



POLYTECH
MONTPELLIER

GenAi Labs

Automatisation de Recherche de Stage

Réalisé par
GOETZ Hugo
D03

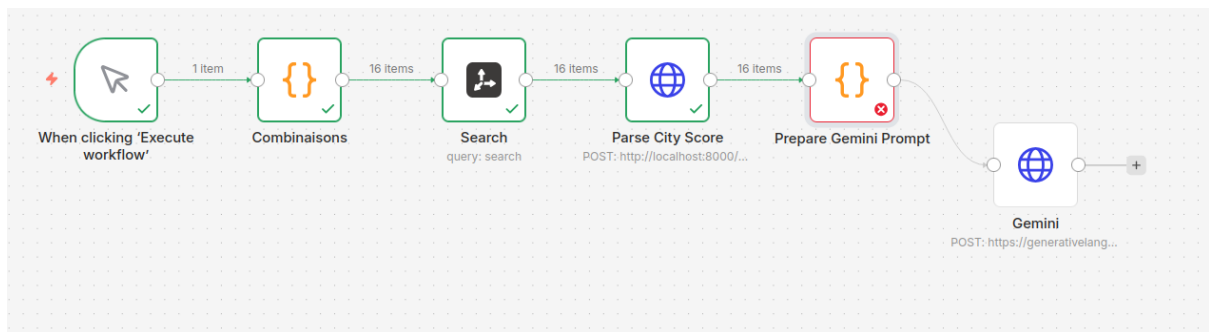
1. Introduction et Objectifs

La recherche de stage à l'international est un processus chronophage qui demande de croiser des offres d'emploi avec des données géographiques (coût de la vie, sécurité). Mon objectif était d'automatiser ce flux pour :

1. Générer des recherches ciblées (Ville × Technologie).
2. Récupérer des offres via un moteur de recherche IA (Tavily).
3. Scorer objectivement chaque ville via une API personnalisée.
4. Extraire les détails clés via un LLM (Gemini).

Le projet a abouti à 70% des objectifs. L'infrastructure (Recherche + Scoring API) est pleinement fonctionnelle, mais l'extraction finale par IA n'a pas pu être finalisée pour des raisons techniques détaillées dans ce rapport.

2. Architecture du Workflow (n8n)



Le flux de données :

1. **Code (JS)** : Génération automatique de 20 combinaisons de recherche (ex: "internship AI in Berlin").
2. **Tavily Search** : Interrogation du web pour récupérer des URLs et des snippets d'offres (16 résultats pertinents obtenus).
3. **HTTP Request (API Flask)** : Envoi de la ville à mon API locale pour obtenir un score de recommandation.
4. **Code (Merge)** : Tentative de fusion des résultats de recherche avec le score de la ville.

5. **Gemini (LLM)** : Étape finale (non aboutie) pour structurer le JSON.

3. Méthode de Scoring

J'ai implémenté un système de notation sur mesure pour évaluer l'attractivité des villes.

L'Algorithme

J'ai utilisé une pondération précise des indices Numbeo :

$$\text{score} = (\text{qualityOfLife} \times 0.25 + \text{safety} \times 0.20 + \text{purchasingPower} \times 0.15 + (100 - \text{costOfLiving}) \times 0.15 + \text{healthCare} \times 0.10 + (100 - \text{pollution}) \times 0.10 + \text{climate} \times 0.05)$$

Justification de la Pondération

Profil cible : Stagiaire 22-25 ans, budget limité (500-1500€/mois), première expérience internationale.

Hiérarchie des priorités :

- 45% - Sécurité & Bien-être (Quality 25% + Safety 20%) : Critères prioritaires pour un départ serein.
- 30% - Finances (Purchasing 15% + Cost 15%) : Le coût de la vie et le pouvoir d'achat sont cruciaux pour un salaire de stagiaire.
- 20% - Santé & Environnement (Health 10% + Pollution 10%) : Pour le confort au quotidien.
- 5% - Confort personnel (Climate 5%) : Bonus subjectif.

Est-ce que cela fonctionne ?

Oui. La méthode a correctement identifié Amsterdam (Score 127.5) comme "Excellent choix" grâce à sa haute qualité de vie, malgré un coût élevé. À l'inverse, Paris (Score 98.4) est pénalisée par son indice de sécurité plus faible et son coût élevé, tombant dans la catégorie "Correct". Cela correspond à la réalité du terrain pour un étudiant.

4. Analyse Critique

Difficultés Rencontrées

Le projet s'est avéré techniquement beaucoup plus ardu que prévu. J'ai rencontré plusieurs blocages majeurs qui ont ralenti l'avancée globale et m'ont empêché de finaliser l'outil :

1. **Configuration du MCP** : J'ai eu beaucoup de mal à comprendre le fonctionnement intrinsèque du protocole au début. La création du serveur et surtout sa connexion au MCP Inspector ont été très problématiques. Je n'arrivais pas à établir la connexion pour déboguer, et je n'ai réussi à faire fonctionner cette brique essentielle que lors de la toute dernière semaine, ce qui a considérablement réduit mon temps de développement pour le reste.
2. **Récupération des données Tavily** : Une fois la recherche effectuée par l'IA, j'ai eu de grosses difficultés à récupérer les données (liens, snippets) pour les transmettre proprement aux nœuds suivants dans n8n. Les références aux données ne fonctionnaient pas comme prévu, et je perdais souvent le contexte ou le contenu de la recherche entre les étapes. C'était un problème de flux de données que je n'ai pas réussi à stabiliser totalement.
3. **Intégration Gemini** : C'est l'échec principal. En plus des erreurs de syntaxe JSON persistantes qui bloquaient l'envoi du prompt. Je n'ai jamais réussi à obtenir une extraction fiable.

Conséquence : Pas de filtrage

À cause du temps considérable passé à essayer de résoudre ces problèmes de connexion MCP et de récupération de données Tavily, je n'ai pas eu le temps d'implémenter le système de filtrage que j'avais prévu. Initialement, je voulais trier les résultats par date ou salaire.

Le workflow a-t-il été utile ?

Le bilan est donc mitigé :

- **Oui pour le Scoring** : L'évaluation des villes fonctionne parfaitement et offre une aide à la décision objective immédiate.
- **Non pour l'automatisation complète** : L'objectif n'est pas atteint car je dois encore traiter les résultats de recherche un par un, l'extraction automatique et le filtrage ayant échoué.

5. Conclusion

Bilan du Projet

Pour être tout à fait honnête, je ne considère pas ce projet comme une réussite complète. Je n'ai pas réussi à aboutir à l'automatisation totale que je visais au départ.

Ce qui n'a pas marché :

- L'extraction intelligente des données par le LLM est un échec.
- Je n'ai pas pu filtrer les résultats comme je le souhaitais faute de temps.
- Le système final demande encore trop d'interventions manuelles pour être vraiment efficace.