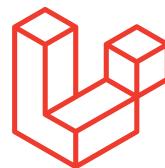




TP4 E-commerce - Application Laravel



Laravel



Application e-commerce complète développée avec Laravel dans le cadre du TP4 de Programmation Web.

📋 Table des matières

- [Aperçu](#)
- [Fonctionnalités](#)
- [Technologies utilisées](#)
- [Prérequis](#)
- [Installation](#)
- [Configuration](#)
- [Utilisation](#)
- [Structure du projet](#)
- [Équipe](#)
- [Contribution](#)
- [Documentation](#)
- [License](#)

🎯 Aperçu

Application e-commerce full-stack permettant aux utilisateurs de :

- Parcourir un catalogue de produits
- Ajouter des articles au panier
- Passer des commandes
- Suivre l'historique des achats

Les administrateurs peuvent :

- Gérer les produits (CRUD)
- Suivre les commandes
- Accéder aux statistiques

✨ Fonctionnalités

Pour les clients

- Authentification (inscription/connexion)
- Catalogue de produits avec recherche et filtres
- Fiche produit détaillée
- Panier d'achat persistant
- Système de commande
- Paiement simulé
- Historique des commandes

Pour les administrateurs

- Dashboard avec statistiques
- Gestion des produits (ajout, modification, suppression)
- Gestion des commandes
- Mise à jour des statuts de commande

Technologies utilisées

Backend

- **Laravel 10.x** - Framework PHP
- **MySQL** - Base de données
- **Eloquent ORM** - Gestion des données
- **Blade** - Moteur de templates

Frontend

- **HTML5 / CSS3**
- **JavaScript (Vanilla)**
- **Bootstrap 5 (optionnel)**
- **Responsive Design**

Outils

- **Composer** - Gestionnaire de dépendances PHP
- **NPM** - Gestionnaire de dépendances JS
- **Git** - Contrôle de version
- **Vite** - Bundler d'assets

Prérequis

Avant de commencer, assurez-vous d'avoir installé :

- PHP >= 8.1
- Composer >= 2.5
- Node.js >= 18.x
- MySQL >= 8.0
- Git

Installation

1. Cloner le

```
git clone https://github.com/Delmat237/Shopecart-Web-Project.git  
cd Shopecart-Web-Project
```

Accéder à la branche

```
git checkout tp/4-laravel-full
```

Accéder au projet

```
cd tp4-ecommerce
```

2. Installer les dépendances PHP

```
composer install
```

3. Installer les dépendances JavaScript

```
npm install
```

4. Créer le fichier de configuration

```
cp .env.example .env
```

5. Générer la clé d'application

```
php artisan key:generate
```

6. Créer la base de données

Créez une base de données MySQL nommée **tp4_ecommerce**:

```
CREATE DATABASE tp4_ecommerce CHARACTER SET utf8mb4 COLLATE  
utf8mb4_unicode_ci;
```

7. Configurer la base de données

Modifiez le fichier `.env` avec vos identifiants :

```
DB_CONNECTION=mysql  
DB_HOST=127.0.0.1  
DB_PORT=3306  
DB_DATABASE=tp4_ecommerce  
DB_USERNAME=root  
DB_PASSWORD=votre_mot_de_passe
```

8. Exécuter les migrations et seeders

```
php artisan migrate --seed
```

9. Créer le lien symbolique pour le stockage

```
php artisan storage:link
```

10. Compiler les assets

```
npm run dev
```

11. Lancer le serveur de développement

```
php artisan serve
```

L'application sera accessible sur : **http://localhost:8000**

Configuration

Comptes de test

Après avoir exécuté les seeders, vous pouvez utiliser ces comptes :

Administrateur

- **Email** : admin@ecommerce.com
- **Mot de passe** : password

Client

- **Email** : client@ecommerce.com
- **Mot de passe** : password

Configuration du mail (optionnel)

Pour tester l'envoi d'emails en local, utilisez Mailtrap :

```
MAIL_MAILER=smtp
MAIL_HOST=smtp.mailtrap.io
MAIL_PORT=2525
MAIL_USERNAME=votre_username
MAIL_PASSWORD=votre_password
```

📖 Utilisation

Commandes Artisan utiles

```
# Vider tous les caches
php artisan optimize:clear

# Réinitialiser la base de données
php artisan migrate:fresh --seed

# Lister toutes les routes
php artisan route:list

# Créer un nouvel utilisateur admin
php artisan tinker
>>> User::create(['name' => 'Admin', 'email' => 'admin@test.com',
  'password' => bcrypt('password'), 'is_admin' => true])
```

Compiler les assets pour la production

```
npm run build
```

📁 Structure du projet

```
tp4-ecommerce/
|
```

```
app/
  └── Http/
    ├── Controllers/
    │   ├── Auth/          # Authentification
    │   ├── Admin/         # Contrôleurs admin
    │   ├── CartController.php
    │   ├── OrderController.php
    │   ├── PaymentController.php
    │   └── ProductController.php
    ├── Middleware/
    └── Requests/
  └── Models/
    ├── User.php
    ├── Product.php
    ├── Category.php
    ├── Cart.php
    ├── CartItem.php
    ├── Order.php
    └── OrderItem.php

  └── database/
    ├── migrations/
    └── seeders/

  └── resources/
    ├── css/
    ├── js/
    └── views/
      ├── layouts/
      ├── components/
      ├── auth/
      ├── products/
      ├── cart/
      ├── orders/
      ├── payment/
      └── admin/

  └── routes/
    └── web.php

  └── public/
    ├── css/
    ├── js/
    └── images/
```

👥 Équipe

Équipe 1 - Gestion Produits & BDD

- **A** - Setup & Architecture Laravel
- **C** - Migrations & Seeders
- **D** - Authentification

Équipe 2 - Frontend Produits

- **E** - CRUD Produits (Backend)
- **F** - Routes & Controllers Produits (Frontend)

Équipe 3 - Panier & Paiement

- **G** - Gestion Commandes
- **I** - Système de Paiement
- **L** - Panier (CRUD)

Équipe 4 - Admin & Intégration

- **K** - Dashboard Admin
- **M** - Intégration Blade & Tests

Équipe CMS (TP3)

- **H, J** - Travaillent sur le CMS (TP3)

🤝 Contribution

Workflow Git

1. Créer une branche pour votre fonctionnalité

```
git checkout -b feature/nom-fonctionnalite
```

2. Faire vos modifications et commits

```
git add .
git commit -m "feat: description de la fonctionnalité"
```

3. Pousser votre branche

```
git push origin feature/nom-fonctionnalite
```

4. Créer une Pull Request sur GitHub

Convention de commits

Utilisez les préfixes suivants :

- **feat:** - Nouvelle fonctionnalité
- **fix:** - Correction de bug
- **docs:** - Documentation

- **style:** - Formatage, style
- **refactor:** - Refactorisation de code
- **test:** - Ajout de tests
- **chore:** - Tâches de maintenance

Exemples :

```
feat: ajout du système de panier  
fix: correction du calcul du total  
docs: mise à jour du README
```

Documentation

- [Guide d'installation complet](#)
- [Documentation API](#)
- [Guide de contribution](#)
- [Architecture du projet](#)

Résolution des problèmes

Erreur de migration

```
php artisan migrate:fresh --seed
```

Erreur de permission

```
chmod -R 775 storage bootstrap/cache
```

Assets non chargés

```
npm run dev  
php artisan storage:link
```

Vider tous les caches

```
php artisan optimize:clear
```

License

Ce projet est sous licence MIT. Voir le fichier [LICENSE](#) pour plus de détails.

 **Support**

Pour toute question ou problème :

- Créer une [issue](#)
 - Contacter l'équipe via Discord/Slack
-

Développé avec ❤️ par l'équipe - Programmation Web 2025