

TEST TECHNIQUE – AGENT IA RH (VERSION AMÉLIORÉE)

Candidat : Espérance AYIWAHOUN

Durée : 8 jours (à rendre le 05/12/2025 à 23h59 au plus tard)

Note : Vous pouvez soumettre votre travail avant cette date.

Livrables obligatoires

1. **Dépôt GitHub complet** avec code source et documentation
2. **Version déployée en ligne** (accessible via URL publique)
3. **Rapport technique PDF** (2-6 pages)

1. Contexte et objectifs

Contexte métier

Une entreprise souhaite développer un **Agent IA RH** capable d'automatiser l'analyse de candidatures en :

- Analysant des CV de manière structurée
- Comprenant les exigences des offres d'emploi
- Identifiant précisément les métiers
- Évaluant la correspondance candidat ↔ offre
- Générant des recommandations actionnables pour les recruteurs

Objectifs du test

Ce test évalue votre capacité à :

- Concevoir un agent IA autonome et fonctionnel
- Extraire et structurer des informations textuelles complexes
- Intégrer une logique métier RH pertinente
- Déployer un prototype accessible en production
- Documenter vos choix techniques de manière claire

Vous disposez d'une totale liberté sur votre organisation, méthodes, outils et stack technique.

2. Mission principale

Entrées attendues

Votre agent doit accepter :

- **Un CV** (format : texte brut, PDF, ou DOCX)
- **Une offre d'emploi** (format : texte brut)

Sorties attendues : Rapport structuré

A. Identification du métier candidat

- **Obligatoire** : Classification parmi la liste officielle des métiers (voir section 3)
- Niveau de confiance de la prédiction (optionnel mais recommandé)

B. Extraction des informations clés du CV

Minimum requis :

- Compétences techniques (technologies, langages, frameworks)
- Compétences comportementales (soft skills)
- Expériences professionnelles significatives (postes, durées, entreprises)
- Outils et technologies maîtrisés
- Estimation du niveau de séniorité (Junior / Intermédiaire / Senior / Expert)
- Formations et certifications (optionnel)

C. Extraction des informations clés de l'offre

Minimum requis :

- Compétences requises (techniques et comportementales)
- Niveau d'expérience demandé
- Technologies et outils attendus
- Missions principales du poste
- Métier recherché
- Secteur d'activité (optionnel)

D. Logique de correspondance (matching)

Approches possibles (libre choix) :

- Règles métier simples (booléennes)
- Calculs de similarité textuelle (cosine, Jaccard)
- Embeddings sémantiques (sentence transformers, OpenAI)
- Pondération personnalisée par critère
- Classification supervisée
- Approche hybride combinant plusieurs méthodes

Attendu minimum :

- Analyse des correspondances compétences CV ↔ offre
- Identification des écarts (compétences manquantes)
- Justification logique du score

E. Génération du rapport final

Structure minimale requise :

1. IDENTIFICATION

- Métier identifié
- Niveau de confiance

2. ANALYSE DE CORRESPONDANCE

- Score global (sur 100, notation libre, ou catégorie)
- Compétences correspondantes (avec détails)
- Compétences manquantes ou partielles




3. POINTS FORTS

- Atouts majeurs du candidat pour ce poste
- Expériences particulièrement pertinentes

4. POINTS D'ATTENTION

- Lacunes identifiées
- Écarts d'expérience
- Compétences à développer

5. RECOMMANDATION FINALE

-  Fortement recommandé
-  À considérer (entretien conseillé)
-  Non recommandé
- Justification de la recommandation

F. Mise à disposition

Deux options acceptées :

Option 1 : Interface web

- Formulaire de saisie/upload CV + offre
- Affichage du rapport généré
- Interface intuitive et responsive

Option 2 : API REST

- Endpoint POST pour soumission
- Documentation API (Swagger/OpenAPI recommandé)
- Exemples de requêtes cURL/Postman

G. Déploiement en ligne

Hébergeurs gratuits suggérés :

- Vercel (recommandé pour Next.js, React)

- Render (fullstack, Node.js, Python)
- Railway (Docker, databases)
- HuggingFace Spaces (Gradio, Streamlit)
- Google Cloud Run (tier gratuit)
- Autre plateforme de votre choix

Requis : URL publique fonctionnelle fournie dans le README

3. Liste officielle des métiers (obligatoire)

Votre agent **doit obligatoirement** classer le candidat parmi ces métiers :

A. Tech – Général

Développeur web, Développeur front-end, Développeur back-end, Développeur full-stack, Développeur mobile, Développeur cloud, Data scientist, Data analyst, Data engineer, Machine learning engineer, Deep learning engineer, Software engineer, DevOps engineer, Tech lead

B. Développement – Back-end

Développeur Python, Développeur Django, Développeur Flask, Développeur Laravel, Développeur PHP, Développeur Java, Développeur Node.js, Développeur Ruby on Rails, Développeur Symfony, Développeur Go, Développeur Rust, Développeur C++, Développeur .NET, Développeur Kotlin, Développeur Swift, Développeur Android, Développeur Salesforce, Développeur Blockchain

C. Développement – Front-end

Développeur React, Développeur Next.js, Développeur Vue.js, Développeur Angular, Développeur Svelte, Développeur Flutter, Développeur Webflow, Développeur WordPress, Développeur HTML/CSS

D. Développement – Spécialités

Développeur Web3, Développeur IA, Développeur AR/VR, Développeur jeux vidéo, Développeur no-code, Développeur RPA

E. IA & Data

AI engineer, NLP engineer, Computer vision engineer, Data warehouse developer, Business intelligence analyst, Chief data officer (CDO), Data quality analyst, Big data engineer

F. Design & Audiovisuel

UX designer, UI designer, Product designer, Designer graphique, Motion designer, Directeur artistique, Illustrateur, Animateur 2D/3D, Vidéaste, Monteur vidéo, Coloriste, Ingénieur du son

G. Marketing & Communication digitale

Social media manager, Community manager, Consultant SEO, Rédacteur SEO, Content manager, Copywriter, Email marketing specialist, PPC expert, CRM manager, Meta Ads manager, Google Ads manager, Growth hacker, Affiliate marketing manager, Product marketer

4. Livrables détaillés

A. Dépôt GitHub

Structure recommandée :

```
/
├── README.md (complet et clair)
├── src/ (code source)
├── data/ (exemples de CV et offres)
├── docs/ (documentation technique)
├── tests/ (tests unitaires - bonus)
├── requirements.txt / package.json
└── .env.example (si applicable)
```

README.md doit contenir :

- Description du projet
- Technologies utilisées
- Instructions d'installation (étape par étape)
- Instructions d'utilisation
- Lien vers la version déployée
- Exemples d'utilisation
- Structure du projet
- Auteur et contact

Données d'exemple obligatoires :

- Minimum 3 CV exemples (formats variés)
- Minimum 3 offres d'emploi exemples
- Sources des données mentionnées

B. Version déployée en ligne

Fonctionnalités minimales :

- ✓ Accessible via URL publique
- ✓ Formulaire ou API fonctionnel(le)
- ✓ Traitement CV + offre
- ✓ Affichage du rapport complet
- ✓ Temps de réponse raisonnable (<30s recommandé)
- ✓ Gestion basique des erreurs (messages clairs)

Bonus apprécié :

- Interface utilisateur soignée
- Historique des analyses
- Export du rapport (PDF, JSON)
- Indicateurs de chargement

C. Rapport technique PDF

Structure suggérée (2-6 pages) :

1. Introduction (0.5 page)

- Résumé de l'approche choisie
- Objectifs atteints

2. Architecture technique (1-2 pages)

- Stack technologique complète
- Architecture globale (schéma recommandé)
- Choix des modèles IA / librairies
- Pipeline de traitement (étapes)

3. Logique métier (1-2 pages)

- Extraction des informations (NER, parsing)
- Méthode de classification des métiers
- Algorithme de matching (détaillé)
- Calcul du score et recommandation

4. Données et tests (0.5 page)

- Sources des données utilisées
- Approche de validation
- Résultats sur exemples

5. Limites et perspectives (0.5-1 page)

- Limites techniques actuelles
- Cas d'usage non couverts
- Améliorations futures possibles
- Scalabilité

Formats acceptés : PDF uniquement

5. Critères d'évaluation

Critère	Poids	Détails
Pertinence de l'analyse CV & offre	20%	Qualité de l'extraction, exhaustivité, précision
Qualité de l'identification du métier	20%	Précision, cohérence, justification
Logique de correspondance & scoring	20%	Pertinence métier, algorithme, transparence
Qualité du rapport final	15%	Clarté, structure, actionnabilité
Qualité et lisibilité du code	15%	Organisation, documentation, bonnes pratiques
Déploiement fonctionnel	10%	Accessibilité, stabilité, UX

Bonus possibles (+5% chacun, max 15%) :

- Tests unitaires / intégration
- Documentation API complète (Swagger)
- Interface utilisateur exceptionnelle
- Gestion avancée des erreurs
- Fonctionnalités additionnelles pertinentes



6. Notes importantes

Libertés accordées

✓ Vous êtes totalement libre sur :

- Organisation du code et architecture
- Approches et algorithmes IA
- Choix des modèles (OpenAI, HuggingFace, Claude, Gemini, open-source...)
- Stack technique (Python, Node.js, Java, etc.)
- Frameworks et librairies
- Sources de données (CV publics sur GitHub, LinkedIn, Kaggle, ou créés par vous)

Périmètre du test

Vous devez :

- Créer un mini-agent simple mais cohérent
- Démontrer une logique IA pertinente
- Assurer un déploiement stable et accessible
- Produire une documentation claire

FAQ

Q : Puis-je utiliser des API payantes (OpenAI, Claude) ?

R : Oui, mais privilégiez les tiers gratuits ou des alternatives open-source pour la version déployée.

Q : Dois-je gérer l'authentification ?

R : Non, ce n'est pas requis pour ce test.

Q : Combien de CV/offres dois-je tester ?

R : Minimum 3 paires CV-offre dans chaque métier, mais testez-en davantage pour valider votre logique.

Q : Puis-je utiliser des données en anglais ?

R : Oui, mais précisez-le clairement dans votre README. Le français est recommandé.

Q : Que faire si je ne termine pas tout ?

R : Priorisez : identification métier + matching basique + déploiement > fonctionnalités avancées.