



MINISTÈRE CHARGÉ  
DE L'EMPLOI

# DOSSIER PROFESSIONNEL (DP)

*Nom de naissance*

▶ Dos Ramos Franco

*Nom d'usage*

▶

*Prénom*

▶ Octavio

*Adresse*

▶ 25 Bd Gambetta

## Titre professionnel visé

Développeur Web & Web Mobile

### MODALITE D'ACCES :

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

# DOSSIER PROFESSIONNEL (DP)

## 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.*



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

# DOSSIER PROFESSIONNEL (DP)

## Sommaire

### Exemples de pratique professionnelle

**Intitulé de l'activité-type n° 1 Développer la partie front-end d'une application web ou web mobile** p.6

- ▶ Intitulé de l'exemple n° 1 : Maquettage & structure de navigation (UX / UI) p.6
- ▶ Intitulé de l'exemple n° 2 :Réalisation d'une interface statique & adaptable (React + Tailwind) p.8
- ▶ Intitulé de l'exemple n° 3 Développement d'une interface dynamique (animations, feedback, conformité p.10

**Intitulé de l'activité-type n° 2 Développer la partie back-end d'une application web ou web mobile.** p.12

- ▶ Intitulé de l'exemple n° 1 Composants d'accès aux données (modélisation → migrations .→ORM p.12
- ▶ Intitulé de l'exemple n° 2 Logique métier & sécurité (authentification admin, formulaires, notifications) p.14
- ▶ Intitulé de l'exemple n° 3 Composants de gestion de contenu & préparation à la production (Filament + Plesk). p.16

**Intitulé de l'activité-type n° 3 Compétences transversales : veille, sécurité.** p.18

- ▶ Intitulé de l'exemple n° 1 Veille sécurité (vulnérabilités,mitigation) p.18
- ▶ Intitulé de l'exemple n° 2 Démarche qualité : tests, accessibilité & performances p.20
- ▶ Intitulé de l'exemple n° 3 Démarche qualité : tests, accessibilité & performances p.22

**Intitulé de l'activité-type n° 4 Développer la partie front-end d'une application web ou web mobile** p.24

- ▶ Intitulé de l'exemple n° 1 Administration type CMS avec Filament (projets & réponses clients) p.24
- ▶ Intitulé de l'exemple n° 2 Mise en production & exploitation (Plesk, SSL, sauvegardes, monitoring) p.26
- ▶ Intitulé de l'exemple n° 3 Organisation & outillage (design, versionning, qualité) p.28

**Titres, diplômes, CQP, attestations de formation (facultatif)**

p. \_\_\_\_\_

**Déclaration sur l'honneur**

p.31 \_\_\_\_\_

**Documents illustrant la pratique professionnelle (facultatif)**

p. \_\_\_\_\_

**Annexes (Si le RC le prévoit)**

p.33 \_\_\_\_\_

# DOSSIER PROFESSIONNEL (DP)

# **EXEMPLES DE PRATIQUE PROFESSIONNELLE**

# DOSSIER PROFESSIONNEL (DP)

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

*Exemple n°1* ► Maquettage & structure de navigation (UX / UI)

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

Je définis l'architecture de l'information et les parcours des deux cibles (recruteur / client).

- Arborescence & parcours : Accueil → Home → (À propos | Tous les projets → Détail projet | Galerie | Contact | Questionnaire client) + pages légales & Accessibilité.
- User stories & critères d'acceptation :
  - En tant que recruteur, je veux voir les projets et les technos en < 2 clics → bouton "Tous les projets" en home + filtres clairs.
  - En tant que client, je veux expliquer mon besoin → "Questionnaire client" accessible depuis le menu et la page Contact.
- Wireframes → Maquettes : low-fi (position des blocs, hiérarchie visuelle) puis écrans haute fidélité (grille, échelles d'espacement, tailles de typo).
- Design system de base : palette, styles typographiques, composants réutilisables (Header/Nav, Footer, Card Projet, Boutons, Form Inputs).

### 2. Précisez les moyens utilisés :

**Figma** pour les maquettes (wireframes, prototypage, écrans haute fidélité), grille 12 colonnes / 4 breakpoints (sm/md/lg/xl), checklist d'accessibilité, charte Tailwind (échelles de spacing, radius, typographies) pour que le design corresponde aux tokens utilisés au build.

# DOSSIER PROFESSIONNEL (DP)

## 3. Avec qui avez-vous travaillé ?

Travail réalisé en autonomie totale sur la lisibilité et la cohérence des parcours.

## 4. Contexte

Nom de l'entreprise, organisme ou association ► **La Plateforme CDPI**

Chantier, atelier, service ► **Formation Développeur Web & Web Mobile**  
Période d'exercice ► **Du 24/06/2024 au 15/10/2025**

## 5. Informations complémentaires (*facultatif*)

Pré-étude et cadrage UX/UI du portfolio (phase amont), objectifs : clarté, rapidité d'accès à l'info, préparation d'une intégration réactive côté front.

J'intègre dès la maquette les obligations légales (mentions, confidentialité, bannière cookies) et une page Accessibilité, afin d'éviter des réorganisations tardives.

Évidences annexer (reco.) : capture de l'arborescence, quelques écrans Figma (home, liste projets, détail projet), extrait de la checklist accessibilité.

Couverture RC AT1 –Maquetter une application : parcours/US, wireframes → maquettes, design system minimal, prise en compte des contraintes d'accessibilité dès la conception.

# DOSSIER PROFESSIONNEL (DP)

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

*Exemple n°2*

- ▶ **Réalisation d'une interface statique & adaptable (React + Tailwind)**

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

Je transforme les maquettes en pages responsives, structurées et réutilisables.

#### Découpage en composants :

- **Layout** (header/nav, contenu, footer).

- **Nav responsive** (menu burger, focus visible, rôle ARIA, liens vers légales/Accessibilité).

- **Cards projets** (vignette, techno, résumé, CTA “Voir le projet”).

- **Pages : Accueil** (entrée vidéo/CTA), Home (sections synthétiques), À propos, Tous les projets, Détail projet, Galerie (lazy), Contact, Login, Accessibilité, Mentions, Politique.

- **Layout** (header/nav, contenu, footer).

Mise en page avec **Tailwind** : grilles, spacing cohérent, typographies, container centré, gestion des colonnes qui se replient sur mobile (sm→md→lg→xl).

**Formulaire Contact (côté UI)** : champs, labels, messages d'erreur, état “disabled” sur soumission, gestion du focus sur l'erreur en haut du formulaire.

**Tests d'adaptabilité** : largeurs 320 / 375 / 768 / 1024 / 1280 px ; iOS/Android (émulateurs), Chrome/Firefox/Edge.

### 2. Précisez les moyens utilisés :

React (components/routing), Tailwind (classes utilitaires, responsive), Vite (dev server & build), Git/GitHub (branches par fonctionnalité), VS Code (formatage/linting).

# DOSSIER PROFESSIONNEL (DP)

## 3. Avec qui avez-vous travaillé ?

Réalisation en autonomie total sur la qualité de l'intégration et la cohérence responsive

## 4. Contexte

Nom de l'entreprise, organisme ou association ► **La Plateforme CDPI**

Chantier, atelier, service ► **Formation Développeur Web & Web Mobile**  
Période d'exercice ► **Du 24/06/2024 au 15/10/2025**

## 5. Informations complémentaires (*facultatif*)

Phase d'intégration “statique & adaptable” pour satisfaire les critères AT1 (conformité maquette, usage mobile, bonnes pratiques de structuration).

J'ai appliqué les fondamentaux d'accessibilité (contrastes, navigation clavier, ARIA) et préparé une page dédiée “Accessibilité”.

Couverture RC AT1 –Réaliser une interface utilisateur web statique et adaptable : pages conformes à la maquette, structure sémantique/SEO, responsive multi-breakpoints.

# DOSSIER PROFESSIONNEL (DP)

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

*Exemple n°3*

- Développement d'une interface dynamique (animations, feedback, conformité)

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

J'ajoute les comportements dynamiques, les feedbacks et la conformité UX/légale.

#### Transitions & micro-interactions :

- **Framer Motion** : AnimatePresence pour transitions route-à-route, propriétés initial/animate/exit sur sections, easing léger pour conserver la lisibilité.
- **Loader** : écran d'attente au 1er chargement (masqué dès que les assets critiques sont prêts).

#### Pages d'état :

- **Maintenance** (personnage/animation + message clair + lien retour).
- **404** (humour + vidéo légère + CTA d'orientation).

**Bannière cookies** : affichage au 1er passage, acceptation/refus, stockage du choix (localStorage), lien "paramètres" via Politique de confidentialité, respect du focus/la navigation clavier.

#### Feedback & accessibilité :

- Messages d'erreur/succès visibles + aria-live="polite".
- Gestion du focus après navigation (focus sur h1 de la page).
- États de chargement pour les boutons (loading, disabled).

#### Perf côté UI :

- Code-splitting avec React.lazy/Suspense sur pages lourdes (Galerie).
- Lazy-loading d'images par IntersectionObserver.
- Limitation du reflow (dimensions réservées).

### 2. Précisez les moyens utilisés :

React hooks (state/effect), Framer Motion, styled-components (loader), utilitaires d'accessibilité (focus management), API Web Storage (localStorage) pour la préférence cookies.

# DOSSIER PROFESSIONNEL (DP)

## 3. Avec qui avez-vous travaillé ?

Réalisation en autonomie totale sur la pertinence des animations et la conformité des messages d'état.

## 4. Contexte

Nom de l'entreprise, organisme ou association ► **La Plateforme CDPI**

Chantier, atelier, service ► **Formation Développeur Web & Web Mobile**  
Période d'exercice ► **Du 24/06/2024 au 15/10/2025**

## 5. Informations complémentaires (*facultatif*)

Les animations sont calibrées “modérées” pour ne pas dégrader l’accessibilité (durées courtes, pas d’effets agressifs). Les pages d’état servent d’anti-frustration et guident l’utilisateur.

Évidences à annexer (reco.) : GIF/vidéos courtes des transitions/loader, capture de la bannière cookies + préférence mémorisée, extrait de code.

Couverture RC AT1 –Développer une interface utilisateur web dynamique : transitions, feedbacks, pages d’état, conformité cookies côté UI, optimisation mobile.

# DOSSIER PROFESSIONNEL (DP)

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

*Exemple n°1*

- ▶ Composants d'accès aux données (modélisation → migrations → ORM)

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

J'ai conçu le **modèle de données** du portfolio (entités : User, Project, Image, ClientAnswer), puis je l'ai traduit en migrations MySQL avec clés étrangères et contraintes d'unicité (slug projet unique). J'ai indexé les colonnes de recherche (titre, statut, dates) pour accélérer les listes. J'ai ensuite implémenté les modèles Eloquent et les relations : User hasMany Project, Project hasMany Image, Project hasMany ClientAnswer. J'ai prévu des seeders pour charger des données de test (projets, compte admin) et valider rapidement les écrans du dashboard. Cette base soutient le CRUD, les formulaires clients et l'affichage public des projets.

### 2. Précisez les moyens utilisés :

Laravel (migrations, seeders, Eloquent), MySQL (schéma + index), phpMyAdmin pour vérifications locales. J'ai documenté le MCD/MLD/MPD et consigné les scripts de création et de seed en annexe.

# DOSSIER PROFESSIONNEL (DP)

## 3. Avec qui avez-vous travaillé ?

Réalisation en autonomie totale sur la cohérence du/des schéma(s) et pertinence des relations

## 4. Contexte

Nom de l'entreprise, organisme ou association ► **La Plateforme CDPI**

Chantier, atelier, service ► **Formation Développeur Web & Web Mobile**  
Période d'exercice ► **Du 24/06/2024 au 15/10/2025**

## 5. Informations complémentaires (*facultatif*)

J'ai anticipé la nettoyage en cascade (suppression projet → images/réponses liées) et documenté un jeu d'essai pour vérifier l'intégrité (données en entrée / attendues / obtenues).

Évidences à annexer (reco.) : extrait de migrations, schéma relationnel, seeders, captures requêtes/index.

Couverture RC AT2 : Développer les composants d'accès aux données (modèle relationnel, migrations, Eloquent, index).

# DOSSIER PROFESSIONNEL (DP)

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

*Exemple n° 2* ▶ Logique métier & sécurité (authentification admin, formulaires, notifications)

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

J'ai mis en place l'authentification administrateur : création du compte via **Tinker**, hachage bcrypt, et middleware auth sur toutes les routes du **Dashboard**. Tous les formulaires (Contact et Questionnaire client) sont validés côté serveur, persistés (table client\_answers) et notifiés par e-mail (accusé au client + notification admin). J'ai appliqué CSRF sur tous les POST, limité l'exposition des routes et centralisé la validation pour éviter l'injection. Les erreurs sont renvoyées avec messages contextualisés pour l'UX.

### 2. Précisez les moyens utilisés :

**Laravel** (validation, middleware, controllers), Inertia comme pont front/back, système mail configuré avec variables d'environnement, vue de login dédiée, redirection post-login vers le dashboard.

# DOSSIER PROFESSIONNEL (DP)

## 3. Avec qui avez-vous travaillé ?

Réalisation en autonomie totale sur la cohérence sur la sécurité (auth, CSRF), l'ergonomie des écrans admin et la conception des formulaires.

## 4. Contexte

Nom de l'entreprise, organisme ou association ► **La Plateforme CDPI**

Chantier, atelier, service ► **Formation Développeur Web & Web Mobile**  
Période d'exercice ► **Du 24/06/2024 au 15/10/2025**

## 5. Informations complémentaires (*facultatif*)

Pour la traçabilité, j'ai prévu des timestamps et, à terme, l'envoi en file (queue) si le volume de mails augmente. Les secrets et accès (SMTP, DB) sont isolés dans le **.env**.

Évidences à annexer (reco.) : contrôleur de formulaire (validation + persist + mail), capture du login, exemple de **.env** anonymisé.

Couverture RC AT2 : Développer la partie back-end (authentification, validation serveur, CSRF, persistance, notifications).

# DOSSIER PROFESSIONNEL (DP)

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

*Exemple n° 3* ▶ **Composants de gestion de contenu & préparation à la production (Filament + Plesk)**

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

J'ai intégré Filament pour fournir un back-office :

- **Resources** “Projects” et “ClientAnswers” : Formulaires (titre, description, images), Tables (tri, filtres, pagination, actions), Policies pour restreindre l'accès.
- **Médias** : upload multiple d'images, nettoyage cohérent en suppression.
- **UX admin** : listes claires, édition rapide, export des réponses clients.

Côté production (Plesk), j'ai préparé l'environnement : dépôt du code (Git/SFTP), php artisan migrate --force, caches (config:cache, route:cache, view:cache), .env sécurisé, SSL Let's Encrypt et redirection HTTPS. Sauvegardes planifiées et monitoring activés.

### 2. Précisez les moyens utilisés :

Laravel (policies, middlewares), Filament (forms/tables), Composer/NPM (build), Plesk (hébergement, DB, certificats, backups).

# DOSSIER PROFESSIONNEL (DP)

## 3. Avec qui avez-vous travaillé ?

Réalisation en autonomie totale sur la structure du Dashboard et la procédure de déploiement

## 4. Contexte

Nom de l'entreprise, organisme ou association ► La Plateforme CDPI

Chantier, atelier, service ► Formation Développeur Web & Web Mobile

Période d'exercice ► Du 24/06/2024 au 15/10/2025

## 5. Informations complémentaires (*facultatif*)

J'ai documenté une checklist de prod (composer install, build front, migrations, caches, SSL, sauvegardes) et un plan de rollback (restauration Plesk + dump DB).

Évidences à annexer (reco.) : captures Filament (liste/édition), extrait Resource Filament, extrait de procédure Plesk (commands + SSL + backups). documenté une checklist de prod (composer install, build front, migrations, caches, SSL, sauvegardes) et un plan de rollback (restauration Plesk + dump DB).

Couverture RC AT2 : Élaborer et mettre en œuvre des composants dans une application de gestion de contenu (Filament) + préparation à la mise en production.

# DOSSIER PROFESSIONNEL (DP)

## Activité-type 3 : Compétences transversales (veille, qualité/sécurité)

*Exemple n° 1* ► **Veille sécurité (vulnérabilités & mesures de mitigation)**

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

J'ai organisé une veille sécurité continue tout au long du projet afin d'identifier et de traiter les vulnérabilités liées au front (React) et au back (Laravel/MySQL) : XSS, CSRF, injections SQL, exposition de secrets, mauvaise configuration de prod. Pour chaque risque, j'ai listé l'impact, la probabilité, puis j'ai déployé des contre-mesures : tokens CSRF sur tous les formulaires, requêtes préparées/ORM Eloquent, hachage des mots de passe, HTTPS obligatoire et en-têtes de sécurité côté serveur. J'ai intégré cette discipline dans le cycle projet et lié la veille aux fonctionnalités dynamiques et à l'admin (CMS-like). Cette démarche répond explicitement aux exigences du référentiel (veille vulnérabilités à décrire dans le dossier de projet).

### 2. Précisez les moyens utilisés :

Tableau de suivi (menaces → mesures → statut), checklists sécurité Laravel (CSRF, validation, policies), bonnes pratiques d'hébergement Plesk (certificat Let's Encrypt, redirection HTTP→HTTPS, permissions répertoires, sauvegardes et monitoring).

# DOSSIER PROFESSIONNEL (DP)

## 3. Avec qui avez-vous travaillé ?

Réalisation en autonomie totale sur la pertinence des mesures (authentification, formulaires, production).

## 4. Contexte

Nom de l'entreprise, organisme ou association ► **La Plateforme CDPI**

Chantier, atelier, service ► **Formation Développeur Web & Web Mobile**

Période d'exercice ► **Du 24/06/2024 au 15/10/2025**

## 5. Informations complémentaires (*facultatif*)

J'ai documenté la procédure de durcissement (caches Laravel, configuration .env, SSL, sauvegardes) qui accompagne chaque mise à jour, afin de rendre la prod reproductible et sûre.

Couverture RC : description de la veille sur les vulnérabilités et des mesures associées (sécurité front/back, prod).

# DOSSIER PROFESSIONNEL (DP)

## Activité-type 3 : Compétences transversales (veille, qualité/sécurité)

*Exemple n° 2* ▶ **Veille sécurité (vulnérabilités & mesures de mitigation)**

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

Pour résoudre des points précis, j'ai mené des recherches en anglais et appliqué les solutions :

- Animations de navigation : comprendre et calibrer les transitions Framer Motion (patterns AnimatePresence, initial/animate/exit) pour éviter le sur-rendu et préserver l'accessibilité.
- Envoi d'emails : configuration SMTP Laravel en production sur Plesk (MAIL\_HOST, PORT, ENCRYPTION, variables dans .env).

J'ai noté les mots-clés, comparé plusieurs sources (docs officielles, issues), retenu celle la plus fiable/à jour, puis déployé la solution dans le code. Cette pratique correspond à l'attendu du RC (décrire une recherche anglophone, critères, solution mise en œuvre).

### 2. Précisez les moyens utilisés :

Documentation (anglais), notes de test, intégration côté React (transitions) et côté Laravel (mailers + .env). La configuration SMTP et la séparation des environnements sont consignées dans le dossier projet.

# DOSSIER PROFESSIONNEL (DP)

## 3. Avec qui avez-vous travaillé ?

En autonomie totale pour valider la/les solution(s) retenue, lorsqu'il existait plusieurs approches possibles.

## 4. Contexte

Nom de l'entreprise, organisme ou association ► **La Plateforme CDPI**

Chantier, atelier, service ► **Formation Développeur Web & Web Mobile**

Période d'exercice ► **Du 24/06/2024 au 15/10/2025**

## 5. Informations complémentaires (*facultatif*)

L'extrait anglophone + traduction personnelle est prêt pour l'annexe conformément au RC.

Couverture RC : recherche en anglais (sources, critères, solution) + extrait traduit en annexe.

# DOSSIER PROFESSIONNEL (DP)

## Activité-type 3 : Compétences transversales (veille, qualité/sécurité)

*Exemple n°3* ▶ Démarche qualité : tests, accessibilité & performances

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

J'ai structuré une démarche qualité itérative couvrant :

- **Tests fonctionnels** des parcours clés (navigation, formulaire, bandeau cookies) et du back-office (login admin, CRUD projets, consultation réponses).
- **Accessibilité** : objectif **Lighthouse** > 90 % (contrastes, textes alternatifs, focus visible, ARIA sur éléments interactifs), corrections après audit.

Performances : build Vite, lazy-loading images, code-splitting React, caches Laravel (**config:cache**, **route:cache**, **view:cache**) pour réduire TTFB et éviter les régressions. Les sections Tests et Déploiement de mon dossier détaillent les scénarios, résultats et ajustements.

### 2. Précisez les moyens utilisés :

Plans de test (parcours front & admin), (pré-)Lighthouse, procédure de build & caches en prod, suivi des corrections dans un changelog court.

# DOSSIER PROFESSIONNEL (DP)

## 3. Avec qui avez-vous travaillé ?

En autonomie totale ; sur la lisibilité (accessibilité) et la robustesse des parcours (tests).

## 4. Contexte

Nom de l'entreprise, organisme ou association ► **La Plateforme CDPI**

Chantier, atelier, service ► **Formation Développeur Web & Web Mobile**  
Période d'exercice ► **Du 24/06/2024 au 15/10/2025**

## 5. Informations complémentaires (*facultatif*)

Des tests automatisés plus poussés (Pest/Dusk) sont envisagés ; le dossier décrit d'ailleurs la tentative d'installation Dusk et ses contraintes, avec un plan d'itération.

Couverture RC : tests, accessibilité (Lighthouse), performances + préparation prod (caches, SSL, sauvegardes).

# DOSSIER PROFESSIONNEL (DP)

## Activité-type 4 : Évolutions & outils complémentaires (CMS-like, déploiement, outillage)

*Exemple n° 1* ▶ **Administration type CMS avec Filament (projets & réponses clients)**

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

J'ai industrialisé la gestion de contenu via Filament (admin Laravel) :

- **Modélisation d'admin** : création de Resources “Projects” et “ClientAnswers” (listes, formulaires, actions).
- **Formulaires** : champs typés (titre, slug auto, description riche, tags/techno, upload multiple d’images), règles de validation (unicité slug, taille/format images), messages d’erreur clairs.
- **Tables** : tri, filtres (par techno/statut), pagination, recherche plein-texte, actions de masse (publier/retirer, supprimer).
- **Médias** : configuration du disk (public) et d’un répertoire par projet, renommage et suppression cohérente (éviter les orphelins).
- **Sécurité** : Authentification admin obligatoire, Policies sur les Resources (seul admin autorisé), middleware anti-CSRF, journalisation des actions sensibles (création/suppression).
- **UX admin** : barres d’état (brouillon/publié), boutons rapides (duplicer un projet), toasts de confirmation.

### 2. Précisez les moyens utilisés :

Laravel (Policies, Middleware, Validation), Filament (Forms/Tables/Actions), Eloquent (relations, casts), stockage public, commandes Artisan pour publier les assets Filament.

# DOSSIER PROFESSIONNEL (DP)

## 3. Avec qui avez-vous travaillé ?

Réalisation en autonomie totale, sur l'ergonomie (lisibilité des listes, intitulés, ordre des champs) et les règles d'accès.

## 4. Contexte

Nom de l'entreprise, organisme ou association ► **La Plateforme CDPI**

Chantier, atelier, service ► **Formation Développeur Web & Web Mobile**  
Période d'exercice ► **Du 24/06/2024 au 15/10/2025**

## 5. Informations complémentaires (*facultatif*)

J'ai documenté un cycle de vie “brouillon → publié” et un plan de nettoyage (suppression en cascade des images liées).

Évidences à annexer (reco.) : captures Filament (liste/édition), extrait de ProjectResource (form + table), config de filesystems.php pour le disk public.

Couverture RC : élaborer et mettre en œuvre des composants dans une application de gestion de contenu (admin sécurisée, formulaires/tables, policies).

# DOSSIER PROFESSIONNEL (DP)

## Activité-type 4 : Évolutions & outils complémentaires (CMS-like, déploiement, outillage)

*Exemple n° 2* ▶ **Mise en production & exploitation  
(Plesk, SSL, sauvegardes, monitoring)**

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

J'ai orchestré la mise en production sur Plesk et instauré des garde-fous d'exploitation :

- **Provisioning** : domaine/sous-domaine, racine pointée vers public/, PHP-FPM activé, version PHP cohérente avec Laravel, time zone et locales.
- **Déploiement** : composer install --no-dev --optimize-autoloader, php artisan key:generate (1ère fois), php artisan migrate --force, build front npm ci && npm run build.
- **Optimisations Laravel** : config:cache, route:cache, view:cache, activation OPcache.
- **Sécurité** : Let's Encrypt, force HTTPS (redirection), permissions minimales (storage/, bootstrap/cache), désactivation d'affichage des erreurs en prod, rotation des logs.
- **Variables d'env.** : .env dédié prod (APP\_KEY, DB\_\*, SMTP, APP\_URL), séparation env dev/ prod, pas de secrets commités.
- **Sauvegardes & monitoring** : planification sauvegardes Plesk (fichiers + DB), rétention multi-versions, test de restore ; suivi des ressources (CPU/RAM/IO), alerte e-mail en cas d'échec de sauvegarde.
- **Procédures** : checklist de release (avant/après), et plan de rollback (restauration + migrate:rollback si nécessaire).

### 2. Précisez les moyens utilisés :

Plesk (hébergement, base MySQL, SSL, backups), SSH/SFTP, Composer/NPM, Artisan, logs Laravel et logs serveur (error/access), health-check applicatif (page /health simple).

# DOSSIER PROFESSIONNEL (DP)

## 3. Avec qui avez-vous travaillé ?

Autonomie totale; validation des étapes critiques (HTTPS, sauvegardes, .env, caches).

## 4. Contexte

Nom de l'entreprise, organisme ou association ► **La Plateforme CDPI**

Chantier, atelier, service ► **Formation Développeur Web & Web Mobile**

Période d'exercice ► **Du 24/06/2024 au 15/10/2025**

## 5. Informations complémentaires (*facultatif*)

J'ai prévu la mise en file (queue) pour les e-mails si le volume monte, et la désactivation du mode debug par défaut (APP\_DEBUG=false).

Évidences à annexer (reco.) : extrait anonymisé de .env, capture certificat SSL actif, copie de la checklist de déploiement, screenshot "tâche de sauvegarde Plesk".

Couverture RC : prise en compte des exigences de production (HTTPS, variables d'environnement, sauvegardes, optimisation, monitoring).

# DOSSIER PROFESSIONNEL (DP)

## Activité-type 4 : Évolutions & outils complémentaires (CMS-like, déploiement, outillage)

*Exemple n° 3* ► **Organisation & outillage (design, versionning, qualité)**

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

J'ai cadré l'organisation et les outils pour fiabiliser le projet et préparer l'évolutivité :

- **Design & specs** : Figma (wireframes → écrans hi-fi), mini design system (typos, couleurs, espacements, composants), exports harmonisés.
- **Versionning Git** : branches par feature (feat/projets, fix/upload), pull-requests locales pour revue, tags/releases pour jalons, changelog synthétique par release.
- **Qualité de code** : Prettier/ESLint (front) et conventions côté PHP ; messages de commit conventionnels (type/scope/message) pour générer le changelog plus tard ; hooks locaux (pré-commit) pour formatage.
- **Documentation** : README d'installation/dev, procédure de déploiement, "runbook" incidents (où regarder, comment restaurer).
- **Gestion des tâches** : to-do priorisé (bugs/tech-debt/features), petites US avec critères d'acceptation, timeboxing.

### 2. Précisez les moyens utilisés :

Figma, Git/GitHub, VS Code + extensions (format/lint), scripts NPM/Composer, templates de PR et d'issues, notes de version.

# DOSSIER PROFESSIONNEL (DP)

## 3. Avec qui avez-vous travaillé ?

En autonomie totale ; sur la clarté des PR, pertinence du changelog, lisibilité du README.

## 4. Contexte

Nom de l'entreprise, organisme ou association ► **La Plateforme CDPI**

Chantier, atelier, service ► **Formation Développeur Web & Web Mobile**  
Période d'exercice ► **Du 24/06/2024 au 15/10/2025**

## 5. Informations complémentaires (*facultatif*)

Prochaines étapes : CI légère (lint + build), tests automatisés (unitaires/front), gabarits de pages supplémentaires, amélioration SEO.

Évidences à annexer (reco.) : captures Figma, extrait de changelog, template de PR, README (section “Installation/Build/Deploy”).

Couverture RC : organisation et documentation appuyant la qualité (versionning, design system, procédures), en soutien des compétences front/back.

## **DOSSIER PROFESSIONNEL (DP)**

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

*(facultatif)*

# DOSSIER PROFESSIONNEL (DP)

## Déclaration sur l'honneur

Je soussigné(e) [prénom et nom] ..... **M. Dos Ramos Franco Octavio** .....,  
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 ..... **10 Novembre 2025** .....

pour faire valoir ce que de droit.

Signature :

## **DOSSIER PROFESSIONNEL (DP)**

## **Documents illustrant la pratique professionnelle**

*(facultatif)*

# DOSSIER PROFESSIONNEL (DP)

## ANNEXES

### AT1 — Front-end

#### A1-1 — Loader / Cookies / Maintenance / 404

##### Loader

```
import React from 'react';
import styled from 'styled-components';

export default function Loader () {
    return (
        <StyledWrapper>
            <div className="load">
                <div className="load-inner load-one" />
                <div className="load-inner load-two" />
                <div className="load-inner load-three" />
                <span className="text">Loading...</span>
            </div>
        </StyledWrapper>
    );
}

const StyledWrapper = styled.div`
    height: 100vh;
    width: 100vw;
    display: flex;
    align-items: center;
    justify-content: center;
    background-color: #000;
    .loader {
        position: relative;
        display: flex;
        align-items: center;
        justify-content: center;
        width: 100px;
        height: 100px;
        perspective: 700px;
    }
    .text {
        font-size: 20px;
        font-weight: 700;
        color: #eeeeee;
        z-index: 10;
    }
    .load-inner {
        position: absolute;
        width: 100%;
        height: 100%;
        box-sizing: border-box;
        border-radius: 50%;
    }
    .load-inner.load-one {
        left: 0%;
        top: 0%;
        border-bottom: 3px solid #5c5edc;
        animation: rotate1 1.15s linear infinite;
    }
    .load-inner.load-two {
        right: 0%;
        top: 0%;
        border-right: 3px solid #9147ff;
        animation: rotate2 1.15s 0.1s linear infinite;
    }
    .load-inner.load-three {
        right: 0%;
        bottom: 0%;
        border-top: 3px solid #3b98f6;
        animation: rotate3 1.15s 0.15s linear infinite;
    }
    @keyframes rotate1 {
        0% {
            transform: rotateX(45deg) rotateY(-45deg) rotateZ(0deg);
        }
        100% {
            transform: rotateX(45deg) rotateY(-45deg) rotateZ(360deg);
        }
    }
    @keyframes rotate2 {
        0% {
            transform: rotateX(45deg) rotateY(45deg) rotateZ(0deg);
        }
        100% {
            transform: rotateX(45deg) rotateY(45deg) rotateZ(360deg);
        }
    }
    @keyframes rotate3 {
        0% {
            transform: rotateX(-60deg) rotateY(0deg) rotateZ(0deg);
        }
        100% {
            transform: rotateX(-60deg) rotateY(0deg) rotateZ(360deg);
        }
    }
`;
```

##### Cookies

```
import { CookieManager } from "react-cookie-manager";
import "react-cookie-manager/style.css";

export default function CookieProvider() {
    return (
        <CookieManager
            enableFloatingButton={true}
            theme="light"
            displayType="popup"
            position="bottom-left"
            translations={{
                // Main consent banner/popup/modal
                title: "Souhaitez-vous un cookie ? ⓘ",
                message: "Nous respectons votre vie privée. Choisissez les cookies que vous souhaitez autoriser. Les cookies essentiels sont toujours activés.",
                buttonText: "Accepter tout",
                declineButtonText: "Refuser tout",
                manageButtonText: "Gérer les cookies",
                privacyPolicyText: "Politique de confidentialité",
                // Manage consent modal
                manageTitle: "Préférences de cookies",
                manageMessage: "Gérez vos préférences de cookies ci-dessous. Les cookies essentiels sont toujours activés.",
                // Essential cookies section
                manageEssentialTitle: "Essentiels",
                manageEssentialSubtitle: "Nécessaires au bon fonctionnement du site",
                manageEssentialStatus: "Statut : Toujours activés",
                manageEssentialStatusText: "Toujours activés",
                // Analytics cookies section
                manageAnalyticsTitle: "Analytiques",
                manageAnalyticsSubtitle: "Aident à comprendre comment vous utilisez le site",
                // Social cookies section
                manageSocialTitle: "Réseaux sociaux",
                manageSocialSubtitle: "Activer les fonctionnalités sociales",
                // Advertising cookies section
                manageAdvertTitle: "Publicité",
                manageAdvertSubtitle: "Personnalisent la publicité",
                // Status messages
                manageCookiesStatus: "Statut : {{status}} le {{date}}",
                manageCookiesStatusConsented: "Accepté",
                manageCookiesStatusDeclined: "Refusé",
                // Buttons in manage modal
                manageCancelButtonText: "Annuler",
                manageSaveButtonText: "Enregistrer"
            }}
            onAccept={() => {
                console.log("Tous les cookies acceptés");
                // Initialiser les outils d'analyse
                window.gtag?."consent", "update", { analytics_storage: "granted" });
            }}
            onDecline={() => {
                console.log("Tous les cookies refusés");
                // S'assurer que le suivi est désactivé
                window.gtag?."consent", "update", { analytics_storage: "denied" });
            }}
            onManage={(preferences) => {
                console.log("Préférences personnalisées enregistrées :", preferences);
                // Gérer le consentement granulaire
                if (preferences?.Analytics) {
                    // Activer l'analyse
                }
                if (preferences?.Advertising) {
                    // Activer la personnalisation des publicités
                }
            }}
        />
    );
}
```

# DOSSIER PROFESSIONNEL (DP)

## ANNEXES

### A1-1 (Suite) — Loader / Cookies / Maintenance / 404

#### Maintenance

```
<!DOCTYPE html>
<html lang="{{ str_replace('_', '-', app()>getLocale()) }}>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Kromaxy Dev</title>
    <link href="{{ asset('css/app.css') }}" rel="stylesheet">
    <style>
        .background {
            position: relative;
            width: 100%;
            height: 100vh; /* hauteur pleine fenêtre */
            display: flex;
            justify-content: center;
            align-items: flex-end; /* Positionner en bas */
            color: white;
            text-shadow: 0 0 10px rgba(0,0,0,0.7);
            padding-bottom: 2rem; /* Espace en bas */
            box-sizing: border-box;
            overflow: hidden;
        }
        .video-container {
            position: absolute;
            top: 0;
            left: 0;
            width: 100%;
            height: 100%;
            object-fit: cover;
            z-index: 1;
        }
        .content {
            position: relative;
            z-index: 2;
            padding: 0.1rem;
            background-color: rgba(0, 0, 0, 0.4); /* fond semi-transparent pour lisibilité */
            border-radius: 10px;
            max-width: 90%;
            text-align: center;
        }
    </style>
</head>
<body>
    <div class="background">
        <video class="video-container" autoplay loop muted playsinline>
            {{-- <source src="/Images/Maintenance/FontMaintenance.mp4" type="video/mp4"> --}}
            <source src="/Images/Maintenance/ASTRONAUTE.mp4" type="video/mp4">
        <!-- Message si la vidéo ne fonctionne pas -->
        Votre navigateur ne prend pas en charge la vidéo.
    </video>
    <div class="content">
        <h1 class="text-2xl font-bold mb-4">
            Le site Kromaxy Dev est actuellement en maintenance.<br/>
            Merci de revenir plus tard.
        </h1>
    </div>
</div>
</body>
</html>
```

#### 404

```
import React from 'react';
import V404 from '/public/VideoPage/V404.mp4';
import AppLayout from ".../layouts/AppLayout";

export default function NotFound() {
    return (
        <div className="flex items-center justify-center min-h-screen text-white">
            <div className="text-center">
                <h1 className="text-4xl font-bold mb-4 mt-18">
                    404 - Page Non Trouvée
                </h1>
                <div className="relative overflow-hidden">
                    <video
                        className="w-auto h-auto max-w-full max-h-full"
                        style={{ width: '580px', height: '480px' }} // Modifie ces valeurs pour
                    redimensionner
                        autoPlay
                        loop
                        muted
                        playsInline
                    >
                        <source
                            src={V404}
                            type="video/mp4" />
                    Votre navigateur ne prend pas en charge la lecture vidéo.
                    </video>
                </div>
                <p className="mt-2 text-white">
                    Désolé, la page que vous recherchez n'existe pas ou a été déplacée.
                </p>
                <a
                    href="/Home"
                    className="mt-4 inline-block px-6 py-2 text-sm font-semibold bg-white text-black
                    hover:bg-gray-400 hover:text-white rounded"
                >
                    Retour à l'accueil
                </a>
            </div>
        </div>
    );
}

NotFound.layout = (page) => <AppLayout children={page} />
```

## **DOSSIER PROFESSIONNEL (DP)**

## ANNEXES

## A1-2 — Pages & composants (Front)

## Accueil

A propos

Home

# **DOSSIER PROFESSIONNEL (DP)**

## ANNEXES

## A1-2 (Suite) — Pages & composants (Front)

## All Projets

Projet

## Galerie Projets

# DOSSIER PROFESSIONNEL (DP)

## ANNEXES

### A1-3 — Animations & états d'UI

### Footer Animation réseaux



The screenshot shows a terminal window with a black background and white text. The text is a large block of Java code, which is mostly illegible due to its length and complexity. It appears to be a class definition with various methods, loops, and conditional statements. Some recognizable words include "import", "public", "private", "String", "int", "for", "if", and "else". The code is likely related to the "Footer Animation réseaux" mentioned in the section header.

## **DOSSIER PROFESSIONNEL (DP)**

## ANNEXES

## AT2 — Back-end

## QuestionnaireSeeder.php

## A2-1 — Migrations & seeders

## CreateProject.php

```
<?php

namespace App\Filament\Resources\ProjectResource\Pages;

use App\Filament\Resources\ProjectResource;
use Filament\Actions;
use Filament\Resources\Pages\CreateRecord;

class CreateProject extends CreateRecord
{
    protected static string $resource = ProjectResource::class;
}
```

# DOSSIER PROFESSIONNEL (DP)

## ANNEXES

### A2-2 — Logique métier & sécurité

#### ProjetController.php

```
<?php

namespace App\Http\Controllers;

use App\Models\Project;
use Illuminate\Http\Request;
use Inertia\Inertia;

class ProjectController extends Controller
{
    private function transformProjects($projects)
    {
        return $projects->map(function ($project) {
            return [
                'id' => $project->id,
                'title' => $project->title,
                'category' => $project->category,
                'description' => $project->description,
                'image' => $project->image1 ? asset('storage/' . $project->image1) :
                '/placeholder.svg',
                'project_url' => $project->project_url,
                'requestLink' => '/questionnaire/client',
            ];
        });
    }

    public function index()
    {
        $projects = Project::latest()->take(6)->get();
        return Inertia::render('Home', [
            'projects' => $this->transformProjects($projects),
        ]);
    }

    public function allProjects()
    {
        // Pour afficher l'ensemble des projets
        $projects = Project::orderBy('created_at', 'desc')->paginate(12);

        return Inertia::render('AllProjects', [
            'projects' => $this->transformProjects($projects),
        ]);
    }

    public function show($id)
    {
        $current = Project::findOrFail($id);
        $projects = Project::orderBy('created_at', 'desc')->get();

        $count = $projects->count();
        $currentIndex = $projects->search(fn($item) => $item->id === $current->id);
        $nextIndex = ($currentIndex - 1 + $count) % $count;
        $prevIndex = ($currentIndex + 1) % $count;

        $prevProject = $projects[$prevIndex];
        $nextProject = $projects[$nextIndex];

        return Inertia::render('Project', [
            'project' => [
                'id' => $current->id,
                'title' => $current->title,
                'category' => $current->category,
                'description' => $current->description,
                'date' => $current->creation_date->format('F d, Y'),
                'client' => $current->client,
                'image' => $current->image1 ? asset('storage/' . $current->image1) :
                '/placeholder.svg',
                'secondary' => [
                    'image' => $current->image2 ? asset('storage/' . $current->image2) : null,
                    'title' => $current->second_title,
                    'description' => $current->second_description,
                    'link' => $current->project_url,
                ],
                'cta' => [
                    'image' => $current->image3 ? asset('storage/' . $current->image3) : null,
                    'questionnaireLink' => '/questionnaire/client',
                ],
            ],
            'prevProject' => $prevProject ? ['id' => $prevProject->id, 'title' => $prevProject->title]
            : null,
            'nextProject' => $nextProject ? ['id' => $nextProject->id, 'title' => $nextProject->title]
            : null,
            'totalProjects' => $projects->count(),
            'projectIndex' => $projects->search(fn($p) => $p->id === $current->id) + 1,
        ]);
    }
}
```

#### ConfirmationPasswordController.php

```
<?php

namespace App\Http\Controllers\Auth;

use App\Http\Controllers\Controller;
use Illuminate\Http\RedirectResponse;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use Illuminate\Validation\ValidationException;
use Inertia\Inertia;
use Inertia\Response;

class ConfirmablePasswordController extends Controller
{
    /**
     * Show the confirm password page.
     */
    public function show(): Response
    {
        return Inertia::render('auth/confirm-password');
    }

    /**
     * Confirm the user's password.
     */
    public function store(Request $request): RedirectResponse
    {
        if (! Auth::guard('web')->validate([
            'email' => $request->user()->email,
            'password' => $request->password,
        ])) {
            throw ValidationException::withMessages([
                'password' => __('auth.password'),
            ]);
        }

        $request->session()->put('auth.password_confirmed_at', time());

        return redirect()->intended(route('dashboard', absolute: false));
    }
}
```

#### Tinker password

```
X
\App\Models\User::create([
    'name' => 'Admin',
    'email' => 'kromaxydev@gmail.com',
    'password' => bcrypt('815jHFrGFGJHDszGÜlj21'),
]);

App\Models\User {#6216
    name: "Admin",
    email: "kromaxydev@gmail.com",
    #password: "$2y$12$x7oR6vN8kXPs/uyFmREQN04V4LpzMtilEuAsWaFwdSSIS86M2gt2i",
    updated_at: "2025-06-24 13:10:27",
    created_at: "2025-06-24 13:10:27",
    id: 1,
```

# DOSSIER PROFESSIONNEL (DP)

## ANNEXES

### A2-2 — Logique métier & sécurité (Suite)

#### Tinker MySQL

Insère le compte admin via SQL  
Adapte la requête suivante avec ton hash et tes infos :

```
sql

INSERT INTO admin_users (name, email, password, created_at, updated_at)
VALUES (
    'Super Admin',
    'admin@example.com',
    '$2y$12$M3Z0R3z7lHPUVmJ9rpfK..V3.zyJXI8VHSCjdmuxDctp5g9mQia60',
    NOW(),
    NOW()
);
ou avec ID

INSERT INTO admin_users (id, name, email, password, created_at, updated_at)
VALUES (
    1,
    'Super Admin',
    'kromaxydev@gmail.com',
    '$2y$12$M3Z0R3z7lHPUVmJ9rpfK..V3.zyJXI8VHSCjdmuxDctp5g9mQia60',
    NOW(),
    NOW()
);
```

# DOSSIER PROFESSIONNEL (DP)

## ANNEXES

### A2-3 — Admin Filament : login & Resource

#### login.blade.php

```
<x-filament-panels::page.simple>
@if (filament()>hasRegistration())
    <x-slot name="subheading">
        {{ __('filament-panels::pages/auth/login.actions.register.before') }}
        {{ $this->registerAction }}
    </x-slot>
@endif

{{

\Filament\Support\Facades\FilamentView::renderHook(\Filament\View\PanelsRenderHook::AUTH_LOGIN_FORM_BEFORE, scopes: $this->getRenderHookScopes()) }}

<x-filament-panels::form id="form" wire:submit="authenticate">
    {{ $this->form }}

    <x-filament-panels::form.actions
        :actions="$this->getCachedFormActions()"
        :full-width="$this->hasFullWidthFormActions()" />
</x-filament-panels::form>

{{

\Filament\Support\Facades\FilamentView::renderHook(\Filament\View\PanelsRenderHook::AUTH_LOGIN_FORM_AFTER, scopes: $this->getRenderHookScopes()) }}
</x-filament-panels::page.simple>
```

#### ProjectResource

```
x
<?php

namespace App\Filament\Resources;

use App\Filament\Resources\ProjectResource\Pages;
use App\Filament\Resources\ProjectResource\RelationManagers;
use App\Models\Project;
use Carbon\Carbon;
use Filament\Forms\Form;
use Filament\Resources\Resource;
use Filament\Tables\Table;
use Illuminate\Database\Eloquent\Builder;
use Illuminate\Database\Eloquent\SoftDeletingScope;

use Filament\Forms\Components\TextInput;
use Filament\Forms\Components\Textarea;
use Filament\Forms\Components\DatePicker;
use Filament\Forms\Components\Select;
use Filament\Forms\Components\FileUpload;

use Filament\Tables\Columns\ImageColumn;

class ProjectResource extends Resource
{
    protected static ?string $model = Project::class;
    protected static ?string $navigationIcon = 'heroicon-o-rectangle-stack';

    public static function form(Form $form): Form
    {
        return $form
            ->schema([
                TextInput::make('title')
                    ->required()
                    ->label('Titre'),
                Textarea::make('description')
                    ->required()
                    ->label('Description'),
                DatePicker::make('creation_date')
                    ->required()
                    ->label('Date de création'),
                TextInput::make('client')
                    ->label('Client'),
                Select::make('category')
                    ->options([
                        'web' => 'Web',
                        'projet perso' => 'Projet Perso',
                        'divers' => 'Divers',
                    ])
                    ->required()
                    ->label('Catégorie'),
                TextInput::make('second_title')
                    ->label('Deuxième Titre'),
                Textarea::make('second_description')
                    ->label('Deuxième Description'),
                TextInput::make('project_url')
                    ->label('URL du projet'),
                FileUpload::make('image1')
                    ->label('Image 1')
                    ->image()
                    ->directory('projects/images')
                    ->visibility('public')
                    ->preservefilenames(),
                FileUpload::make('image2')
                    ->label('Image 2')
                    ->image()
                    ->directory('projects/images')
                    ->visibility('public')
                    ->preservefilenames(),
                FileUpload::make('image3')
                    ->label('Image 3')
                    ->image()
                    ->directory('projects/images')
                    ->visibility('public')
                    ->preservefilenames(),
            ]);
    }

    public static function table(Table $table): Table
    {
        return $table
            ->columns([
                ImageColumn::make('image1')
                    ->label('Aperçu')
                    ->size('sm')
                    ->optional(),
                CircularImageColumn::make('image2')
                    ->optional()
                    ->label('Image ronde'),
                Tables\Columns\TextColumn::make('title')->label('Titre'),
                Tables\Columns\TextColumn::make('category')->label('Catégorie'),
                Tables\Columns\TextColumn::make('creation_date')->label('Date')->date(),
            ])
            ->filters([
                // ...
            ])
            ->actions([
                Tables\Actions>EditAction::make(),
            ])
            ->bulkActions([
                Tables\Actions\BulkActionGroup::make([
                    Tables\Actions>DeleteAction::make(),
                ]),
            ]);
    }

    public static function getRelations(): array
    {
        return [
            // ...
        ];
    }

    public static function getPages(): array
    {
        return [
            'index' => Pages\ListProjects::route('/'),
            'create' => Pages\CreateProject::route('/create'),
            'edit' => Pages\EditProject::route('/{record}/edit'),
        ];
    }
}
```

# DOSSIER PROFESSIONNEL (DP)

## ANNEXES

### A2-4 — Réponses clients & mails

#### ClientQuestionnaireResponseResource

```
<?php

namespace App\Filament\Resources;
use App\Filament\Resources\ClientQuestionnaireResponseResource;
use App\Models\ClientQuestionnaireResponse;
use Filament\Resources\Resource;
use Filament\Tables;
use Filament\Tables\Columns\TextColumn;

class ClientQuestionnaireResponseResource extends Resource
{
    protected static ?string $model = ClientQuestionnaireResponse::class;
    protected static ?string $navigationIcon = 'heroicon-o-clipboard-document-list';
    protected static ?string $navigationLabel = 'Réponses clients';

    // protected $casts = ['responses' => 'array'];

    public static function table(Tables\Table $table): Tables\Table
    {
        return $table
            ->columns([
                TextColumn::make('client_name')->label('Nom du client'),
                TextColumn::make('client_email')->label('Email'),
                TextColumn::make('created_at')->label('Date')->dateTime('d/m/Y H:i');

                // Champs extraits du JSON "responses"
                TextColumn::make('responses')
                    ->label('Nom')
                    ->formatStateUsing(fn ($$state) => $$state['nom'] ?? ''),
                TextColumn::make('responses')
                    ->label('Prénom')
                    ->formatStateUsing(fn ($$state) => $$state['prenom'] ?? ''),
                TextColumn::make('responses')
                    ->label('Téléphone')
                    ->formatStateUsing(fn ($$state) => $$state['telephone'] ?? ''),
                TextColumn::make('responses')
                    ->label('Entreprise')
                    ->formatStateUsing(fn ($$state) => $$state['entreprise'] ?? ''),
                TextColumn::make('responses')
                    ->label('Type de projet (création)')
                    ->formatStateUsing(fn ($$state) => $$state['projectType'] ? 'Oui' : 'Non'),
                TextColumn::make('responses')
                    ->label('Type de projet (refonte)')
                    ->formatStateUsing(fn ($$state) => $$state['projectRefonte'] ? 'Oui' : 'Non'),
                TextColumn::make('responses')
                    ->label('Type de projet (autre)')
                    ->formatStateUsing(fn ($$state) => $$state['projectOther'] ? 'Oui' : 'Non'),
                TextColumn::make('responses')
                    ->label('Présentation projet')
                    ->formatStateUsing(fn ($$state) => $$state['projectDescription'] ?? ''),
                TextColumn::make('responses')
                    ->label('Autre projet')
                    ->formatStateUsing(fn ($$state) => $$state['projectImportance'] ?? ''),
                TextColumn::make('responses')
                    ->label('Cible')
                    ->formatStateUsing(fn ($$state) => $$state['targetAudience'] ?? ''),
                TextColumn::make('responses')
                    ->label('Utilisateurs finaux')
                    ->formatStateUsing(fn ($$state) => $$state['endUsers'] ?? ''),
                TextColumn::make('responses')
                    ->label('Concurrents')
                    ->formatStateUsing(fn ($$state) => $$state['competitors'] ?? ''),
                TextColumn::make('responses')
                    ->label('Échéance')
                    ->formatStateUsing(fn ($$state) => $$state['timeline'] ?? ''),
                TextColumn::make('responses')
                    ->label('Budget')
                    ->formatStateUsing(fn ($$state) => $$state['budget'] ?? ''),
                TextColumn::make('responses')
                    ->label('Exigences design')
                    ->formatStateUsing(fn ($$state) => $$state['designRequirements'] ?? ''),
                TextColumn::make('responses')
                    ->label('Technologie préférée')
                    ->formatStateUsing(fn ($$state) => $$state['technologyPreference'] ?? ''),
                TextColumn::make('responses')
                    ->label('Sites inspirants')
                    ->formatStateUsing(fn ($$state) => $$state['inspirationalSites'] ?? ''),
                TextColumn::make('responses')
                    ->label('Besoin admin')
                    ->formatStateUsing(fn ($$state) => isset($$state['needAdmin']) ? ($$state['needAdmin'] ? 'Oui' : 'Non'))
                TextColumn::make('responses')
                    ->label('Besoin support')
                    ->formatStateUsing(fn ($$state) => isset($$state['needSupport']) ? ($$state['needSupport'] ? 'Oui' : 'Non'))
                TextColumn::make('responses')
                    ->label('Gestion contenu')
                    ->formatStateUsing(fn ($$state) => $$state['otherContentManagement'] ?? ''),
                TextColumn::make('responses')
                    ->label('Objectifs')
                    ->formatStateUsing(fn ($$state) => $$state['helpObjectives'] ?? ''),
                TextColumn::make('responses')
                    ->label('Pourquoi travailler avec moi')
                    ->formatStateUsing(fn ($$state) => $$state['workWithMeReason'] ?? ''),
                TextColumn::make('responses')
                    ->label('CGU acceptées')
                    ->formatStateUsing(fn ($$state) => isset($$state['acceptTerms']) ? ($$state['acceptTerms'] ? 'Oui' : 'Non'))
            ]);
    }

    public static function getPAGES(): array
    {
        return [
            'index' => Pages\ListClientQuestionnaireResponses::route('/'),
            'create' => Pages\CreateClientQuestionnaireResponse::route('/create'),
            'edit' => Pages>EditClientQuestionnaireResponse::route('/{record}/edit'),
            'view' => Pages\ViewClientQuestionnaireResponse::route('/{record}'),
        ];
    }
}
```

#### QuestionnaireController

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Inertia\Inertia;
use Illuminate\Support\Facades\Mail;
use Illuminate\Support\Facades\Log;
use App\Mail\QuestionnaireMail;
// use App\Models\Question;

use App\Models\ClientQuestionnaireResponse;

class QuestionnaireController extends Controller
{
    public function index()
    {
        return Inertia::render('FormulaireQuestionnaire');
    }

    public function store(Request $request)
    {
        $validated = $request->validate([
            'nom' => 'required|string|max:255',
            'prenom' => 'required|string|max:255',
            'telephone' => 'required|string|max:20',
            'email' => 'required|email|max:255',
            'entreprise' => 'required|string|max:255',
            'projectType' => 'nullable|boolean',
            'projectRefonte' => 'nullable|boolean',
            'projectOther' => 'nullable|boolean',
            'projectDescription' => 'required|string',
            'projectImportance' => 'required|string',
            'targetAudience' => 'required|string',
            'endUsers' => 'required|string',
            'competitors' => 'required|string',
            'timeline' => 'required|string',
            'budget' => 'required|string',
            'designRequirements' => 'nullable|string',
            'technologyPreference' => 'nullable|string',
            'inspirationalSites' => 'nullable|string',
            'needAdmin' => 'nullable|boolean',
            'needSupport' => 'nullable|boolean',
            'otherContentManagement' => 'nullable|string',
            'helpObjectives' => 'nullable|string',
            'workWithMeReason' => 'nullable|string',
            'acceptTerms' => 'accepted',
            // 'acceptTerms' => 'required|boolean',
            // 'responses.acceptTerms' => 'accepted',
        ]);

        try {
            // Enregistrement dans la table client_questionnaire_responses
            ClientQuestionnaireResponse::create([
                'client_name' => $validated['nom'] . ' ' . $validated['prenom'],
                'client_email' => $validated['email'],
                // Si tu as une notion de questionnaire_id, mets-la ici, sinon mets 1 ou null
                'questionnaire_id' => 1,
                'responses' => json_encode($validated),
            ]);

            // Enregistrement en base
            // Question::create($validated);

            // Envoyer l'email avec les données du questionnaire
            Mail::to(config('mail.admin_email', 'kromaxydev@gmail.com'))
                ->send(new QuestionnaireMail($validated));

            return response()->json([
                'success' => true,
                'message' => 'Questionnaire envoyé et enregistré avec succès !'
            ]);
        } catch (\Exception $e) {
            Log::error('Erreur envoi questionnaire: ' . $e->getMessage());
            return response()->json([
                'success' => false,
                'message' => 'Une erreur est survenue lors de l\'envoi.',
                500);
        }
    }
}
```

# DOSSIER PROFESSIONNEL (DP)

## ANNEXES

### AT4 — Évolutions & outils (CMS-like, déploiement, outillage)

#### A4-1 — Admin CMS-like (Filament)

##### ProjectResource

```
<?php

namespace App\Filament\Resources;

use App\Filament\Resources\ProjectResourcePages;
use App\Filament\Resources\ProjectResource\RelationManagers;
use App\Models\Project;
use Filament\Filters;
use Filament\Filters\TextFilter;
use Filament\Resources\Resource;
use FilamentTables;
use FilamentTables\Table;
use Illuminate\Database\Eloquent\Builder;
use Illuminate\Database\Eloquent\SoftDeletingScope;

use FilamentForms\Components\TextInput;
use FilamentForms\Components\Textarea;
use FilamentForms\Components\TextEditor;
use FilamentForms\Components>Select;
use FilamentForms\Components\FileUpload;
use FilamentTables\Columns\ImageColumn;

class ProjectResource extends Resource
{
    protected static ?string $model = Project::class;

    protected static ?string $navigationIcon = 'heroicon-o-rectangle-stack';

    public static function form(Form $form): Form
    {
        return $form
            ->schema([
                TextInput::make('title')
                    ->required()
                    ->label('Titre'),
                Textarea::make('description')
                    ->required()
                    ->label('Description'),
                Datepicker::make('creation_date')
                    ->required()
                    ->label('Date de création'),
                TextInput::make('client')
                    ->label('Client'),
                Select::make('category')
                    ->options([
                        'web' => 'Web',
                        'projet perso' => 'Projet Perso',
                        'divers' => 'Divers',
                    ])
                    ->required()
                    ->label('Catégorie'),
                TextInput::make('second_title')
                    ->label('Deuxième titre'),
                Textarea::make('second_description')
                    ->label('Deuxième description'),
                TextInput::make('project_url')
                    ->label('URL du projet'),
                FileUpload::make('image1')
                    ->label('Image 1')
                    ->image()
                    ->directory('projects/images')
                    ->visibility('public')
                    ->preservefilenames(),
                FileUpload::make('image2')
                    ->label('Image 2')
                    ->image()
                    ->directory('projects/images')
                    ->visibility('public')
                    ->preservefilenames(),
                FileUpload::make('image3')
                    ->label('Image 3')
                    ->image()
                    ->directory('projects/images')
                    ->visibility('public')
                    ->preservefilenames(),
            ])
            ->publicPages([
                Tables\Column::make('title')->label('Titre'),
                Tables\Column::make('category')->label('Catégorie'),
                Tables\Column::make('creation_date')->label('Date')->date(),
            ])
            ->filters([
                Tables\Actions\EditAction::make(),
            ])
            ->actions([
                Tables\Actions\BulkActionGroup::make([
                    Tables\Actions\DeleteBulkAction::make(),
                ]),
            ]);
    }

    public static function table(Table $table): Table
    {
        return $table
            ->columns([
                Tables\Column::make('image1')
                    ->label('Aperçu')
                    ->disk('public')
                    ->size(100),
                Tables\Column::make('title')->label('Titre'),
                Tables\Column::make('category')->label('Catégorie'),
                Tables\Column::make('creation_date')->label('Date')->date(),
            ])
            ->filters([
                Tables\Actions\EditAction::make(),
            ])
            ->actions([
                Tables\Actions\BulkActionGroup::make([
                    Tables\Actions\DeleteBulkAction::make(),
                ]),
            ]);
    }

    public static function getRelations(): array
    {
        return [
            // ...
        ];
    }

    public static function getPages(): array
    {
        return [
            'index' => Pages\ListProjects::route('/'),
            'create' => Pages\CreateProject::route('/create'),
            'edit' => Pages>EditProject::route('/{record}/edit'),
        ];
    }
}
```

##### Auth

```
<?php

return [
    /*
     * Authentication Defaults
     */
    /*
     * This option defines the default authentication "guard" and password
     * reset "broker" for your application. You may change these values
     * as required, but they're a perfect start for most applications.
     */
    'defaults' => [
        'guard' => env('AUTH_GUARD', 'web'),
        'passwords' => env('AUTH_PASSWORD_BROKER', 'users'),
    ],
    /*
     * Authentication Guards
     */
    /*
     * Next, you may define every authentication guard for your application.
     * Of course, a great default configuration has been defined for you
     * which utilizes session storage plus the Eloquent user provider.
     *
     * All authentication guards have a user provider, which defines how the
     * users are actually retrieved out of your database or other storage
     * system used by the application. Typically, Eloquent is utilized.
     */
    'guards' => [
        'web' => [
            'driver' => 'session',
            'provider' => 'users',
        ],
    ],
    /*
     * User Providers
     */
    /*
     * All authentication guards have a user provider, which defines how the
     * users are actually retrieved out of your database or other storage
     * system used by the application. Typically, Eloquent is utilized.
     *
     * If you have multiple user tables or models you may configure multiple
     * providers to represent the model / table. These providers may then
     * be assigned to any extra authentication guards you have defined.
     */
    'providers' => [
        'users' => [
            'driver' => 'eloquent',
            'model' => env('AUTH_MODEL', App\Models\User::class),
        ],
        // 'users' => [
        //     'driver' => 'database',
        //     'table' => 'users',
        // ],
    ],
    /*
     * Resetting Passwords
     */
    /*
     * These configuration options specify the behavior of Laravel's password
     * reset functionality, including the table utilized for token storage
     * and the user provider that is invoked to actually retrieve users.
     */
    /*
     * The expiry time is the number of minutes that each reset token will be
     * considered valid. This security feature keeps tokens short-lived so
     * they have less time to be guessed. You may change this as needed.
     */
    /*
     * The throttle setting is the number of seconds a user must wait before
     * generating more password reset tokens. This prevents the user from
     * quickly generating a very large amount of password reset tokens.
     */
    'passwords' => [
        'users' => [
            'provider' => 'users',
            'table' => env('AUTH_PASSWORD_RESET_TOKEN_TABLE', 'password_reset_tokens'),
            'expire' => 60,
            'throttle' => 60,
        ],
    ],
    /*
     * Password Confirmation Timeout
     */
    /*
     * Here you may define the amount of seconds before a password confirmation
     * window expires and users are asked to re-enter their password via the
     * confirmation screen. By default, the timeout lasts for three hours.
     */
    'password_timeout' => env('AUTH_PASSWORD_TIMEOUT', 10800),
];

```

##### LoginRequest

```
<?php

namespace App\Http\Requests\Auth;

use Illuminate\Foundation\Http\FormRequest;
use Illuminate\Support\Facades\RateLimiter;
use Illuminate\Support\Facades\Validator;
use Illuminate\Validation\Rule;
use Illuminate\Validation\ValidationException;

class LoginRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize(): bool
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array|string
     */
    public function rules(): array
    {
        return [
            'email' => ['required', 'string', 'email'],
            'password' => ['required', 'string'],
        ];
    }

    /**
     * Attempt to authenticate the request's credentials.
     *
     * @throws \Illuminate\Validation\ValidationException
     */
    public function authenticated(): void
    {
        $this->ensureIsNotRateLimited();

        if ($this->attempt(['email', 'password'], $this->boolean('remember'))) {
            RateLimiter::hit($this->throttleKey());

            throw ValidationException::withMessages([
                'email' => ['auth.failed'],
            ]);
        }

        RateLimiter::clear($this->throttleKey());
    }

    /**
     * Ensure the login request is not rate limited.
     *
     * @throws \Illuminate\Validation\ValidationException
     */
    public function ensureIsNotRateLimited(): void
    {
        if (!RateLimiter::tooManyAttempts($this->throttleKey(), 5)) {
            return;
        }

        event(new Lockout($this));

        $seconds = RateLimiter::availableIn($this->throttleKey());

        throw ValidationException::withMessages([
            'email' => ['auth.throttle'],
            'seconds' => [
                'seconds' => $seconds,
                'minutes' => ceil($seconds / 60),
            ],
        ]);
    }

    /**
     * Get the rate limiting throttle key for the request.
     */
    public function throttleKey(): string
    {
        return Str::transliterate(Str::lower($this->string('email'))).'.{$this->ip()});
    }
}
```

# DOSSIER PROFESSIONNEL (DP)

## ANNEXES

### A4-2 — Middlewares & session (Inertia/Authent)

#### AuthenticatedSessionController.php

```
<?php

namespace App\Http\Controllers\Auth;

use App\Http\Controllers\Controller;
use App\Http\Requests\Auth\LoginRequest;
use Illuminate\Http\RedirectResponse;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Route;
use Inertia\Inertia;
use Inertia\Response;

class AuthenticatedSessionController extends Controller
{
    /**
     * Show the login page.
     */
    public function create(Request $request): Response
    {
        return Inertia::render('auth/login', [
            'canResetPassword' => Route::has('password.request'),
            'status' => $request->session()->get('status'),
        ]);
    }

    /**
     * Handle an incoming authentication request.
     */
    public function store(LoginRequest $request): RedirectResponse
    {
        $request->authenticate();

        $request->session()->regenerate();

        return redirect()->intended(route('dashboard', ['absolute' => false]));
    }

    /**
     * Destroy an authenticated session.
     */
    public function destroy(Request $request): RedirectResponse
    {
        Auth::guard('web')->logout();

        $request->session()->invalidate();
        $request->session()->regenerateToken();

        return redirect('/');
    }
}
```

#### HandleInertiaRequests.php

```
<?php

namespace App\Http\Middleware;

use Illuminate\Foundation\Inspiring;
use Illuminate\Http\Request;
use Inertia\Middleware;
use TightenZiggy\Ziggy;

class HandleInertiaRequests extends Middleware
{
    /**
     * The root template that's loaded on the first page visit.
     *
     * @see https://inertiajs.com/server-side-setup#root-template
     */
    protected $rootView = 'app';

    /**
     * Determines the current asset version.
     *
     * @see https://inertiajs.com/asset-versioning
     */
    public function version(Request $request): ?string
    {
        return parent::version($request);
    }

    /**
     * Define the props that are shared by default.
     *
     * @see https://inertiajs.com/shared-data
     */
    public function share(Request $request): array
    {
        [$message, $author] = str(Inspiring::quotes()->random())->explode('-');

        return [
            ...parent::share($request),
            'name' => config('app.name'),
            'quote' => ['message' => trim($message), 'author' => trim($author)],
            'auth' => [
                'user' => $request->user(),
                'ziggy' => fn (): array => [
                    ...new Ziggy)->toArray(),
                    'location' => $request->url(),
                ],
            ],
        ];
    }
}
```

# DOSSIER PROFESSIONNEL (DP)

## ANNEXES

### A4-3 — Pages Back-office (CRUD)

#### Create

```
<?php

namespace App\Filament\Resources\ProjectResource\Pages;

use App\Filament\Resources\ProjectResource;
use Filament\Actions;
use Filament\Resources\Pages\CreateRecord;

class CreateProject extends CreateRecord
{
    protected static string $resource = ProjectResource::class;
}
```

#### ClientQuestionnaireResponse.php

##### Envoi réponse

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class ClientQuestionnaireResponse extends Model
{

    protected $table = 'client_questionnaire_responses';

    protected $fillable = [
        'client_name',
        'client_email',
        'questionnaire_id',
        'responses',
    ];

    protected $casts = [
        'responses' => 'array',
    ];

    // Acces pour forcer le décodage JSON
    public function getResponsesAttribute($value)
    {
        if (is_string($value)) {
            return json_decode($value, true);
        }
        return $value;
    }

    // Mutator pour encoder en JSON lors de la sauvegarde
    public function setResponsesAttribute($value)
    {
        $this->attributes['responses'] = is_array($value) ? json_encode($value) :
$value;
    }

    public function questionnaire()
    {
        // return $this->belongsTo(\App\Models\Question::class, 'questionnaire_id');
        return $this->belongsTo(\App\Models\Questionnaire::class, 'questionnaire_id');
    }
}
```

#### Edit

```
<?php

namespace App\Filament\Resources\ProjectResource\Pages;

use App\Filament\Resources\ProjectResource;
use Filament\Actions;
use Filament\Resources\Pages>EditRecord;

class EditProject extends EditRecord
{
    protected static string $resource = ProjectResource::class;

    protected function getHeaderActions(): array
    {
        return [
            Actions\DeleteAction::make(),
        ];
    }
}
```

#### List

```
<?php

namespace App\Filament\Resources\ProjectResource\Pages;

use App\Filament\Resources\ProjectResource;
use Filament\Actions;
use Filament\Resources\Pages\ListRecords;

class ListProjects extends ListRecords
{
    protected static string $resource = ProjectResource::class;

    protected function getHeaderActions(): array
    {
        return [
            Actions\CreateAction::make(),
        ];
    }
}
```