

Projet REACT GLSI

1. Fonctionnalités principales

Frontend (React) :

- **Authentification** : Connexion pour les employés et les administrateurs RH.
- **Tableau de bord personnalisé** :
 - **Employés** : Consultation des informations personnelles, suivi des congés, gestion des demandes de congé, affichage des évaluations de performance.
 - **Administrateurs RH** : Gestion des employés, des congés, des évaluations et suivi des feuilles de temps.
- **Gestion des congés** : Système pour demander, approuver ou refuser les congés, avec affichage du solde de congés restants.
- **Gestion des évaluations de performance** : Permettre aux RH d'évaluer les employés, et aux employés de consulter leurs évaluations.
- **Feuilles de temps** : Les employés peuvent saisir leur temps de travail quotidien, et les RH peuvent les valider.
- **Notifications** : Alertes pour les événements importants comme l'approbation de congé, des évaluations ou des rappels de feuille de temps.
- **Recherche et filtrage** : Rechercher des employés ou filtrer les congés par type (maladie, vacances, etc.).
- **Exportation de données** : Exporter des rapports de présence, des évaluations de performance ou des congés en format Excel ou PDF.

Backend (springBoot Or nodejs) :

- **Gestion des utilisateurs (Employés/RH)** : CRUD pour les employés avec rôles (employé, RH, admin), gestion des authentifications (via JWT).
- **Gestion des congés** : CRUD pour les demandes de congé, avec système d'approbation par les RH.
- **Gestion des évaluations** : CRUD pour les évaluations de performance, système de notation par les RH et affichage des résultats pour les employés.
- **Gestion des feuilles de temps** : CRUD pour enregistrer les heures de travail et suivi des validations par les RH.
- **Notifications** : Système de notifications pour les demandes de congé, évaluations et rappels.
- **Reporting** : Génération de rapports statistiques sur les employés (nombre de congés, performance, heures travaillées, etc.).
- **Sécurité(optional)** : Mise en place de protections de sécurité (XSS, CSRF) et d'une gestion des permissions en fonction des rôles.

2. Technologies à utiliser

- **Frontend (React)** :
 - React.js avec Hooks pour la gestion de l'État.
 - Redux ou Context API pour la gestion des utilisateurs (RH, employés).

- Axios pour les appels API.
- Tailwind CSS ou Ant Design pour l'interface utilisateur.
- Formik pour la gestion des formulaires (demandes de congé, évaluations).
- **Backend (springBoot Or nodejs) :**
 - TypeORM pour la gestion des bases de données (MySQL).
 - Passport.js avec JWT pour la gestion des sessions et des autorisations.
 - Multer pour l'upload des fichiers (comme les documents liés aux congés).
 - Rate limiting pour sécuriser l'API.
 - Bull (cron) ou une autre bibliothèque de gestion de tâches pour envoyer des notifications automatiques (par exemple, par e-mail pour les évaluations ou les demandes de congé).

3. Étapes de développement

1. **Mise en place du projet :**
 - Créer un projet React avec create-react-app ou Vite.
 - créer projet côté Serveur
 - Mettre le projet dans github
2. **Authentification et rôles :**
 - Backend : Création du système d'authentification avec JWT et gestion des rôles (employé, RH, admin).
 - Frontend : Page de connexion et redirection basée sur le rôle de l'utilisateur (RH ou employé).
3. **Gestion des employés :**
 - Backend : API CRUD pour la gestion des employés.
 - Frontend : Interface pour que les RH puissent voir, ajouter, modifier ou supprimer des employés.
4. **Gestion des congés :**
 - Backend : API pour les demandes de congé, système d'approbation par les RH.
 - Frontend : Interface pour demander des congés, voir le solde restant, et pour les RH d'approuver ou rejeter les demandes.
5. **Feuilles de temps :**
 - Backend : API pour la gestion des heures travaillées (saisie par les employés et validation par les RH).
 - Frontend : Tableau pour saisir les heures travaillées, avec validation et possibilité de correction.
6. **Évaluations de performance :**
 - Backend : API pour les évaluations annuelles ou périodiques avec un système de notation.
 - Frontend : Interface pour les RH pour ajouter des évaluations et pour les employés de les consulter.
7. **Notifications et reporting :**
 - Backend : Implémentation des notifications via des WebSockets ou des emails.
 - Frontend : Interface pour consulter les notifications et afficher les rapports (feuilles de temps, performances).

8. **Recherche et filtrage :**

- Backend : Création d'une API pour rechercher et filtrer les employés ou les congés.
- Frontend : Barre de recherche et filtres pour simplifier la gestion des employés et des demandes de congé.

9. **Déploiement** (optional):

- Frontend : Hébergement sur Vercel ou Netlify.
- Backend : Déploiement du backend sur Heroku, AWS ou DigitalOcean.
- Base de données : Utilisation d'une base de données cloud comme PostgreSQL.