# Cahier des charges : Système centralisé de recommandation d’offres d’emploi

## Introduction

Dans un marché du travail en constante évolution, en particulier dans le secteur de la data et du numérique, les chercheurs d'emploi sont confrontés à la dispersion des offres sur plusieurs plateformes. Cette situation complexifie le processus de recherche et réduit l’efficacité du matching entre les candidats et les employeurs.

Ce projet vise à développer un système centralisé de recommandation d’offres d’emploi, permettant de simplifier la recherche et d’optimiser la correspondance entre les profils des candidats et les offres disponibles

## Contexte et enjeux

L’émergence de nouvelles compétences et la transformation numérique du marché du travail exigent une approche plus efficace pour mettre en relation les recruteurs et les chercheurs d’emploi. Les plateformes classiques ne permettent pas une agrégation optimale des offres, ce qui nuit à la visibilité des opportunités et complique la tâche des candidats.

Le projet répond donc à ce besoin en offrant un système intelligent capable de centraliser, traiter et recommander automatiquement les offres les plus pertinentes aux utilisateurs.

## 2. Objectifs généraux

* Centralise automatiquement les offres d'emploi issues de plusieurs sources.
* Implémente un algorithme de recommandation basé sur les compétences et préférences des utilisateurs.
* Améliore l’expérience utilisateur grâce à une interface intuitive et efficace.
* Optimise le matching entre offres et candidats pour faciliter l’accès aux opportunités.

## 3. Périmètre du projet

* Centralisation des offres d’emploi : agréger automatiquement des offres provenant de diverses sources (sites web, API publiques, etc.).
* Système de stockage : mettre en place une base de données centralisée et structurée.
* Système de recommandation : développer un algorithme adapté aux besoins des utilisateurs (basé sur les compétences, l’expérience et les préférences).
* Interface utilisateur : concevoir une plateforme intuitive pour les chercheurs d’emploi et les recruteurs.

## 4. Analyse des besoins

### 4.1 Utilisateurs cibles

* **Chercheurs d’emploi :** professionnels et étudiants du secteur de la data et du numérique.
* **Recruteurs :** entreprises et cabinets de recrutement spécialisés.

### 4.2 Besoins fonctionnels

* Collecte automatique des offres d’emploi de diverses plateformes.
* Affichage clair et filtrable des offres centralisées.
* Mettre en place un système de recommandation basé sur l’IA.
* Assurer une mise à jour automatique des offres.
* Possibilité pour les recruteurs de publier directement leurs offres.

### 4.3 Besoins non fonctionnels

* Système rapide, résilient et scalable.
* Base de données performante pour stocker et gérer les offres et les profils.
* API pour récupérer les offres depuis différentes plateformes.

## 5. Contraintes

* Date limite : WP1.0 et WP1.1 doivent être terminés pour le dimanche 23 février 2025.
* Budget limité : utilisation préférable de technologies open source.
* Conformité aux régulations RGPD pour le traitement des données personnelles.

## 6. Planification et livrables

| **Étape** | **Description** | **Livrable** |
| --- | --- | --- |
| Analyse des besoins | Identifier les besoins fonctionnels et non fonctionnels. | Rapport d’analyse |
| Design et architecture | Conception de l’interface utilisateur et architecture du système. | Diagrammes d’architecture et maquettes |
| Ingestion de données (WP1.1) | Mise en place des processus d’agrégation et de stockage des offres. | Scripts d’ingestion et base de données |
| Développement du système de recommandation | Implémentation des algorithmes de matching. | Code source et modèle d’IA |
| Tests et optimisation | Tests unitaires, intégration et validation du système. | Rapport de test |
| Documentation | Création d’une documentation complète. | Documentation finale |

## 7. Technologie prévue

* **Collecte de données :** API REST, scraping avec Python (BeautifulSoup, Scrapy).
* **Base de données :** PostgreSQL ou MongoDB.
* **Back-end :** Python (Django ou Flask).
* **Front-end :** HTML, CSS, JavaScript (React ou Vue.js).
* **Recommandation :** Algorithmes de machine learning (scikit-learn, TensorFlow).

## 8. Critères de réussite

* Centralisation de 90 % des offres pertinentes disponibles sur le marché.
* Taux d’adéquation entre les offres recommandées et les profils utilisateurs évalué à plus de 80 %.
* Temps de réponse du système inférieur à 2 secondes.
* Satisfaction des utilisateurs mesurée à travers des enquêtes (note moyenne > 4/5).

## 9. Risques et plan de mitigation

* **Problèmes techniques :** Plan de secours avec des technologies alternatives.
* **Délais serrés :** Priorisation des tâches critiques.
* **Problèmes de conformité RGPD :** Audit juridique et anonymisation des données sensibles.

Avec ce cahier des charges complet, nous pouvons entamer les premières étapes de l’analyse des besoins et de la conception. Préviens-moi une fois que tu es prête à passer à l’étape suivante !

# ANALYSE DU PROJET

## Description de l’existant

Actuellement, la recherche d’emploi repose majoritairement sur des plateformes traditionnelles telles que LinkedIn, Indeed, et Glassdoor. Chacune de ces plateformes fonctionne de manière isolée, obligeant les utilisateurs à naviguer entre plusieurs interfaces pour consulter des offres variées. Cette fragmentation des données engendre une perte de temps considérable, une expérience utilisateur peu fluide, et des difficultés à identifier des opportunités correspondant précisément aux compétences des candidats. De plus, les systèmes de recommandation existants sont souvent basés sur des algorithmes basiques, limitant la personnalisation des résultats.

## Problématique

Face à la multiplicité des plateformes et à l’hétérogénéité des offres, comment concevoir un système centralisé capable de regrouper efficacement les offres d’emploi et de proposer des recommandations personnalisées, tout en optimisant l’adéquation entre les compétences des candidats et les exigences des recruteurs ?

## Modélisation avec Méthode d’analyse UML

### Diagramme de cas d’utilisation

Identification des acteurs.

|  |  |
| --- | --- |
| Acteurs | Cas d’utilisation |
| Chercheur d’emploi | * Créer un compte / Se connecter * Compléter son profil (compétences, expériences, préférences d’emploi) * Rechercher des offres d’emploi * Recevoir des recommandations personnalisées * Filtrer et trier les offres selon des critères (localisation, type de contrat, etc.) * Postuler à des offres d’emploi * Sauvegarder des offres d’emploi * Suivre l’état de ses candidatures * Recevoir des notifications (nouvelles offres correspondant au profil, réponse à une candidature, etc.) |
| Recruteur | * Créer un compte / Se connecter * Publier des offres d’emploi * Gérer ses offres publiées (modifier, supprimer, archiver) * Consulter les candidatures reçues * Rechercher des profils de candidats * Recevoir des recommandations de candidats pertinents * Contacter des candidats directement * Gérer les réponses aux candidatures |