
N8n Job Recommendation Automation

Status: Completed
Timing: Aug 14, 2025 – Aug 20, 2025
Owners: Tejeswarreddy Mutchu

Overview

This project automates the process of fetching job postings, scoring them, generating personalized cover letters, and notifying users with top job recommendations via email.
It integrates n8n, Google Gemini API, Google Sheets, and Email services into one seamless workflow.

Problem Explanation

The traditional job search process is repetitive, time-consuming, and inefficient:

- Candidates browse multiple job portals manually.
- Relevant jobs are shortlisted by hand.
- Personalized cover letters are written individually for each application.

Objectives

Project objectives
The primary objective of this project is to automate and optimize the job search process using AI and workflow automation tools.
The system aims to reduce manual effort, improve job matching accuracy, and deliver a personalized application experience for candidates.

- Automate job data collection and extraction
- AI-powered job relevance scoring
- Personalized cover letter generation
- Centralized job tracking in Google Sheets
- Automated top job recommendations via email

Key Success Metrics

- Job matching accuracy (≥ 80%)
- Automation coverage (≥ 90%)
- Email engagement rate (≥ 70%)
- Time saved (≥ 60% effort reduction)

- System scalability (≥ 50 job listings per run)

Strategy & Approach

- Requirement Analysis – Define workflow scope (job fetch, scoring, cover letter, notifications).
- Workflow Design – Build modular n8n pipeline with Google API and Gemini models.
- Model Integration – Use Gemini 2.5 Flash for scoring & extraction, Gemini 2.0 Flash-Lite for cover letters.
- Data Management – Store structured results in Google Sheets for centralized tracking.
- Notification System – Automate email delivery with top job recommendations.
- Testing & Iteration – Validate workflow with sample jobs, refine prompts & parsing logic.
- Team Collaboration – Assign ownership for APIs, prompts, and workflow nodes to ensure timely delivery.

---

n8n Workflow Breakdown (Node-by-Node)

1. **Schedule Trigger** – Triggers the workflow periodically (daily).
2. **RSS Read** – Pulls job postings from RSS feeds (LinkedIn Jobs, Google Jobs, etc.).
3. **Code Node** – Flattens nested JSON structures; removes unwanted HTML tags and extracts clean text.
4. **Gemini 2.5 Flash (Job Extraction)** – Extracts structured job info (Title, Company, Description, Location, Qualifications).
5. **Code2 Node** – Parses Gemini output and ensures valid JSON format.
6. **Gemini 2.5 Flash (Job Scoring)** – Evaluates job relevance based on candidate profile; returns { "score": X }.
7. **Code3 Node** – Extracts only the score value.
8. **Gemini 2.0 Flash-Lite (Cover Letter Generation)** – Generates personalized cover letters (higher RPM/TPM limits).
9. **Code4 Node** – Cleans and flattens the cover letter output.
10. **Google Sheets (Append/Update Row)** – Stores structured job data (Title, Company, Score, Link, Benefits, Cover Letter, Location).
11. **Sort Jobs by Score (Code Node)** – Sorts all jobs in descending order.
12. **Format Email Body (Code Node)** – Creates structured email template (Title, Company, Score, Link, Benefits, Location).
13. **Gmail API (Send Email)** – Sends an email with Top 5 job recommendations.

Resources

- Daily email report
- Google Sheet

API & Prompt Usage

APIs Used:

- Google Gemini API
    - Gemini 2.5 Flash: Job detail extraction & scoring.
    - Gemini 2.0 Flash-Lite: Cover letter generation.
- Google Sheets API: Append/update job data.
- Gmail API: Send email notifications.

Prompts Used:

A] Model for Extract Information

- System Prompt: "You are an intelligent bot capable of pulling out data from a job listing site."
- User Prompt: here's the job site: {{ $json.link }}
- Assistant Prompt: Output JSON with company_name, benefits, job_description, location.

B] Model for Score

- System Prompt: "You are an intelligent bot rating how closely a job listing is to a candidate skill set, on a 5 score."
    - 3 points for skills matching, 1 point for mostly matching
    - 1 point for correct experience
    - 1 point for remote position
    - 1 point if current job role matches past role
    - 1 point if resume skills match job
    - 3 points if candidate meets job qualifications
- User Prompt: Job details + sample resume
- Assistant Prompt: Output JSON { "score": "" }

C] Model for Cover Letter

- System Prompt: "You are perfect at creating a cover letter for a job. Please take the candidate's resume and create a customized cover letter."
- User Prompt: Job details (Title, Description, Company).
- Assistant Prompt: Output JSON { "cover_letter": "" }

Challenges & Solutions

1. **API Rate Limits** – Solved with Gemini 2.0 Flash-Lite and batching jobs (groups of 4–5).
2. **Text Parsing Issues** – Solved with regex and JSON.parse() in Code nodes.
3. **Email Formatting** – Solved by creating custom templates for readability.
4. **Selecting Top-N Jobs** – Sorted jobs and sliced to Top 5.

Summary of Learnings

- Integrated AI-powered workflows with automation tools.
- Applied prompt engineering for structured outputs.
- Explored Gemini API trade-offs (Flash vs Flash-Lite).
- Used Google APIs for Sheets and Gmail integration.
- Designed scalable workflows handling errors, limits, and formatting.
- Realized potential of AI + automation in job search.

Conclusion

This project shows how AI + workflow automation can transform the job search process.
By combining Gemini models, n8n, and Google APIs, we built a pipeline that fetches job data, scores it, generates cover letters, and emails top opportunities.
It reduces manual effort while improving recommendation quality.
Future improvements: resume parsing, multi-source aggregation, and advanced personalization.
