

JobPortalX – Plateforme de gestion des offres d’emploi

## JobPortalX est une application web qui permet aux entreprises X de publier leurs offres d’emploi et de gérer les candidatures, tout en offrant aux candidats la possibilité de créer un compte, postuler aux offres et suivre l’état de leurs candidatures. Les RH peuvent gérer les employés, les offres, et décider de l’état des candidatures, tandis que l’administrateur valide les comptes RH et consulte les statistiques via un tableau de bord interactif. Cette plateforme centralise et simplifie le processus de recrutement pour toutes les parties prenantes.



01/10/2025 - 31/10/2025

Réalisé avec : ITGust-GustTrainig

## ****Nom de l’application :****

**JobPortalX**

Le nom **JobPortalX** se décompose ainsi :

* **Job** → signifie « emploi » ou « travail ».
* **Portal** → signifie « portail » ou « site permettant d’accéder à des services ou informations ».
* **X** → référence au nom d’entreprise.

Donc **JobPortalX** peut se traduire librement par : « Plateforme d’offres d’emploi X ».

## ****Cahier des charges :****

### **1. Objectif**

Créer une application web pour gérer les offres d’emploi, permettant aux entreprises de publier des offres, aux candidats de postuler, et aux RH et administrateurs de gérer les comptes et les candidatures.

### **2. Acteurs et profils**

|  |  |
| --- | --- |
| **Profil** | **Rôle principal** |
| **Utilisateur / Candidat** | Consulter les offres, créer un compte, postuler aux offres, suivre l’état de ses candidatures |
| **RH / Recruteur** | Créer un compte (validé par l’admin), gérer les offres d’emploi, gérer les candidats, accepter ou refuser les candidatures |
| **Admin** | Valider les comptes RH, consulter les statistiques, gérer l’accès et les droits, dashboard global |

### **3. Fonctionnalités par profil**

#### **3.1 Utilisateur / Candidat**

* Créer un compte et se connecter.
* Modifier son profil (CV, informations personnelles).
* Consulter la liste des offres publiées.
* Filtrer les offres par catégorie, lieu, type de contrat.
* Postuler à une offre.
* Suivre l’état de ses candidatures (en attente, acceptée, refusée).

#### **3.2 RH / Recruteur**

* Créer un compte RH (doit être validé par l’admin avant login).
* Se connecter après validation par l’admin.
* Ajouter, modifier, supprimer les offres d’emploi.
* Consulter les candidatures reçues.
* Accepter ou refuser une candidature (changer l’état de la candidature).
* Consulter la liste des employés gérés (si applicable).

#### **3.3 Admin**

* Se connecter avec des droits d’administration.
* Valider ou refuser les comptes RH.
* Accéder à un **dashboard** pour :
  + Statistiques sur les offres publiées, candidatures reçues, taux d’acceptation.
  + Nombre de candidats et RH actifs.
* Gérer les accès et droits.
* Consulter l’historique des activités.

### **4. Architecture et technologies proposées**

|  |  |
| --- | --- |
| **Composant** | **Technologie suggérée** |
| Frontend | React.js, Tailwind CSS, React Router |
| Backend | Node.js + Express ou NestJS |
| Base de données | PostgreSQL ou MongoDB |
| Authentification | JWT ou OAuth2 |
| Stockage CV | AWS S3 / Cloud storage |
| Dashboard | Recharts / Chart.js pour les statistiques |

### **5. Sécurité**

* Validation et filtrage côté serveur des entrées utilisateurs.
* Mots de passe stockés en hash sécurisé (bcrypt).
* Contrôle des accès par rôle (Utilisateur, RH, Admin).
* Vérification des comptes RH avant connexion.

### **6. Workflow général**

1. **Utilisateur** crée un compte → peut consulter les offres et postuler.
2. **RH** crée un compte → admin valide le compte → RH peut se connecter et gérer les offres et candidatures.
3. **Admin** supervise la plateforme → valide les comptes RH → consulte les statistiques et gère les accès.

### **7. Bonus / fonctionnalités futures**

* Notifications par e-mail pour nouvelles offres ou changements d’état de candidature.
* Recherche avancée par mots-clés.
* Système de commentaires ou feedback pour les candidatures.
* Mobile responsive pour tous les profils.

# Architecture générale

Frontend (React) Backend (Node.js/Express)

------------------------------- ------------------------

- Pages / Routes - REST API / GraphQL

- Composants - Authentification JWT

- React Router - Contrôle d’accès par rôle

- Tailwind CSS / UI - Gestion utilisateurs (candidats, RH, admin)

- Axios / fetch API - Gestion offres et candidatures

- Statistiques pour dashboard

Database (PostgreSQL / MongoDB)

--------------------------------------------------

- Users (candidats, RH, admin)

- Jobs (offres d’emploi)

- Applications (candidatures)

- Roles & Permissions

# *Projet Frontend (******React App******)*

## Créer le projet

1. **Ouvrir le terminal**.
2. **Créer le projet** :

Autoriser l’exécution des scripts PowerShell (si erreur npx : Impossible de charger le fichier)

Set-ExecutionPolicy -Scope Process -ExecutionPolicy Bypass

Créer le projet React

npx create-react-app frontend

1. **Entrer dans le dossier du projet** :

cd frontend

1. **Lancer le projet** :

npm start

L’application sera accessible sur http://localhost:3000.

## ****Installer des outils utiles****

Installer la bibliothèque de routage

npm install react-router-dom

Permet de gérer les routes pour les pages candidats, RH, admin, etc.

## ****Extensions utiles pour VS Code****

* **ES7+ React/Redux/React-Native snippets**

Permet de générer rapidement des composants, hooks et autres snippets.

* + Exemple : rafce → crée automatiquement un **functional component** avec export.

# Projet Backend (Node.js/Express)