# **Cahier des Charges : Projet Mini Blog**

## **Description du projet**

Ce projet vise à développer une application web simple permettant aux utilisateurs de publier et consulter des articles. Les fonctionnalités principales incluent l'ajout, la visualisation et la lecture détaillée d'articles. L'application sera conçue pour être intuitive, légère et accessible sur différentes plateformes.

## **Analyse des exigences**

### **Objectifs clés :**

* Créer une plateforme simple et conviviale pour publier et lire des articles.
* Assurer une navigation fluide et rapide.
* Permettre une gestion efficace des articles.

### **Cibles principales :**

* Blogueurs amateurs.
* Utilisateurs souhaitant partager des idées rapidement.
* Lecteurs cherchant des articles courts et intéressants.

### **Tendances du marché :**

* Forte préférence pour des interfaces épurées et faciles à utiliser.
* Intérêt croissant pour des plateformes minimalistes sans publicités invasives.
* Importance de la compatibilité mobile pour répondre aux usages actuels.

## **Analyse des solutions**

### **Choix technologiques :**

* **Backend** : Flask (Python) pour sa simplicité et son efficacité dans la création d'API RESTful.
* **Frontend** : React pour une interface utilisateur réactive et moderne.
* **Base de données** : PostgreSQL pour sa robustesse et sa gestion avancée des données relationnelles.

### **Avantages de la solution proposée :**

* Technologie légère et adaptée aux besoins d’un projet minimaliste.
* Facilité de développement et de maintenance.
* Possibilité d’évolution future pour intégrer des fonctionnalités supplémentaires.

## **Besoins fonctionnels**

Les fonctionnalités suivantes doivent être incluses :

### **Gestion des articles :**

* Création d’un article avec un titre et un contenu.
* Visualisation de la liste des articles publiés.
* Lecture détaillée d’un article individuel.

### **Navigation :**

* Interface utilisateur claire et intuitive.
* Possibilité de trier ou rechercher les articles (optionnel).

### **Accessibilité :**

* Design responsive pour une utilisation optimale sur desktop, tablette et mobile.

## **Besoins non fonctionnels**

### **Performance :**

* Temps de réponse inférieur à 500 ms pour les opérations courantes.
* Optimisation des requêtes pour gérer plusieurs utilisateurs simultanés.

### **Sécurité :**

* Protection des données utilisateurs via HTTPS.
* Validation des données saisies pour éviter les failles XSS et SQL Injection.

### **Scalabilité :**

* Prévoir une base de données évolutive si le volume d’utilisateurs augmente.

### **Modularité :**

* Code structuré pour permettre l’ajout futur de fonctionnalités comme les commentaires ou le partage d’articles.

## **Exigences techniques**

### **Backend :**

* Framework Flask pour gérer les API RESTful.
* Gestion de la base de données avec SQLAlchemy.

### **Frontend :**

* React pour une interface utilisateur moderne.
* Utilisation de Bootstrap ou Tailwind CSS pour un design responsive.

### **Base de données :**

* PostgreSQL pour sa robustesse et sa gestion avancée des données relationnelles.
* Structuration efficace pour gérer les articles et utilisateurs.

## **Livrables**

### **Application fonctionnelle :**

* Frontend complet et connecté au backend.
* Backend avec des API RESTful documentées.

### **Documentation technique :**

* Guide d’installation et de déploiement.
* Documentation des API.

### **Code source :**

* Bien commenté pour faciliter la maintenance.
* Conforme aux bonnes pratiques de développement.

Ce cahier des charges servira de base pour le développement et le suivi du projet Mini Blog.