Cahier des charges

1. Contexte du projet

Cette application web a pour objectif de (décrire l'objectif principal).

Elle fournira aux utilisateurs (les principales fonctionnalités), le tout présenté à travers une interface ergonomique.

Exemple : L'objectif est de rationaliser la gestion des projets d'équipe en consolidant les tâches, les conversations et les ressources au sein d'une interface partagée.

2. Objectifs

• Accessibilité : L'application est conçue pour fonctionner sur les principaux navigateurs web (Chrome, Firefox, Safari,) et est optimisée pour les téléphones mobiles et les tablettes.

• Facilité d'utilisation : Une interface intuitive est essentielle, éliminant le besoin de formation approfondie.

• Protection : Des protocoles sécurisés (HTTPS, authentification renforcée) protégeront les données des utilisateurs.

• Croissance : L'application doit s'adapter à une augmentation substantielle de la base d'utilisateurs.

3. Fonctionnalités principales

3.1 Interface utilisateur (UI)

• Page d'accueil : Affichage des fonctionnalités principales ainsi que des points d'accès rapides.

• Tableau de bord : Résumé des tâches, des discussions et des informations pertinentes supplémentaires.

• Notifications : Système d'alertes.

• Multilingue : Offert en plusieurs langues (comme le français, l'anglais, etc.).

3.2 Gestion des utilisateurs

• Connexion/Inscription : Les utilisateurs peuvent s'inscrire en utilisant leur e-mail ou via les réseaux sociaux tels que Google, Facebook et autres.

• Gestion des informations personnelles : Profils d'utilisateurs.

• Gestion des niveaux d'accès : Système de rôles (administrateur, utilisateur, invité).

• Récupération de mot de passe : Une fonctionnalité conçue pour récupérer l'accès lorsqu'un mot de passe est oublié.

3.3. Caractéristiques distinctes

La gestion de projet implique le développement et la supervision de projets qui incluent diverses sous-tâches. La messagerie interne fait référence à un système de chat en temps réel qui facilite la communication entre les utilisateurs.

Partage de fichiers : les utilisateurs auront la possibilité de télécharger et de distribuer des documents.

Agenda : une fonction de calendrier pour superviser les délais, les réunions et les événements.

Statistiques : un tableau de bord affichant des données sur l'utilisation de l'application et les activités du projet.

3.4. Fonctionnalités améliorées

La fonctionnalité de recherche améliorée permet aux utilisateurs d'explorer les projets, les tâches et les discussions. Les utilisateurs peuvent participer à des commentaires et à des discussions liées aux tâches.

Intégrations avec des outils tiers : connexion à des applications externes comme Google Drive, Slack et autres.

4. Limitations de la technologie.

4.1. Technologies et langues

Frontend : Utilisation des technologies web modernes (HTML5, CSS3, JavaScript, React, Vue.js ou framework Angular).
Backend : L'application utilisera un serveur en Node.js, Python (Django, Flask), PHP (Laravel) ou autre technologie définie en fonction des besoins.
Base de données : Utilisation d'une base de données relationnelle (MySQL, PostgreSQL) ou NoSQL (MongoDB, Firebase).
API REST/GraphQL : l'application doit proposer une API pour permettre l'intégration avec d'autres services.

4.2. Hébergement et infrastructures

L'application sera hébergée sur un serveur cloud (AWS, Google Cloud, Azure ou autre).
Utiliser un CDN (content delivery network) pour améliorer les performances.
Sauvegardez régulièrement vos données et mettez en œuvre un plan de reprise après incident.

4.3. Sécurité

Une connexion sécurisée à internet.
La gestion des droits d'accès des utilisateurs via des tokens (JWT).
Les mots de passe sont stockés sous forme cryptée (bcrypt ou autre).
La protection contre les attaques courantes (injection SQL, XSS, CSRF).

5. Performance et évolutivité

L'application doit avoir la capacité de supporter au moins X utilisateurs simultanés.
Les temps de chargement doivent être inférieurs à 3 secondes pour chaque page.
Des tests de charge doivent être réalisés pour évaluer la stabilité du système lors d'une utilisation intensive.

6. Design

Charte écrite : La rédaction doit suivre la charte écrite que le client a fournie.
UI/UX : L'accent sera mis sur une interface intuitive et minimisée tout en respectant l'accessibilité des composants de conception les plus courants.
Responsive : L'interface doit pouvoir s'adapter à différentes tailles d'écran.

7. Budget prévisionnel

Estimation des dépenses de développement backend : [Montant].
Estimation des dépenses de développement frontend : [Montant].
Coût de la location annuelle : [Montant].
Coût de maintenance (annuel ou mensuel) : [Montant].

8. Évolutions futures

La conception de l'application doit favoriser l'ajout de nouvelles fonctionnalités dans le futur, telles que :

Le développement d'une application native pour appareils mobiles (iOS/Android).
L'intégration de fonctionnalités issues de l'intelligence artificielle pour améliorer la productivité (par exemple, des suggestions automatiques).

9. Programmation et délais

Phase 1 : Recueil des besoins et finalisation du cahier des charges [durée estimée].
Phase 2 : Développement du backend et des fonctionnalités initiales - [Durée estimée].
Phase 3 : Développement du frontend et intégration UI/UX - [Durée estimée].
Phase 4 : Tests unitaires, d'intégration et de validation - [Durée estimée].
Phase 5 : Mise en production et préparation - [Durée estimée].