

# **Cahier des charges : Site Web de Portfolio avec Opérations CRUD pour l'Administration**

## **Vue d'ensemble :**

L'objectif de ce projet est de créer un site web de portfolio où les visiteurs peuvent consulter des projets et des compétences sans avoir besoin de s'authentifier. De plus, un panneau d'administration sera mis en place pour effectuer des opérations CRUD permettant de gérer les compétences et les projets. Le frontend sera développé en utilisant React avec Tailwind CSS, tandis que le backend sera alimenté par Laravel Breeze avec Inertia.js. Nous allons également mettre en place des microservices pour gérer l'authentification admin ainsi que les opérations CRUD sur les tables des projets et des compétences.

## **Fonctionnalités :**

### **1. Pages Publiques du Portfolio**

- Créer des pages publiques pour présenter les projets, les compétences et d'autres contenus pertinents.
- Inclure des liens de navigation pour faciliter le passage entre différentes sections (par exemple, Accueil, Projets, Compétences, À propos).

### **2. Présentation des Projets**

- Afficher une grille ou une liste des projets du portfolio avec les détails du projet (nom, description, technologies utilisées, captures d'écran, liens).
- Fournir une page dédiée à chaque projet avec des informations détaillées et des liens vers le projet.

### **3. \*\*Affichage des Compétences\*\***

- Présenter les compétences avec des descriptions, des niveaux de maîtrise et toute autre information pertinente.
- Possibilité de catégoriser les compétences ou de les regrouper en fonction des domaines d'expertise.

### **4. Connexion Admin :**

- Implémenter une page de connexion pour permettre l'accès à l'administration du panneau.
- Utiliser Laravel Breeze pour l'authentification de l'admin seulement.

### **5. Tableau de bord de l'Administration**

- Afficher un tableau de bord avec des liens de navigation pour gérer les compétences et les projets (par exemple, Liste des Compétences, Liste des Projets, Ajouter une Compétence, Modifier une Compétence, Ajouter un Projet, Modifier un Projet).

### **6. Gestion des Compétences**

- Permettre à l'administrateur d'effectuer des opérations CRUD pour gérer les compétences.
- Implémenter des fonctionnalités pour ajouter de nouvelles compétences, modifier les compétences existantes et supprimer des compétences.

### **7. Gestion des Projets**

- Permettre à l'administrateur de gérer les projets du portfolio avec des opérations CRUD.

- Créer des fonctionnalités pour ajouter de nouveaux projets, modifier les détails des projets et supprimer des projets.

## 8. Design Responsive :

- S'assurer que le site web du portfolio et le panneau d'administration sont responsifs et accessibles sur différents appareils (ordinateur de bureau, tablette, mobile).
- Utiliser Tailwind CSS pour le styling afin de maintenir un design cohérent sur différentes tailles d'écran.

## 9. Optimisation SEO :

- Appliquer des pratiques SEO de base telles que les balises meta, les données structurées et les mots-clés pertinents pour améliorer la visibilité dans les moteurs de recherche pour les pages publiques du portfolio.

## 10. Mesures de Sécurité

- Sécuriser les routes du panneau d'administration en utilisant l'authentification Laravel Breeze.
- Implémenter une protection CSRF et une validation des entrées pour les formulaires afin de prévenir les vulnérabilités de sécurité.

## 11. API Backend

- Configurer des endpoints API en utilisant Laravel Breeze et Inertia.js pour la communication entre le frontend et le backend.
- Gérer les opérations CRUD (Créer, Lire, Mettre à jour, Supprimer) pour les compétences et les projets en utilisant des routes API.

## Technologies Utilisées :

- **Frontend** : React (avec Inertia.js pour le rendu côté serveur), Tailwind CSS pour le styling.
- **Backend** : Laravel Breeze pour l'authentification admin, Inertia.js pour l'intégration API.
- **Base de Données** : MySQL.

## Microservices :

- Mettre en place des microservices pour l'authentification admin, les opérations CRUD sur les tables des projets et des compétences.

## Déploiement :

- Héberger le site web du portfolio et le panneau d'administration frontend sur une plateforme adaptée (par exemple, Vercel, Netlify) pour le déploiement de fichiers statiques.

Cette approche devrait fournir une architecture solide pour votre projet de site web de portfolio avec des fonctionnalités avancées d'administration et une expérience utilisateur responsive et sécurisée.