



# DOSSIER PROFESSIONNEL (DP)

<i>Nom de naissance</i>	▶ MOREIRA
<i>Nom d'usage</i>	▶ MOREIRA
<i>Prénom</i>	▶ Stéphane
<i>Adresse</i>	▶ 2 Place des arcades 06250 Mougins

## Titre professionnel visé

Développeur Web et Web Mobile

### MODALITÉ D'ACCÈS :

- ☒ Parcours de formation
- ☐ Validation des Acquis de l'Expérience (VAE)

# DOSSIER PROFESSIONNEL <sup>(DP)</sup>

## Présentation du dossier

Le dossier professionnel (DP) constitue un élément du système de validation du titre professionnel.  
**Ce titre est délivré par le Ministère chargé de l'emploi.**

Le DP appartient au candidat. Il le conserve, l'actualise durant son parcours et le présente **obligatoirement à chaque session d'examen.**

Pour rédiger le DP, le candidat peut être aidé par un formateur ou par un accompagnateur VAE.

Il est consulté par le jury au moment de la session d'examen.

### Pour prendre sa décision, le jury dispose :

1. des résultats de la mise en situation professionnelle complétés, éventuellement, du questionnaire professionnel ou de l'entretien professionnel ou de l'entretien technique ou du questionnement à partir de productions.
2. du **Dossier Professionnel** (DP) dans lequel le candidat a consigné les preuves de sa pratique professionnelle.
3. des résultats des évaluations passées en cours de formation lorsque le candidat évalué est issu d'un parcours de formation
4. de l'entretien final (dans le cadre de la session titre).

*[Arrêté du 22 décembre 2015, relatif aux conditions de délivrance des titres professionnels du ministère chargé de l'Emploi]*

### Ce dossier comporte :

- ▶ pour chaque activité-type du titre visé, un à trois exemples de pratique professionnelle ;
- ▶ un tableau à renseigner si le candidat souhaite porter à la connaissance du jury la détention d'un titre, d'un diplôme, d'un certificat de qualification professionnelle (CQP) ou des attestations de formation ;
- ▶ une déclaration sur l'honneur à compléter et à signer ;
- ▶ des documents illustrant la pratique professionnelle du candidat (facultatif)
- ▶ des annexes, si nécessaire.

*Pour compléter ce dossier, le candidat dispose d'un site web en accès libre sur le site.*

# DOSSIER PROFESSIONNEL <sup>(DP)</sup>



<http://travail-emploi.gouv.fr/titres-professionnels>

## Sommaire

### Exemples de pratique professionnelle

<b>Développer la partie front-end d'une application web ou web mobile sécurisée</b>	<b>p.</b>	<b>5</b>
▶ Installer et configurer son environnement de travail en fonction du projet web ou web mobile	p.	5
▶ Maquetter des interfaces utilisateur web ou web mobile	p.	6
▶ Réaliser des interfaces utilisateur statiques web ou web mobile	p.	7
▶ Développer la partie dynamique des interfaces utilisateur web ou web mobile	p.	8
<b>Développer la partie back-end d'une application web ou web mobile sécurisé</b>	<b>p.</b>	<b>9</b>
▶ Mettre en place une base de données relationnelle	p.	9
▶ Développer des composants d'accès aux données SQL et NoSQL	p.	10
▶ Développer des composants métier coté serveur	p.	11
▶ Documenter le déploiement d'une application dynamique web ou web mobile		12
<b>Titres, diplômes, CQP, attestations de formation</b> <i>(facultatif)</i>	<b>p.</b>	<b>13</b>
<b>Déclaration sur l'honneur</b>	<b>p.</b>	<b>14</b>
<b>Documents illustrant la pratique professionnelle</b> <i>(facultatif)</i>	<b>p.</b>	<b>15</b>
<b>Annexes</b> <i>(Si le RC le prévoit)</i>	<b>p.</b>	<b>16</b>

# **EXEMPLES DE PRATIQUE PROFESSIONNELLE**

# DOSSIER PROFESSIONNEL <sup>(DP)</sup>

## Activité-type 1 Développer la partie front-end d'une application web ou web mobile sécurisée

Exemple n°1 ► *Installer et configurer son environnement de travail en fonction du projet web ou web mobile*

### 1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Les stacks techniques que j'ai choisis pour mon projet sont:

- Backend : Laravel 12.13.0 (PHP ^8.2)
- Frontend : React 19.0, Inertia.js 2.0.2, JavaScript.
- Styles : TailwindCSS 4.0, DaisyUI 5.0.28, Framer Motion
- Base de données : SQLite
- Administration : Filament v3.3.14
- Build : Vite 6.0

Suite a ma veille , j'ai opté pour un projet monolithique avec Laravel. Étant plus a l'aise avec React et souhaitant avoir une SinglePageApplication, j'ai fait le choix de combiner Laravel avec Inertia pour pouvoir utiliser React.

Filament va me permettre de créer un backoffice très simplement et rapidement.

### 2. Précisez les moyens utilisés :

IDE : VisualStudioCode

Versionning et gestion de projet : Git GitHub GitHubProject

Prérequis:

- PHP 8.2
- Composer
- Node.js 20
- SQLITE

Pour Laravel + Inertia , j'ai choisis d'utiliser un starter déjà configuré en JSX (et non TSX):

<https://medium.com/@yazidkhaldi/convertng-laravels-react-starter-kit-from-typescript-to-javascript-1691d38e5d3d>

TailwindCSS est inclus dans le starterkit.

DaisyUI : a la racine de mon projet bien initialisé, j'ai utilisé la commande

```
npm i -D daisyui@latest
```

## DOSSIER PROFESSIONNEL <sup>(DP)</sup>

### 3. Avec qui avez-vous travaillé ?

Seul

### 4. Contexte

Nom de l'entreprise, organisme ou association ► Ecole LaPlateforme.io

Chantier, atelier, service ► Projet E-Commerce

Période d'exercice ► Du : 01/04/2025 au : 15/07/2025

### 5. Informations complémentaires *(facultatif)*

# DOSSIER PROFESSIONNEL (DP)

## Exemple n°2 ► Maquetter des interfaces utilisateur web ou web mobile

### 1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

J'ai effectué une petite veille technologique afin de déterminer les objectifs fonctionnels de base pour un e-commerce. J'ai donc visité les sites des principaux acteurs français afin de déterminer fonctionnalités incontournables du secteur, j'ai réalisé un moodboard et un wireframe afin de prévisualiser l'apparence de mon site, ne sachant pas complètement quelle technologie j'allais utiliser pour mettre en page mon site. Lorsque mes stacks techniques ont été validés, j'ai effectué une maquette rapide de ma page d'accueil pour desktop

### 2. Précisez les moyens utilisés :

- Figma pour le wireframe / maquette de la page d'accueil.
- Excalidraw pour la prise d'informations.
- Principaux acteurs français: Zalando, Sarenza, Showroomprivé etc..

### 3. Avec qui avez-vous travaillé ?

Seul

### 4. Contexte

Nom de l'entreprise, organisme ou association ► Ecole LaPlateforme.io

Chantier, atelier, service ► Projet E-Commerce

Période d'exercice ► Du : 01/04/2025 au : 15/07/2025

### 5. Informations complémentaires (facultatif)

# DOSSIER PROFESSIONNEL (DP)

## Exemple n°3 ► Réaliser des interfaces utilisateur statiques web ou web mobile

### 1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

J'ai réalisé plusieurs page statique pour mon site: CGU, Politique de cookies , page de contacte, etc...  
Pour cet exemple , j'ai choisis de vous présenter ma page «A propos»

### 2. Précisez les moyens utilisés :

Via React , j'ai d'abord créé une constante 'section' afin pour que mon code soit scalable et maintenable rapidement et facilement.

```
const sections = [
  {
    title: "1. Notre histoire",
    content: (
      <p>
        Piwee! a été créé pour offrir une expérience de shopping unique aux amoureux de mode. Passionnés par l'innovation et le
      </p>
    ),
  },
  {
    title: "2. Notre mission",
    content: (
      <p>
```

*Début de mon tableau 'section'*

J'ai ensuite mis en place une boucle pour afficher celui-ci:

```
<div className="">
  {sections.map((section) => (
    <div>
      <h2 className="">{section.title}</h2>
      {section.content}
    </div>
  ))}
</div>
```

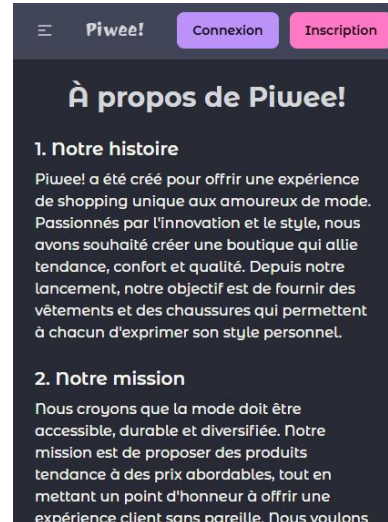


# DOSSIER PROFESSIONNEL (DP)

Et enfin : mise en page avec TailwindCSS et DaisyUI , puis petites animations avec motion:

```
export default function About() {
  return (
    <Layout className="min-h-screen">
      <Head title="À propos" />
      <motion.section
        className="max-w-4xl mx-auto p-6 bg-base-100 shadow-lg my-10"
        initial={{ opacity: 0, y: 30 }}
        animate={{ opacity: 1, y: 0 }}
        transition={{ duration: 0.6, ease: "easeOut" }}
      >
        <h1 className="text-3xl font-bold mb-6 text-center">À propos de Piwee!</h1>

        <div className="space-y-6 text-base-content">
          {sections.map((section, index) => (
            <motion.div
              key={index}
              initial={{ opacity: 0, y: 20 }}
              animate={{ opacity: 1, y: 0 }}
              transition={{ duration: 0.5, delay: 0.2 + index * 0.1 }}
            >
              <h2 className="text-xl font-semibold mb-2">{section.title}</h2>
              <div>{section.content}</div>
            </motion.div>
          ))}
        </div>
      </motion.section>
    </Layout>
  );
}
```



## 3. Avec qui avez-vous travaillé ?

Seul

## 4. Contexte

Nom de l'entreprise, organisme ou association ► Ecole LaPlateforme.io

Chantier, atelier, service ► Projet E-Commerce

Période d'exercice ► Du : 01/04/2025 au : 15/07/2025

## 5. Informations complémentaires (facultatif)

# DOSSIER PROFESSIONNEL (DP)

## Exemple n°4 ► Développer la partie dynamique des interfaces utilisateur web ou web mobile

### 1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Lorsqu'un utilisateur clique sur un article, il est redirigé vers la page des détails de l'article en question. J'ai donc mis en place un système de gestion dynamique des tailles et des quantités à renvoyer au panier en fonction des stocks.

### 2. Précisez les moyens utilisés :

J'ai utilisé `useState()` pour initialiser des états dynamiques:

```
const [selectedSize, setSelectedSize] = useState(item.sizes[0]?.size || "");
const [quantity, setQuantity] = useState(1);
const selectedStock = item.sizes.find(s => s.size === selectedSize)?.stock || 0;
const isQuantityTooHigh = quantity > selectedStock;
```

On instancie deux états locaux pour gérer la taille sélectionnée (`selectedSize`) et la quantité (`quantity`).

L'init de `selectedSize` propose par défaut la première taille disponible (UX).

`selectedStock` calcule dynamiquement le stock disponible pour la taille choisie, via un simple `find()` dans le tableau des tailles.

`isQuantityTooHigh` sert à afficher un message d'erreur en cas de dépassement du stock.

Pour la sélection dynamique des tailles : mapping, animation et accessibilité

```
{item.sizes.map(({ size }) => (
  <motion.p
    key={size}
    whileHover={{ scale: 1.05 }}
    className={`badge px-2.5 place-self-center cursor-pointer ${
      selectedSize === size ? "badge-primary" : "badge-outline"
    }`}
    variants={{
      hidden: { opacity: 0, y: 10 },
      visible: { opacity: 1, y: 0 },
    }}
    onClick={() => setSelectedSize(size)}
  >
    {size}
  </motion.p>
)}
```

On affiche dynamiquement toutes les tailles disponibles pour le produit, sous forme de badges interactifs.

On utilise les capacités d'animation de Motion (`motion.p`, variante `visible/hidden` et effet au survol) pour améliorer l'UX.

Le visuel du badge reflète la `selectedSize` via la classe CSS (`badge-XXX`).

## DOSSIER PROFESSIONNEL (DP)

Voici le rendu pour la gestion de la quantité:

```
<motion.button
  whileHover={{ scale: 1.1 }}
  className="btn btn-primary"
  onClick={() => setQuantity(q => Math.max(1, q - 1))}
>
-
</motion.button>
<p className="place-self-center">{quantity}</p>
<motion.button
  whileHover={{ scale: 1.1 }}
  className="btn btn-primary"
  onClick={() => setQuantity(q => q + 1)}
>
+
</motion.button>
```

*L'utilisateur ajuste la quantité via deux boutons (+/-), la décrémentation ne descend jamais en dessous de 1 (sécurité UX).*

*Animation sur les boutons pour inviter l'action.*

*Affichage du nombre sélectionné au centre, mise à jour instantanée.*

Et la le système d'alerte et validation du stock:

```
{isQuantityTooHigh && (
  <div className="text-error text-center mb-2">
    La quantité demandée dépasse le stock disponible ({selectedStock}).
  </div>
)}
```

*Dès que l'utilisateur essaie de dépasser le stock, un message s'affiche en temps réel, évitant de mauvaises surprises au checkout.*

*Validation immédiate côté front sans attendre une réponse serveur.*

### 3. Avec qui avez-vous travaillé ?

Seul

### 4. Contexte

Nom de l'entreprise, organisme ou association ► Ecole LaPlateforme.io

Chantier, atelier, service ► Projet E-Commerce

Période d'exercice ► Du : 01/04/2025 au : 15/07/2025

## Activité-type 2 Développer la partie back-end d'une application web ou web mobile sécurisé

### Exemple n° 1 ► Mettre en place une base de données relationnelle

#### 1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Conception de la base de données avec MERISE. Mise en place de la base de données en SQLite grâce aux migrations et seeders Laravel

#### 2. Précisez les moyens utilisés :

J'ai d'abord conceptualisé ma base de données avec la méthode Merise (dictionnaire non-technique et technique, MCD, MLD et MPD, voir annexes).

Ensuite , on met en place notre structure avec les migrations. Celles ci nous permettent de versionnée et automatisée nos données, ce qui permet de faire évoluer le schéma sans perte de données (et de collaborer avec le même jeu de données lors de travail en équipe.

```
Schema::create('items', function (Blueprint $table) {
    $table->id();
    $table->foreignId('brand_id')->constrained('brands')->cascadeOnDelete();
    $table->foreignId('category_id')->constrained('categories')->cascadeOnDelete();
    $table->string('name');
    $table->string('slug');
    $table->string('description');
    $table->unsignedInteger('price');
    $table->boolean('isDeleted');
    $table->timestamps();
});
```

*Création de la table items*

'Schema' est la façade qui va nous permettre de gérer nos structures de tables. Grâce a la fonction create(),a qui on passe en paramètre le nom de la table, puis avec la syntaxe proposée sur la documentation nous pouvons spécifier les nom des champs ainsi que leurs type.

Pour les champs brand\_id et category\_id, j'ai spécifié que ce sont des clefs étrangères, et ajouté la contrainte avec la table concernée ainsi que le fait que si une catégorie ou une marque est supprimée, les articles liés sont aussi supprimés (suppression en cascade)

#### 3. Avec qui avez-vous travaillé ?

Seul

# DOSSIER PROFESSIONNEL <sup>(DP)</sup>

## 4. Contexte

Nom de l'entreprise, organisme ou association ► Ecole LaPlateforme.io

Chantier, atelier, service ► Projet E-Commerce

Période d'exercice ► Du : 01/04/2025 au : 15/07/2025

## 5. Informations complémentaires *(facultatif)*

# DOSSIER PROFESSIONNEL (DP)

## Exemple n° 2 ► Développer des composants d'accès aux données SQL et NoSQL

### 1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Après avoir créé les modèles Eloquent, j'ai pu créer la plupart de mes CRUD grâce à Filament. J'ai quand même utilisé Eloquent pour les composants métiers user.

### 2. Précisez les moyens utilisés :

#### Eloquent ORM

Syntaxe orientée objet expressive.

Avantages: Évite les boilerplates\* et les injections SQL.

J'utilise Eloquent lorsque Filament n'est pas pertinent (pour l'inscription par exemple).

```
public function registerPost(Request $request): RedirectResponse
{
    $request->validate([
        'firstname' => 'required|string|max:50',
        'lastname' => 'required|string|max:50',
        'birthday' => 'required|date|before:today',
        'gender' => 'required|string|in:homme,femme,autre',
        'address' => 'required|string|max:255',
        'zip' => 'required|string|max:50',
        'city' => 'required|string|max:50',
        'phone' => 'required|string|max:20',
        'email' => 'required|string|lowercase|email|max:255|unique:'.User::class,
        'password' => ['required', 'confirmed', Rules\Password::defaults()],
    ]);

    $user = User::create([
        'firstname' => $request->firstname,
        'lastname' => $request->lastname,
        'birthday' => $request->birthday,
        'gender' => $request->gender,
        'address' => $request->address,
        'zip' => $request->zip,
        'city' => $request->city,
        'phone' => $request->phone,
        'email' => $request->email,
        'password' => Hash::make($request->password),
        'verified' => false, // Valeur par défaut
        'role' => 'user', // Valeur par défaut
        'is_deleted' => false, // Valeur par défaut
    ]);
    event(new Registered($user));
    Auth::login($user);
    return redirect()->route('auth.login')->with('success', 'Votre compte a bien été créé');
}
```

Plutôt présenter la fonction pour les favoris, et remettre en place l'auth native laravel. Présenter un modèle????

# DOSSIER PROFESSIONNEL (DP)

## Filament

Permet la création de panel d'administration complet.

Grâce aux ressources Filament, on peut très rapidement créer un CRUD sur la base d'un modèle Laravel.

Voici un exemple pour la table Item:

```
class ItemResource extends Resource
{
    protected static ?string $model = Item::class;
```

On crée l'objet «ItemRessource» et on y récupère le modèle Item existant.

### Create/Update

```
public static function form(Form $form): Form
```

La fonction form correctement utilisée initialise le formulaire de création et de modification !

On va donc y déclarer les inputs (soit les champs de notre table item)

```
Forms\Components\TextInput::make('price')
->numeric()
->prefix('€')
->maxValue(42949672.95)
->required(),
```

Voici un exemple concret avec le champ «price». Ce bout de code va créer un input de type texte. On peut ajouter des paramètres à celui-ci afin de limiter l'utilisateur pour qu'il entre des données valides pour notre base de données, rendre le champ obligatoire, etc

```
Forms\Components\Select::make('brand_id')
->relationship('brand', 'name')
->searchable()
->preload()
->createOptionForm([
    Forms\Components\TextInput::make('name')
        ->label('Nom de la marque')
        ->required()
        ->maxLength(50),
])
->required(),
```

Ici un exemple d'une clé étrangère avec la marque. ->relationship() avec en paramètre la table et le champ en relation avec notre champ «brand\_id»

->searchable() limite l'administrateur à utiliser une entrée existante dans la table «brand», ou à en créer une à la volée grâce à ->createOptionForm



# DOSSIER PROFESSIONNEL (DP)

Items > Créer

Créer Item

Brand\*

Sélectionnez une option

+

Category\*

Sélectionnez une option

+

Nom\*

Description\*

Price\*

€

Créer

Créer & Ajouter un autre

Annuler

Items > Modifier

Modifier Item

Brand\*

Nike

×

+

Category\*

T-shirt

×

+

Nom\*

StreetTone Classic

Description\*

Un t-shirt au style urbain intemporel, parfait pour afficher un

Price\*

€

19,99

Marquer comme supprimé

Sauvegarder

Annuler

Supprimer

Voici le rendu!

## Read




```
public static function table(Table $table): Table
{
    return $table
        ->columns([
            Tables\Columns\TextColumn::make('name')
                ->tooltip(fn ($record) => $record->name)
                ->limit(30)
                ->searchable(),
        ])
```

J'utilise la fonction table, pour lire les données, qui seront automatiquement mis en page avec le panel filament

Items > Liste

Items

Rechercher

Name	Image	Price	Brand	Category	Description	Tailles disponibles	
StreetTone Classic		19.99	Nike	T-shirt	Un t-shirt au style urbain int...	M L	Modifier Supprimer
Drip Motion Tree		24.99	Salomon	T-shirt	Exprime ton flow avec ce t-shi...	L S	Modifier Supprimer
Pulse Fade		29.99	Salomon	T-shirt	Un dégradé moderne qui capte l...	M L XL	Modifier Supprimer

Le hard delete se met en place automatiquement, et j'ai implémenté un soft delete afin de pouvoir garder une traçabilité dans ma base de donnée

### 3. Avec qui avez-vous travaillé ?

Seul



---

# DOSSIER PROFESSIONNEL <sup>(DP)</sup>

---

## 4. Contexte

Nom de l'entreprise, organisme ou association ► Ecole LaPlateforme.io

Chantier, atelier, service ► Projet E-Commerce

Période d'exercice ► Du : 01/04/2025 au : 15/07/2025

## 5. Informations complémentaires *(facultatif)*

# DOSSIER PROFESSIONNEL (DP)

## Exemple n° 3 ► Développer des composants métier coté serveur

### 1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

J'ai développé plusieurs composants métier et j'ai choisi de vous présenter la gestion de mes articles par un administrateur grâce à Filament. Son rôle sera de:

- Permettre la création et la modification rapide des articles via un formulaire ergonomique.
- Gérer les liaisons essentielles : chaque article est relié à une marque, une catégorie, des images et des stocks (par taille).
- Superviser le catalogue : visualiser, filtrer, éditer et supprimer les articles selon les besoins.
- Assurer la gestion du cycle de vie de l'article (activation, suppression logique, réactivation).

### 2. Précisez les moyens utilisés :

J'ai mis en place le modèle Eloquant en paramétrant les fillables. C'est aussi ici que j'ai mis en place le système de création de slug:

```
protected static function booted()
{
    static::saving(function ($item) {
        if (!empty($item->name)) {
            $item->slug = Str::slug($item->name);
        }
    });
}
```

L'événement saving se déclenche avant que l'enregistrement ne soit sauvegardé en base de données, que ce soit lors d'une création (create) ou d'une mise à jour (update).

et défini les relations avec les autres entités:

```
public function brand(): BelongsTo
{
    return $this->belongsTo(Brand::class);
}
public function category(): BelongsTo
{
    return $this->belongsTo(Category::class);
}
```

Un article appartient à une marque et une catégorie

```
public function images(): HasMany
{
    return $this->hasMany(Image::class);
}
public function stocks()
{
    return $this->hasMany(Stock::class);
}
```

Un article peut avoir plusieurs images et plusieurs variantes/tailles en stock

```
public function favorites()
{
    return $this->hasMany(Favorite::class);
}
public function favoritedBy()
{
    return $this->belongsToMany(User::class, 'favorites');
}
```

Relations pour gérer les favoris

belongToMany(): Chaque entité peut être liée à plusieurs autres entités, et réciproquement.

# DOSSIER PROFESSIONNEL (DP)

Puis on a centralisé toute la logique métier dans une ressource filament :  
La ressource ItemResource regroupe toute la logique d'administration des articles :

- Formulaire de création/édition (vu précédemment )
- Tableau de bord (listing, tris, filtres)
- Actions personnalisées (suppression logique, réactivation)
- Gestion des relations (images, stocks)

## Gestion des images d'un article

ImageRelationManager.php va nous permettre de customiser notre composant Image:

```
class ImagesRelationManager extends RelationManager
{
    protected static string $relationship = 'images';
}
```

Dans notre nouvelle classe , on appelle la methode image() de notre ressource

```
public function form(Forms\Form $form): Forms\Form
{
    return $form->schema([
        Forms\Components\FileUpload::make('url')
            ->disk('public')
            ->directory('img')
            ->image()
            ->required(),
    ]);
}
```

On crée un champ d'upload pour le champ url de la base de données, en spécifiant le disk (défini dans config/filesystems.php) et le dossier dans lequel sera enregistré le fichier. On limite le type de fichier a des images et on rend le champ obligatoire

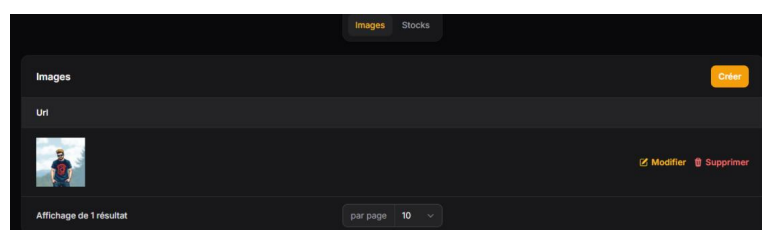
```
public function table(Tables\Table $table): Tables\Table
{
    return $table
        ->columns([Tables\Columns\ImageColumn::make('url')->disk('public')->size(80),])
        ->headerActions([CreateAction::make(),])
        ->actions([Tables\Actions\EditAction::make(), Tables\Actions\DeleteAction::make(),]);
}
```

Pour l'affichage:

->columns() : on recupere l'url de notre bdd, on cherche dans le disk public et on donne une taille de 80 a nos miniatures

->headerActions().... Nous permet d'ajouter un bouton de création d'image en haut du composant et ->actions permet d'ajouter des boutons de modification et de suppression à chaque ligne du composant

Le resultat:



## DOSSIER PROFESSIONNEL <sup>(DP)</sup>

### 3. Avec qui avez-vous travaillé ?

Seul

### 4. Contexte

Nom de l'entreprise, organisme ou association ► Ecole LaPlateforme.io

Chantier, atelier, service ► Projet E-Commerce

Période d'exercice ► Du : 01/04/2025 au : 15/07/2025

### 5. Informations complémentaires *(facultatif)*

# DOSSIER PROFESSIONNEL <sup>(DP)</sup>

**Exemple n° 4 ► Documenter le déploiement d'une application dynamique web ou web mobile**

## 1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

J'ai documenté le déploiement en local dans un fichier README.

## 2. Précisez les moyens utilisés :

Tout au long du développement, j'ai pris des notes sur la mise en place de mon environnement , de mes dépendances et de ma base de données.  
Vous trouverez mon README complet en annexe.

## 3. Avec qui avez-vous travaillé ?

Seul

## 4. Contexte

Nom de l'entreprise, organisme ou association ► Ecole LaPlateforme.io

Chantier, atelier, service ► Projet E-Commerce

Période d'exercice ► Du : 01/04/2025 au : 15/07/2025

## 5. Informations complémentaires (facultatif)

---

## DOSSIER PROFESSIONNEL <sup>(DP)</sup>

---

### Titres, diplômes, CQP, attestations de formation

*(facultatif)*

Intitulé	Autorité ou organisme	Date
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.

---

## DOSSIER PROFESSIONNEL <sup>(DP)</sup>

---

### Déclaration sur l'honneur

---

Je soussigné(e) Stéphane MOREIRA ,  
déclare sur l'honneur que les renseignements fournis dans ce dossier sont exacts et que je suis  
l'auteur(e) des réalisations jointes.

Fait à Cannes le 20/08/2025

pour faire valoir ce que de droit.

Signature :

---

## DOSSIER PROFESSIONNEL <sup>(DP)</sup>

---

### Documents illustrant la pratique professionnelle

*(facultatif)*

Intitulé
Cliquez ici pour taper du texte.



---

# DOSSIER PROFESSIONNEL <sup>(DP)</sup>

---

## ANNEXES

---

*(Si le RC le prévoit)*  
*Methode MERISE (MCD MLD MPD Dictionnaire)*  
*README.pdf*