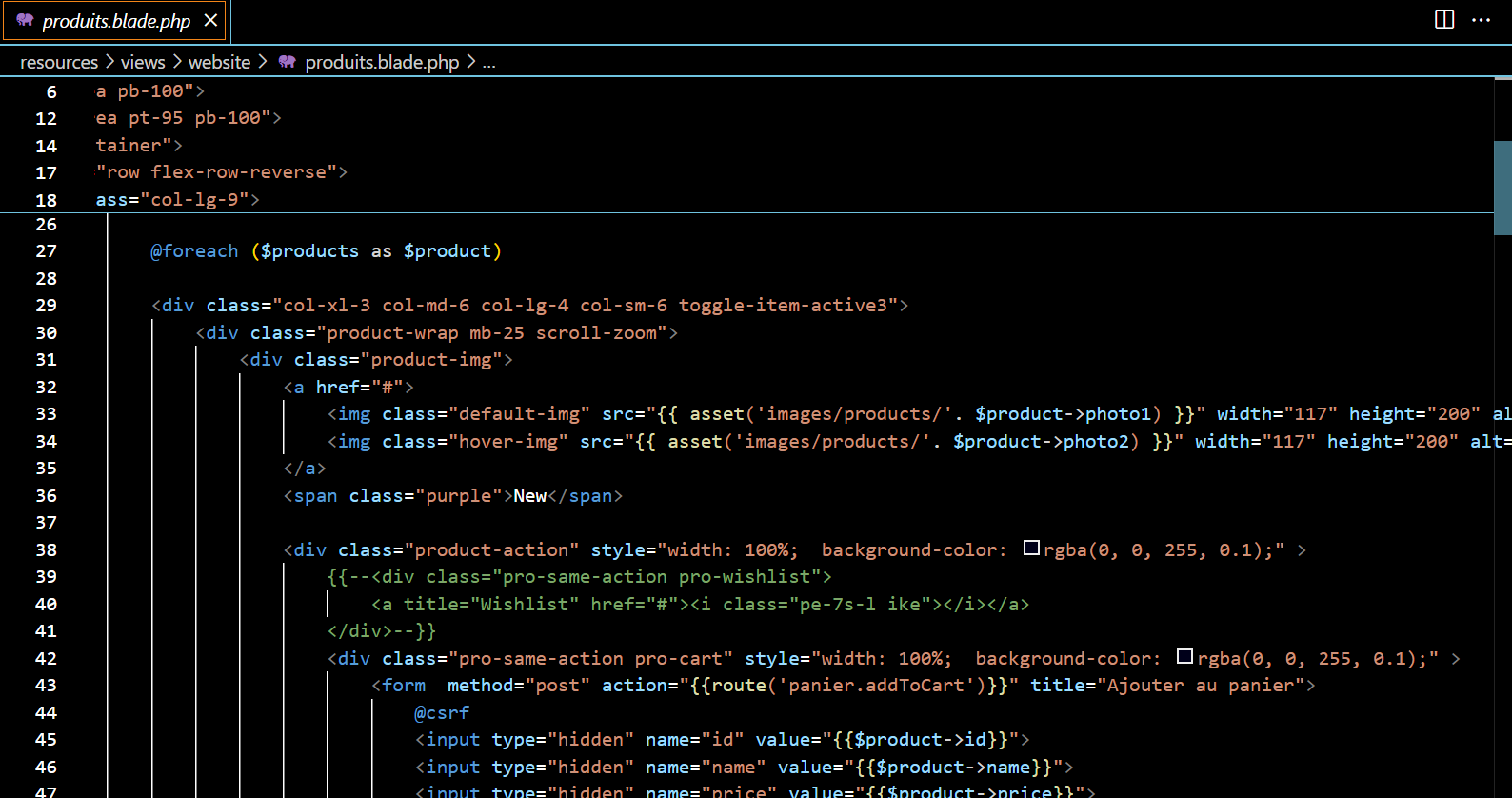
Puis dans les routes je dois configurer les routes corresponds à la partie admin :

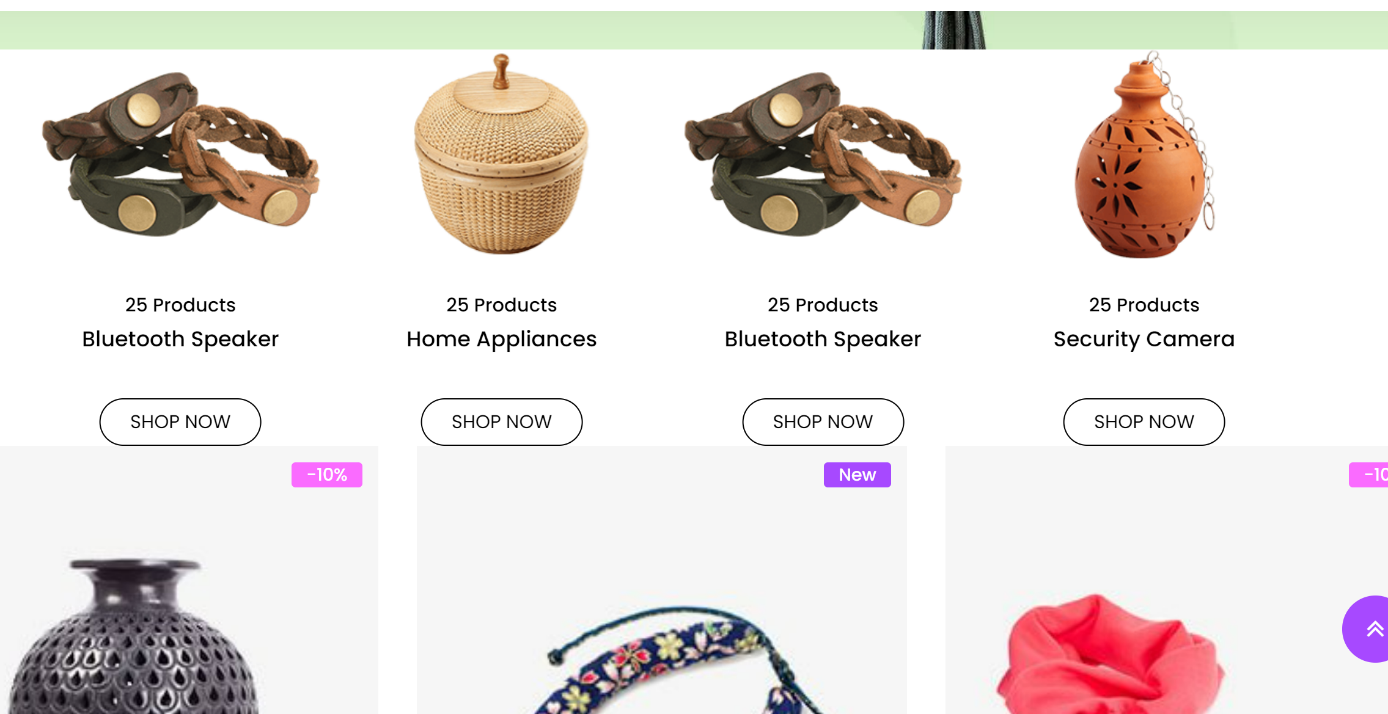


Et j’obtiens la vue login avant d’accéder à cette partie :

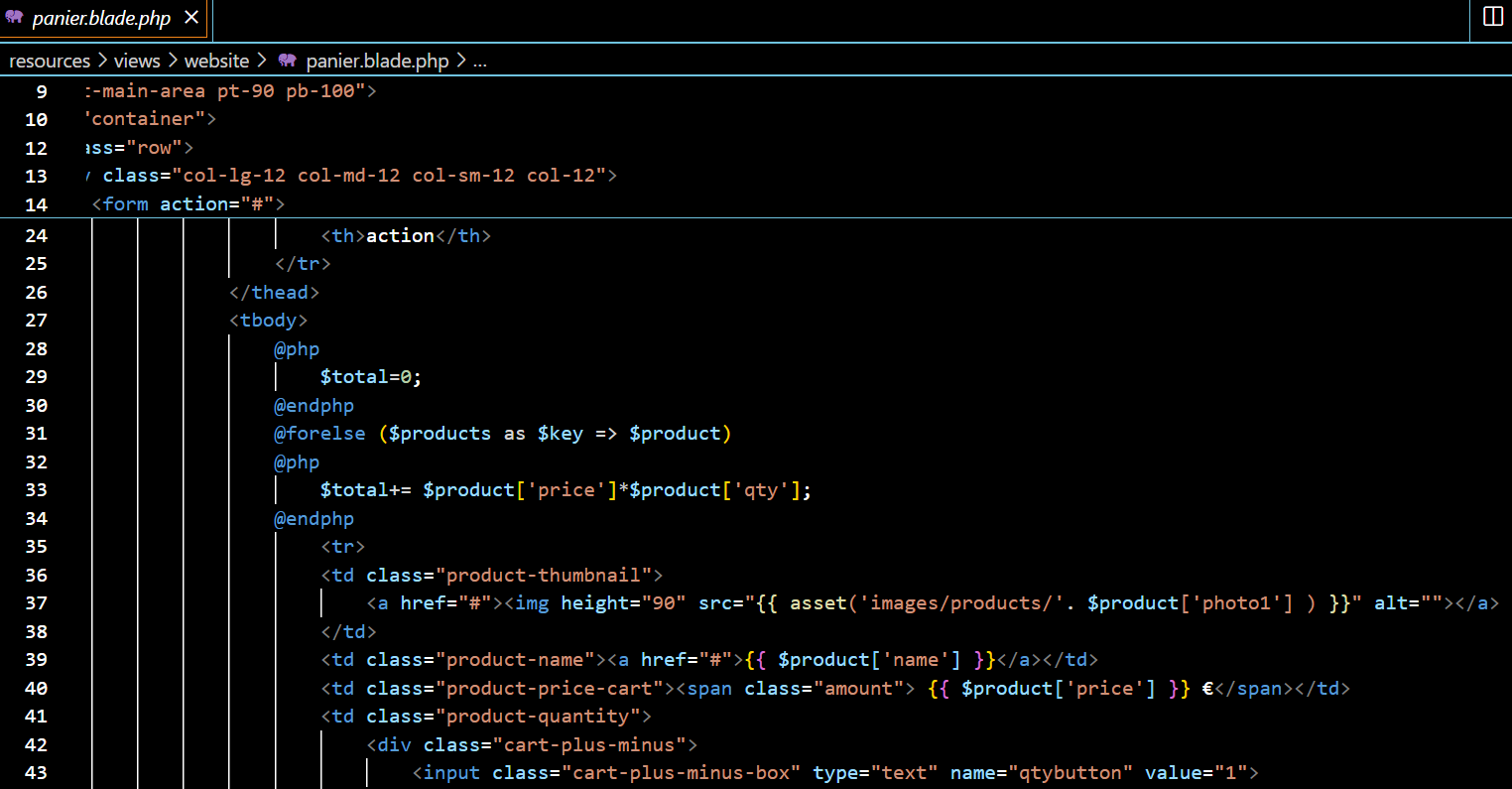


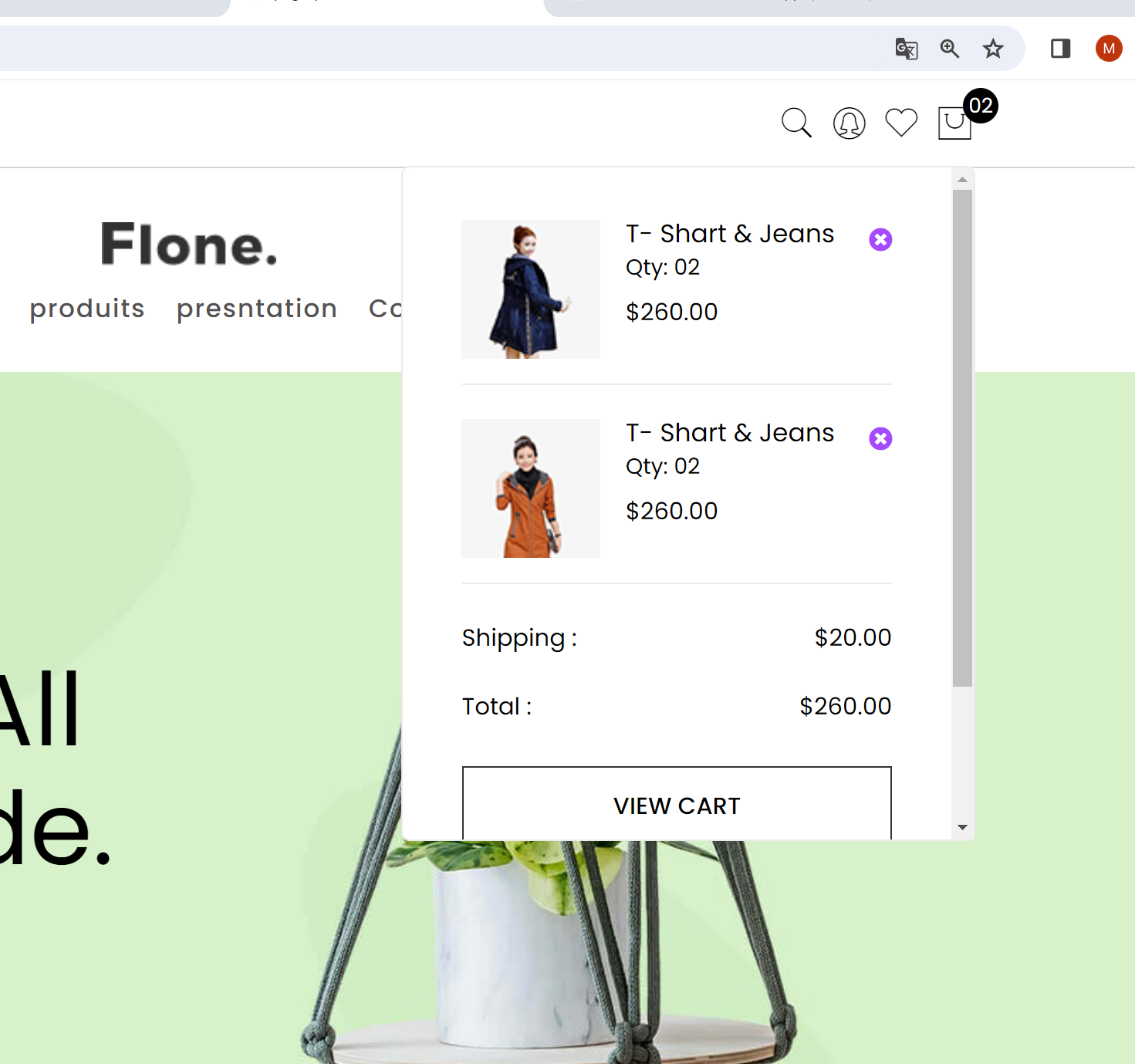
Amélioration de la page produits : extraction des données de la base de données, présentation dynamique avec pagination et fonctionnalités de filtrage/recherche, optimisation esthétique pour une expérience utilisateur optimale.



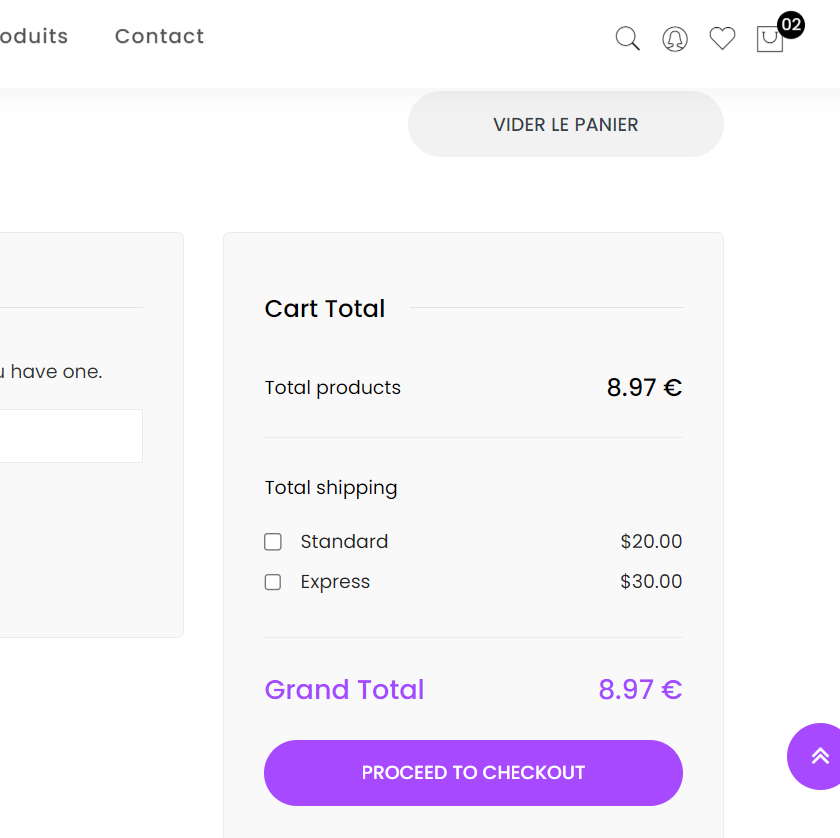


Puis je développe le remplissage, l’affichage de panier et la vue panier :





Après le remplissage je développe la partie vérification (checkout) et son bouton de redirection



Les utilisateurs peuvent s'authentifier en remplissant des formulaires, leur donnant accès aux paiements une fois cette étape terminée.

**Conclusion générale**

En résumé, j'ai suivi une approche méthodique basée sur UML pour concevoir et développer l'application étape par étape. Après avoir compris les besoins du projet, j'ai établi les modèles conceptuels et choisi les méthodologies appropriées. J'ai ensuite procédé à la mise en œuvre technique en utilisant divers outils et technologies, illustrant les aspects cruciaux à travers des extraits de code et des interfaces utilisateur.

Cette expérience m'a permis d'appliquer mes connaissances académiques et de me familiariser avec de nouvelles technologies. En intégrant ces compétences, j'ai consolidé mon apprentissage et j'ai pu me préparer à une transition réussie vers le monde professionnel.