**Cahier des charges — Démo de l’application RH (React/NestJS/MongoDB)**

**1) Contexte & objectif**

Mettre en démonstration un tableau de bord RH dédié aux salariés d’une PME, couvrant la gestion des utilisateurs, des congés, des documents, des projets/sprints/tâches (Kanban), du recrutement (avec module IA), des événements et des notifications temps réel.

**2) Périmètre fonctionnel (démo)**

* **Authentification & rôles** : Login JWT, rôles Admin, RH, Manager, Employé (RBAC).
* **Employés (Users)** : Liste + recherche, création/édition/suppression, affectation rôle/équipe.
* **Congés** : Demande avec pièce jointe, workflow *En attente → Approuvé/Refusé*, calendrier, “Who’s on leave”, historique, types de congés (Tunisie).
* **Documents RH** : Génération/gestion (ex. attestation), statut (à traiter/validé), date limite de traitement, aperçu/téléchargement.
* **Recrutement** : Offres d’emploi, candidatures, matching compétences (IA) avec score, filtre par catégorie/type.
* **Projets / Sprints / Tâches** : Vue projets, sprints, Kanban drag & drop, estimation/temps, recherche.
* **Événements** : Agenda (react-big-calendar), création d’événements/salles.
* **Dashboard & KPIs** : Widgets (compteurs, tendances), prochaines fêtes/jours fériés (Tunisie).
* **Notifications** : Temps réel via WebSockets (demandes approuvées, documents à traiter, etc.).

**3) Exigences non fonctionnelles**

* **UX/UI** : Material UI, responsive desktop/tablette, i18n (fr par défaut).
* **Performance** : Listes paginées, recherche serveur, chargement paresseux des médias.
* **Sécurité** : JWT + refresh, RBAC, validation DTO, protections basiques (rate-limit, CORS).
* **Qualité** : Linting, structure modulaire, tests unitaires ciblés (services/utile).

**4) Architecture technique**

* **Frontend** : React 18 + Vite, Material UI, Redux Toolkit (store, slices), React Router, i18next.
* **Backend** : NestJS (Modules/Controllers/Services), Mongoose, class-validator, WebSockets (gateway).
* **Base de données** : MongoDB (schemas : User, Leave, LeaveType, Document, JobOffer, Application, Project, Sprint, Task, Event, Notification).
* **Stockage fichiers** : Upload justificatifs/documents (ex: dossier uploads/ ou GridFS).
* **Intégrations** : (Option démo) import CSV, envoi mail/Toast pour notifications.

**5) Modèles de données (extraits)**

* **User** : \_id, nom, email, rôle, équipe, avatar, actif.
* **Leave** : \_id, employé, type, dates, statut, motif, pièceJointe, historique.
* **Document** : \_id, employé, type, métadonnées, statut, deadline, fichier.
* **JobOffer/Application** : titre, catégorie, type contrat, compétences; CV, score, statut.
* **Project/Sprint/Task** : liens hiérarchiques, états, estimation/temps, assignees.
* **Event** : titre, salle, date/heure, participants.

**6) API principales (exemples)**

* **Auth** : POST /auth/login, POST /auth/refresh.
* **Users** : GET /users?search=, POST /users, PATCH /users/:id, DELETE /users/:id.
* **Leaves** : GET /leaves, POST /leaves (upload), PATCH /leaves/:id/approve|reject.
* **Documents** : GET /documents?status&deadline, POST /documents, PATCH /documents/:id.
* **Jobs** : GET /joboffre, POST /joboffre, GET /applications, POST /applications, POST /applications/:id/score.
* **Projects/Sprints/Tasks** : GET /projects, POST /tasks/move (Kanban).
* **Events** : GET/POST /events.
* **Notifications** : WS notifications (subscribe/broadcast).

**7) Jeux de données (seed démo)**

* 1 Admin, 1 RH, 2 Managers, 6 Employés (équipes A/B).
* Types de congés (Tunisie) : Annuel, Maladie, Maternité, Paternité, Mariage, Décès, Sans solde.
* 3 Offres d’emploi (Catégories IT), 6 candidatures avec CV factices.
* 2 Projets, 3 sprints, ~12 tâches (todo/doing/done).
* 6 Documents RH (dont 3 “à traiter” avec deadline proche).
* 5 Événements (réu/formation), 5 notifications prêtes.

**8) Parcours de démo (5–8 min)**

1. **Login RH** → tableau de bord (KPIs + notifications).
2. **Congés** : Filtre + approbation d’une demande (toast + WS).
3. **Documents** : Tri par deadline, validation d’un document, téléchargement.
4. **Recrutement** : Ouverture d’une offre, scoring IA d’une candidature, tri par score.
5. **Projets/Tâches** : Drag & drop d’une tâche entre colonnes, recherche par titre.
6. **Événements** : Ajout rapide d’un événement, affichage calendrier.
7. **Employés** : Recherche + édition rapide d’un profil.
8. **Switch Employé** (ou session simulée) : Soumission d’une demande de congé avec pièce jointe → retour RH (workflow OK).

**9) Critères d’acceptation (extraits)**

* RBAC appliqué (chaque rôle voit ce qu’il doit).
* Workflow congé complet (création, décision, historique, notification).
* Tri/filtre documents par **date limite** opérationnel.
* Kanban persiste l’ordre/colonne côté serveur.
* Scoring IA visible (valeur de score + tri).
* Événements affichés en calendrier (fr).
* Recherche globale fonctionnelle (Users, Tasks, Jobs).

**10) Déploiement (démo)**

* Variables d’environnement (.env) : DB\_URI, JWT\_SECRET, UPLOAD\_PATH, WS\_ORIGIN.
* Lancement : npm run start:prod (API) / vite build && preview (Front) ou docker-compose (api, mongo, front).

**11) Risques & limites (démo)**

* IA en mode “lite” (modèle local/simplifié, pas de cloud).
* Envoi email optionnel remplacé par toasts/WS.
* Données seed non sensibles, anonymisées.

**12) Planning (démo)**

* J-3 : gel des features, seed final.
* J-2 : tests de parcours + perf listes.
* J-1 : répétition, sauvegarde/exports.

**Livrables démo** : build front, API Nest en marche, dump/seed Mongo, comptes de test, script de scénario (ci-dessus).