

# **Projet : Plateforme intelligente de mise en relation**

## **employeurs-employés avec IA**

### **1. Contexte et problématique**

Le marché du recrutement est saturé d'offres et de candidatures, ce qui rend difficile pour les employeurs de trouver rapidement les profils pertinents, et pour les candidats d'accéder à des opportunités adaptées à leurs compétences.

Les méthodes de sélection traditionnelles sont chronophages et parfois subjectives, alors que les entreprises cherchent à digitaliser et optimiser leurs processus RH.

Dans ce contexte, l'intelligence artificielle devient un levier incontournable pour automatiser et personnaliser la mise en relation.

### **2. Objectifs du projet**

- **Automatiser le matching** entre offres d'emploi et profils candidats grâce à l'IA.
- **Permettre la personnalisation** automatique des CVs et lettres de motivation selon chaque offre.
- **Optimiser le temps de recrutement** pour les employeurs via des suggestions ciblées et justifiées.
- **Améliorer l'expérience utilisateur** (employeurs & candidats) grâce à une interface moderne, intuitive et réactive.

### 3. Fonctionnalités principales

#### Pour les candidats :

- Création et gestion de profil
- Téléversement et édition de CVs
- Génération automatique de lettres de motivation personnalisées (IA)
- Suggestions d'offres d'emploi pertinentes

#### Pour les employeurs:

- Création de profil entreprise
- Publication et gestion des offres d'emploi
- Réception de candidatures pré-analysées et justifiées par l'IA

#### Pour la plateforme:

- Matching intelligent offres ↔ profils (NLP/IA)
- Génération automatique de documents (lettres, CV adaptés)
- Tableau de bord et notifications en temps réel
- Gestion sécurisée des données (RGPD, chiffrement)

### 4. Contraintes techniques

- **Frontend:** React.js (web), React Native (mobile)
- **Backend:** Node.js + Express.js (API REST)
- **IA:** OpenAI API, modèles HuggingFace (NLP)
- **Base de données :** PostgreSQL (données structurées), MongoDB (NoSQL)
- **Conteneurisation:** Docker, Kubernetes
- **CI/CD:** GitHub Actions
- **Cloud:** AWS
- **Sécurité :** Chiffrement AES-256, authentification OAuth 2.0/JWT, gestion des accès RBAC

- **Conformité:** RGPD
- 

## 5. Architecture technique (schéma simplifié)

*Les diagrammes UML complets et à jour sont accessibles dans le dépôt GitHub du projet, section UML:*

<https://github.com/elatrous0299/recrutement-ia-platform/tree/main/uml>

La plateforme est basée sur une architecture microservices :

- Un frontend React.js pour le web (et React Native pour mobile)
  - Un backend Node.js/Express découpé en microservices (gestion utilisateurs, matching IA, gestion offres/candidatures, génération documents)
  - Des bases de données PostgreSQL et MongoDB
  - Intégration IA via APIs externes (OpenAI, HuggingFace)
  - Conteneurisation Docker et orchestration Kubernetes
  - CI/CD via GitHub Actions pour les tests et le déploiement continu
  - Hébergement cloud
- 

## 6. Livrables attendus

- Application web et mobile fonctionnelle (MVP puis version avancée)
- Code source complet sur GitHub (avec pipelines CI/CD)
- Cahier des charges détaillé
- Diagrammes UML (cas d'utilisation, classes, séquence, activités, modèle ER)
- Documentation API
- Rapports de tests unitaires et fonctionnels
- Présentation finale

## 9. Glossaire

**Matching IA** : Processus d'appariement automatique via intelligence artificielle

- **NLP** : Natural Language Processing (traitement automatique du langage)
  - **CI/CD** : Intégration et déploiement continus
- 

## 10. Contact

- **Auteur:** Baraa Elatrous
- **Encadrement:** Steve Ataky, PhD
- **Année:** 2024-2025

### Note de auteur:

Ce projet a été réalisé dans le cadre du projet de fin d'études du programme **Techniques de l'informatique** au **Collège Teccart, Montréal**.