

# Cahier des Charges – Projet Full Stack Web

## 1. Contexte général

Un investisseur veut se lancer dans un nouveau projet innovant qui vise à réaliser une application Full Stack moderne. Vous considérant des associés de confiance pour son futur projet, il vous délègue les tâches pour concevoir et développer cette application en respectant une architecture client–serveur.

Le projet doit permettre de gérer un ensemble d'objets ou d'entités (ex. utilisateurs, produits, tâches, réservations, articles, etc.), à travers une interface dynamique et un backend RESTful. Le domaine d'application est libre, à condition qu'il soit cohérent et permette d'illustrer toutes les

## 2. Fonctionnalités

### Authentification & Sécurité

- Inscription / Connexion avec hachage de mot de passe (JWT ou session).
- Protection des routes sensibles.
- Gestion des rôles (admin, user).

### CRUD complet sur l'entité principale

- Créer, Lire, Mettre à jour, Supprimer des données via une API RESTful.
- Validation côté client et serveur.
- Messages d'erreur clairs et statuts HTTP adaptés.

### Recherche, filtrage et tri

- Recherche par mots-clés et filtrage dynamique.
- Tri ascendant/descendant selon des critères.

### Pagination

- Chargement paginé des données pour améliorer la performance.

## **Interface utilisateur (Frontend)**

- Pages : Accueil, Liste, Détail, Formulaires, Connexion.
- Navigation fluide sans rechargement complet.
- Design responsive avec Bootstrap, TailwindCSS ou Material UI.

## **Validation des données**

- Validation des champs requis, formats et contraintes.
- Messages d'erreur utilisateur explicites.

## **Upload de fichiers (optionnel)**

- Possibilité d'ajouter des images ou documents.
- Aperçu et gestion sécurisée des fichiers.

## **Tableau de bord / statistiques (optionnel)**

- Affichage de statistiques simples (nombre, graphiques, répartition).

## **Déploiement**

- Backend sur Render, frontend sur Vercel/Netlify.
- Connexion front-back configurée avec variables d'environnement.

## **3. Technologies**

- Frontend : React.js, Angular ou Next.js
- Backend : Node.js (Express) ou Spring Boot
- Base de données : MongoDB, PostgreSQL ou MySQL
- Déploiement : Render, Vercel ou Netlify

## **4. Livrables attendus**

- Dépôt GitHub public contenant le code source.
- Documentation technique (schéma BD, diagramme de classes, séquence).
- Rapport avec instructions, captures et endpoints API.
- Lien de déploiement du frontend et backend.
- Vidéo de présentation (3 à 5 minutes).