



# Plan de développement du projet Laravel

## 1) Mise en place du projet

- Créer un nouveau projet Laravel 12 :

```
composer create-project laravel/laravel:^12 nom_du_projet
```

- Configurer le fichier `.env` (base de données MySQL, driver mail, etc.).
- Générer la clé de l'application :

```
php artisan key:generate
```

- Lancer les migrations de base :

```
php artisan migrate
```

- **Documentation de référence** : [Laravel 12 - Installation](#)
- 

## 2) Authentification (Login / Signup)

- Installer Laravel UI (exigence du projet) :

```
composer require laravel/ui  
php artisan ui:auth  
php artisan migrate
```

- Vérifier la redirection des routes protégées avec `auth`.
  - Réécrire les vues générées (par défaut en Bootstrap) en **HTML + CSS pur**.
  - **Documentation** : [Laravel Authentication](#)
- 

## 3) Modèles & Migrations

- Tables nécessaires :
- `users` (déjà fournie par Laravel)
- `dishes` : contient les informations d'un plat (créateur, nom, description, recette chiffrée, image, timestamps).
- `favorites` : table pivot `user_id`, `dish_id` pour gérer les favoris (N-N).
- Génération avec :

```
php artisan make:model Dish -m
php artisan make:migration create_favorites_table
```

- Mise en place des relations Eloquent (`hasMany`, `belongsTo`, `belongsToMany`).
- **Documentation** : [Eloquent ORM](#)

---

## 4) Chiffrement de la recette

- Créer un **trait** `Encryptable` qui utilise `Crypt` pour chiffrer/déchiffrer le champ `recipe`.
- Ajouter `protected $encryptable = ['recipe'];` dans le modèle `Dish`.
- **Documentation** : [Encryption](#)

---

## 5) Gestion des images

- Utiliser le **disk** `public`.
- Créer le dossier `storage/app/public/dishes`.
- Lier le stockage public :

```
php artisan storage:link
```

- Validation des fichiers : `image|mimes:jpeg,png,webp|max:2048`.
- **Documentation** : [File Storage](#)

---

## 6) Rôles & Permissions (Spatie)

- Installer la dépendance :

```
composer require spatie/laravel-permission
php artisan vendor:publish --
provider="Spatie\Permission\PermissionServiceProvider"
php artisan migrate
```

- Ajouter le **trait** `HasRoles` dans le modèle `User`.
  - Créer les permissions : `create dishes`, `delete dishes`.
  - Créer un rôle `admin` et lui attribuer ces permissions.
  - Associer un **user admin** par défaut via un seeder.
  - Restreindre les routes avec `->middleware('permission:create dishes')` ou `@can` dans les vues.
  - **Documentation** : [Spatie Laravel Permission](#)
-

## 7) Contrôleurs & Routage

- Créer le contrôleur :

```
php artisan make:controller DishController
```

- Méthodes :

- `index()` : liste paginée des plats.
  - `show()` : afficher un plat en lecture seule.
  - `store()` : création (admin uniquement + envoi de notification).
  - `update()` : modification (admin uniquement).
  - `destroy()` : suppression (admin uniquement).
  - Routes définies dans `web.php` et protégées par `auth` + `permission`.
  - **Documentation** : [Routing](#)
- 

## 8) Notifications par email

- Configurer le driver mail dans `.env` (Mailtrap ou Mailhog en dev).
- Générer une notification :

```
php artisan make:notification DishPublished
```

- Envoyer la notif lors de la création d'un plat : `auth()->user()->notify(new DishPublished($dish));`
  - **Documentation** : [Notifications](#)
- 

## 9) Seeders & Données de test

- Installer Faker pour plats et images :

```
composer require jzonta/faker-restaurant  
composer require xvladqt/faker-lorem-flickr
```

- Créer un `DishFactory` pour générer :
    - noms de plats réalistes (faker-restaurant)
    - images aléatoires (lorem-flickr)
    - recettes chiffrées
  - Créer un `DatabaseSeeder` pour :
    - ajouter un user admin + rôles/permissions
    - générer plusieurs users classiques
    - générer des plats + favoris aléatoires
  - **Documentation** : [Seeding](#)
-

## 10) Frontend (HTML + CSS sans framework)

- Organisation des vues Blade :
    - `layout.blade.php` (header, footer, nav)
    - `home.blade.php` (liste paginée des plats)
    - `dishes/_form.blade.php` (formulaire partagé ajout/modif)
    - `dishes/show.blade.php` (affichage détail)
  - **Login / Signin** : réécriture des vues générées par Laravel UI (sans Bootstrap).
  - **Accueil** :
    - Grille CSS (cartes avec image, nom, créateur, boutons actions).
    - Pagination custom (sans Tailwind, boutons classiques).
  - **Popup Ajout / Modif** : utilisation de `<dialog>` natif ou d'un overlay CSS/JS léger.
  - Respect du design sobre, moderne, uniquement avec CSS.
  - **Documentation** : [Blade Templates](#)
- 

## 11) Tests (optionnel mais recommandé)

- Créer des tests pour vérifier :
    - Auth obligatoire.
    - Création plat (admin OK, user simple KO).
    - Ajout/retrait des favoris.
    - Chiffrement/déchiffrement de la recette.
  - **Documentation** : [Testing](#)
- 

## 12) Estimation du temps

- Setup + Auth : **0,5 j**
- Migrations & modèles : **0,5 j**
- Chiffrement : **0,25 j**
- Rôles & permissions : **0,5 j**
- Upload images : **0,25 j**
- Notifications : **0,25 j**
- Seeders : **0,25 j**
- Front (HTML/CSS, vues) : **0,75 à 1 j**
- Tests & finitions : **0,25 j**

**Total estimé : ~3,5 à 4 jours effectifs**

---

✓ Ce plan servira de guide clair et progressif pour ton correcteur et pour toi lors du développement.