

Cahier des charges – Application Web Blog Laravel

Table des matières

1. Présentation du projet.....	1
2. Objectifs	2
3. Fonctionnalités principales	2
3.1. Interface utilisateur (Front-End)	2
3.2. Gestion du contenu (Back-End)	2
3.3. Gestion des utilisateurs	3
3.4. Autres fonctionnalités.....	3
4. Contenu.....	3
5. Contraintes techniques.....	4
6. Maintenance et évolutivité.....	4
7. Validation du projet.....	4

1. Présentation du projet

- **Nom du projet :** Blog Laravel 10 (ou un nom plus spécifique si tu en as un)
- **Objectif principal :** Développer une application web de type blog moderne et fonctionnelle, utilisant le framework Laravel 10. Le site doit permettre la consultation d'articles, l'interaction via les commentaires, et offrir une interface d'administration pour la gestion du contenu. L'objectif est de mettre en pratique les concepts du développement web full-stack avec Laravel.

2. Objectifs

- **Plateforme responsive et moderne** : Développement du site sous Laravel 10 avec Tailwind CSS et Alpine.js pour un rendu esthétique, fluide et adapté à tous les écrans (PC, tablette, mobile).
- **Navigation et recherche** : Mise en place d'un système de filtrage par catégories et étiquettes, ainsi qu'une barre de recherche textuelle pour faciliter l'accès au contenu.
- **Gestion de contenu simplifiée** : Fournir une interface d'administration claire pour la gestion des articles par les utilisateurs autorisés.

3. Fonctionnalités principales

3.1. Interface utilisateur (Front-End)

- **Design et expérience utilisateur** :
 - Interface épurée et orientée contenu, utilisant Tailwind CSS pour le style.
 - Navigation fluide et intuitive entre les différentes sections.
 - Pages principales : accueil (liste des articles), page article détaillée, pages de filtrage par catégorie/tag, pages d'authentification (connexion, inscription), page compte utilisateur.
- **Fonctionnalités intégrées** :
 - Affichage des articles avec titre, contenu, image miniature, catégorie, étiquettes et date de publication.
 - Pagination sur la liste des articles.
 - Affichage des commentaires associés à un article.
 - Possibilité pour les utilisateurs connectés de laisser des commentaires.
 - Recherche d'articles par mots-clés dans le titre ou le contenu.
 - Filtrage des articles par catégorie ou par étiquette via des pages dédiées.

3.2. Gestion du contenu (Back-End)

- **Interface d'administration (pour rôle Admin)** :
 - Tableau de bord listant tous les articles avec options de visualisation, modification et suppression.
 - Formulaire de création et de modification d'articles permettant de définir :

- Titre, Slug (génééré ou manuel), Contenu.
- Téléversement d'une image miniature (thumbnail).
- Assignment d'une catégorie (obligatoire).
- Assignment d'une ou plusieurs étiquettes (optionnel).
- Suppression sécurisée d'un article

3.3. Gestion des utilisateurs

- **Comptes utilisateurs :**
 - Inscription de nouveaux utilisateurs (Nom, Email, Mot de passe).
 - Connexion des utilisateurs existants.
 - Déconnexion.
 - Page "Mon Compte" pour modifier son propre mot de passe.
- **Rôles et Autorisations :**
 - Deux rôles définis : Utilisateur standard (Default) et Administrateur (Admin).
 - Le rôle Administrateur donne accès à la section /admin/* via un middleware dédié.

3.4. Autres fonctionnalités

- **Optimisation SEO :** Utilisation de slugs uniques et lisibles pour les URLs des articles, catégories et tags.
- **Sécurité :** Utilisation des mécanismes de sécurité de Laravel (protection CSRF, hashage des mots de passe, protection contre les injections SQL via Eloquent).

4. Contenu

- **Données gérées :**
 - Utilisateurs (nom, email, mot de passe hashé, rôle).
 - Articles (titre, slug, extrait, contenu, miniature, catégorie_id).
 - Catégories (nom, slug).
 - Étiquettes (nom, slug).
 - Commentaires (contenu, user_id, post_id).

- Table pivot post_tag pour la relation ManyToMany.

5. Contraintes techniques

- **Technologies :**

- Back-end : PHP 8.1+, Laravel 10.
- Front-end : HTML5, CSS3 (Tailwind CSS), JavaScript (Alpine.js).
- Base de données : MySQL.
- Outil de build : Vite.

- **Hébergement :**

- Compatible avec les serveurs supportant PHP et MySQL.
- Serveur Web (Apache, o2switch).

- **Tests & compatibilité :**

- Compatibilité navigateur visée : versions récentes de Chrome, Firefox, Safari, Edge.

6. Maintenance et évolutivité

- **Maintenance corrective :**

- Suivi des erreurs via les logs Laravel.
- Mises à jour régulières de Laravel et des dépendances pour la sécurité et les correctifs.

- **Maintenance évolutive :**

- Ajout possible de fonctionnalités : gestion des catégories/tags par l'admin, système de recherche plus avancé, gestion des utilisateurs plus fine, API REST pour des clients externes, etc.
- Structure Laravel favorisant l'évolutivité.

7. Validation du projet

- **Contexte :** Projet réalisé dans un cadre de formation/apprentissage.