

Rapport de Projet GenAI

Automatisation de recherche de stage à l'étranger

Gayrard Matéo

31 Janvier 2026

Table des matières

1	Introduction	2
2	Partie 1 : Serveur MCP de Scoring de Ville	2
2.1	Choix de la méthode de scoring	2
2.2	Implémentation Technique	2
3	Partie 2 : Workflow d'automatisation n8n	2
3.1	Vue d'ensemble	2
3.2	Étapes clés du processus	3
4	Partie 3 : Résultats et Analyse	3
4.1	Exemple de résultats extraits	3
4.2	Fichiers livrés	4
5	Analyse Critique	4
5.1	Choix des critères de filtrage	4
5.2	Difficultés rencontrées et Limites	4
5.3	Bilan	4

1 Introduction

Ce projet s'inscrit dans le cadre du cours GenAI de Polytech (DO3). L'objectif est de concevoir un système automatisé combinant un serveur MCP (Model Context Protocol), un workflow d'automatisation (n8n) et des LLM pour rechercher, filtrer et analyser des offres de stage à l'étranger.

2 Partie 1 : Serveur MCP de Scoring de Ville

2.1 Choix de la méthode de scoring

Pour évaluer l'attractivité des villes, j'ai choisi de baser mon scoring sur des données issues de *Nomad List 2024*.

J'ai constitué un dataset JSON ('city_data.json') contenant pour chaque ville :

- Un score global (sur 10).
- Le coût de la vie.
- La qualité d'internet (Mbps).
- Le niveau de sécurité.

2.2 Implémentation Technique

Le serveur MCP a été développé en Python en utilisant la librairie `fastmcp`. Le script de lancement ('run_mcp.sh') initialise l'environnement virtuel et expose le serveur sur le port 8000.

```
1 #!/bin/bash
2 # Script pour lancer le serveur MCP
3 source .venv/bin/activate
4 fastmcp run mcp_server.py:mcp --transport http --path /mcp --port 8000
   --host 0.0.0.0
```

Listing 1 – Script de lancement du serveur MCP

3 Partie 2 : Workflow d'automatisation n8n

3.1 Vue d'ensemble

Le workflow a été conçu pour orchestrer les différentes étapes de la recherche. Il combine des nœuds de code JavaScript, des appels API (Tavily), l'intégration du serveur MCP local et le traitement par LLM (OpenAI/Gemini).

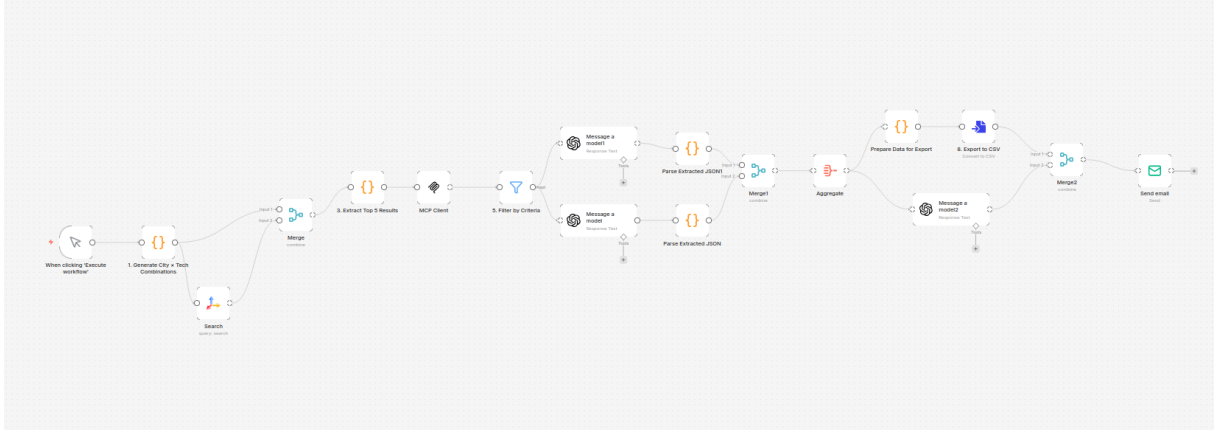


FIGURE 1 – Vue globale du workflow n8n (Fichier : Gayrard-workflow.json)

3.2 Étapes clés du processus

1. **Génération des combinaisons (Code Node)** : Création dynamique de paires Ville × Techno (ex : Berlin + DevOps, Tokyo + AI).
2. **Recherche Web (Tavily)** : Récupération des URL d’offres de stage récentes.
3. **Enrichissement (MCP)** : Appel au serveur local pour ajouter le score de la ville à chaque offre.
4. **Extraction et Résumé (LLM)** : Utilisation d’un modèle de langage pour structurer les données brutes (HTML/Texte) en format JSON (Entreprise, Salaire, Deadline) et générer un résumé court.
5. **Filtrage et Export** : Agrégation des données et export vers un fichier CSV.
6. **Synthèse et Notification** : Envoi d’un email récapitulatif contenant les meilleures offres.

4 Partie 3 : Résultats et Analyse

4.1 Exemple de résultats extraits

Voici un extrait des offres pertinentes trouvées par le workflow et exportées dans le fichier `exemple.csv`.

Entreprise	Poste	Ville (Score)	Résumé LLM
CIEE	Global Internship	Berlin (8.5)	Stage international couvrant l’IA et le dév logiciel. Expérience immersive à Berlin.
Superhuman	Software Engineer Intern	Berlin (8.5)	Stage hybride été 2026. Stack : Java, Python, Go. 2 jours sur site/semaine.
HENNGE	Global Internship	Tokyo (9.2)	Stage 4-6 semaines. Dév Python/Go sur AWS. Subventionné mais non rémunéré.

TABLE 1 – Extrait du fichier de sortie `exemple.csv`

4.2 Fichiers livrés

Le repository GitHub contient :

- `Gayrard-workflow.json` : Le code source du workflow.
- `city_data.json` : La base de données de scoring.
- `exemple.csv` : L'export complet des résultats.
- `run_mcp.sh` : Le script d'exécution.

5 Analyse Critique

5.1 Choix des critères de filtrage

Pour ne retenir que les offres les plus pertinentes, j'ai implémenté un nœud de filtrage ("Filter Node") avec des critères "classiques" mais stricts, adaptés à ma situation :

- **Score global > 8/10** : Pour s'assurer d'une excellente qualité de vie sur place.
- **Coût de la vie < 2500 / mois** : En tant qu'étudiant, la contrainte budgétaire est prioritaire. J'ai donc exclu les villes excessivement chères (comme San Francisco ou Zurich) si elles dépassaient ce seuil.
- **Sécurité > 75/100** : Un critère indispensable pour envisager un départ à l'étranger en toute sérénité.

5.2 Difficultés rencontrées et Limites

Le développement de ce workflow a mis en lumière plusieurs défis techniques et structurels :

1. **Prise en main de Tavily** : Le manque d'expérience préalable avec l'API de recherche Tavily a rendu l'étape de recherche initiale laborieuse. Il a été difficile d'affiner les "queries" pour obtenir des résultats très récents et éviter le bruit (vieux stages, articles de blog). Une meilleure maîtrise des opérateurs de recherche avancés aurait permis d'optimiser cette étape plus rapidement.
2. **Architecture Monolithique vs Agents** : Une des limites majeures de ce projet est la longueur du workflow n8n. Nous effectuons beaucoup de tâches de natures différentes (Recherche, Scoring,... dans un seul flux linéaire. L'utilisation d'Agents IA spécialisés aurait été plus judicieuse. Par exemple, un agent "Recruteur" pour la recherche et un agent "Analyste" pour le scoring auraient permis de regrouper les tâches et de simplifier le workflow

5.3 Bilan

Malgré ces difficultés, le workflow est fonctionnel et permet d'économiser des heures de recherche manuelle. Il transforme une tâche fastidieuse en un processus décisionnel basé sur la donnée (Data-Driven).

J'ai donc réussi à m'envoyer un mail de confirmation contenant un résumé global des offres ainsi que le fichier csv en pièce-jointe.