

Cahier des charges pour l'application "Office Master"

Pour : Aglan yao elom

1. Contexte du projet

Office Master est une application de gestion administrative visant à centraliser et automatiser les processus de gestion des employés, des documents, des réunions, et des demandes. Elle est destinée à améliorer l'efficacité et la productivité des entreprises en fournissant une interface simple et intuitive pour l'ensemble de ces tâches administratives.

2. Objectifs du projet

1. Centraliser les processus de gestion administrative dans une seule plateforme.
 2. Fournir une interface utilisateur ergonomique pour faciliter l'utilisation de l'application.
 3. Automatiser les tâches récurrentes telles que la gestion des documents, des présences et des demandes de congé.
 4. Assurer la sécurité des données des employés, notamment via un système d'authentification sécurisé.
 5. Mettre en place une fonctionnalité de messagerie interne pour faciliter la communication entre les employés.
-

3. Fonctionnalités principales

3.1 Gestion des employés

- **Ajout, modification et suppression d'employés** : Permet de gérer la base de données des employés avec des champs comme le nom, prénom, département, poste, horaire, email, etc.
- **Visualisation des employés** : Liste des employés avec possibilité de rechercher et filtrer selon différents critères.
- **Attribution de rôles** : Assignment de rôles (ex : employé, manager, administrateur) avec des niveaux d'accès différents.

3.2 Gestion des documents

- **Téléversement de documents** : Les utilisateurs peuvent ajouter des documents avec une catégorie associée.
- **Affichage et téléchargement** : Les utilisateurs peuvent visualiser et télécharger des documents enregistrés dans la base de données.
- **Suppression de documents** : Possibilité de supprimer un document aussi bien de la base de données que du serveur.

3.3 Gestion des demandes

- **Demandes variées** : Possibilité de faire des demandes (congé, absence, réclamation, heures supplémentaires, recrutement, stage, démission).
- **Suivi des demandes** : Historique et suivi des demandes faites par les employés, avec statut (en cours, approuvée, rejetée).
- **Validation des demandes** : Interface dédiée pour les responsables permettant de traiter les demandes des employés.

3.4 Système de gestion des réunions

- **Planification et envoi d'invitations automatiques** : Outil pour gérer et automatiser la planification des réunions et des événements officiels.
- **Compte-rendu automatisé** : Système de création et partage des comptes rendus avec un suivi des actions.

3.5 Système de messagerie interne

- **Messagerie entre employés** : Les utilisateurs peuvent envoyer des messages à leurs collègues ou managers directement dans l'application.
- **Historique des conversations** : Possibilité de conserver un historique des conversations entre utilisateurs.

3.6 Paramètres et gestion des rôles

- **Gestion des rôles** : Création et modification des rôles au sein de l'application.
- **Accès par rôle** : Définir les droits d'accès aux différentes sections de l'application en fonction des rôles.
- **Personnalisation des paramètres** : Option pour ajuster certaines configurations de l'application (ex : langue, notifications).

3.7 Authentification et sécurité

- **Inscription et connexion** : Les employés peuvent s'inscrire et se connecter via une page dédiée avec gestion des sessions.
 - **Système JWT** : Utilisation de JSON Web Tokens (JWT) pour sécuriser l'authentification.
 - **Mot de passe sécurisé** : Stockage des mots de passe en utilisant des techniques de hachage (ex : bcrypt).
-

4. Contraintes techniques

4.1 Technologies à utiliser

- **Front-end** : React.js avec Tailwind CSS pour un design moderne et réactif.
- **Back-end** : Node.js avec Express pour la gestion des API.
- **Base de données** : MySQL pour la gestion des données des employés, des documents et des demandes.
- **Authentification** : JWT pour sécuriser l'accès à l'application.
- **Téléversement de fichiers** : Multer pour la gestion des fichiers téléversés (CV, documents, etc.).

4.2 Performance

- L'application doit être optimisée pour un grand nombre d'utilisateurs simultanés.
- La récupération et le traitement des données doivent être rapides (chargement rapide des employés, documents et demandes).

4.3 Compatibilité

- L'application doit être compatible avec les navigateurs modernes (Chrome, Firefox, Safari, Edge).
 - Compatible avec les appareils mobiles (tablettes et smartphones).
-

5. Sécurité

- **Protection des données** : Toutes les données sensibles (mot de passe, documents confidentiels) doivent être chiffrées.

- **Gestion des accès** : Chaque utilisateur doit avoir accès uniquement aux sections de l'application correspondant à son rôle.
 - **Sauvegardes** : Mécanisme de sauvegarde des données pour éviter la perte en cas de panne.
-

6. Interface utilisateur

L'interface utilisateur doit être :

- **Simple et intuitive** : Accessible aux utilisateurs non techniques avec des actions claires.
 - **Responsive** : S'adapter à tous les types d'appareils (ordinateurs de bureau, tablettes, téléphones mobiles).
 - **Moderne et esthétique** : Utilisation de Tailwind CSS pour un design professionnel et élégant.
-

7. Étapes du projet et livrables

1. **Analyse des besoins et élaboration du cahier des charges** (2 semaines)
 2. **Développement du front-end** (3 semaines)
 - Mise en place de React avec Tailwind CSS.
 - Création des pages (connexion, inscription, tableau de bord, gestion des employés, gestion des documents).
 3. **Développement du back-end** (3 semaines)
 - Création des routes API avec Node.js et Express.
 - Connexion à la base de données MySQL.
 4. **Implémentation de la messagerie interne** (1 semaine)
 5. **Tests et corrections** (2 semaines)
 6. **Livraison finale** (1 semaine)
-

8. Suivi et maintenance

- **Support technique** : Une période de support de 3 mois après la livraison pour corriger les éventuels bugs.
- **Mises à jour** : Mises à jour régulières pour l'amélioration de la sécurité et l'ajout de nouvelles fonctionnalités