**Automated Job Search & Cover Letter Workflow with n8n + Gemini**

**1. Problem Explanation**

Job search and application management can be highly repetitive and time-consuming. Applying to multiple targeted positions daily requires:

* Manually searching for relevant job openings
* Reading and comparing job descriptions to personal skills
* Customizing cover letters for each position
* Tracking applications and communication

This project automates the above workflow using n8n, Google Gemini, and various job APIs. The solution ensures consistent, timely, and high-quality applications with minimal manual effort.

**2. n8n Workflow Breakdown (Node-by-Node)**

Below is a node-by-node explanation of the n8n workflow logic as reflected in your supplied .json:

**2.1 Schedule Trigger**

* Node: Schedule Trigger
* Function: Initiates the workflow daily at a fixed hour (e.g., 8:00 AM IST).
* Purpose: Ensures the job search and application process runs consistently each day.

**2.2 Fetch Job Listings**

* Node: RSS Read
* Function: Reads from a specified RSS feed URL that aggregates job postings (e.g. LinkedIn).
* Purpose: Captures new job listings to process downstream.

**2.3 Limit Results**

* Node: Limit
* Function: Restricts the number of jobs processed in each run for efficiency and API safety.
* Purpose: Prevents API overload and maintains focus on top job listings.

**2.4 Job Detail Fetch**

* Node: HTTP Request
* Function: Retrieves full job details by directly querying the job listing page.
* Purpose: Collects comprehensive description and metadata for the match scoring and cover letter generator.

**2.5 Job Data Extraction via Gemini**

* Node: Message a model
* Function: Uses Gemini to extract structured job details (company, job role, benefits, location, etc.) from raw job page data.
* Purpose: Cleans and normalizes job data for further automation.

**2.6 JSON Clean-Up**

* Node: Code
* Function: Validates and cleans the JSON output from Gemini to ensure consistent downstream processing.
* Purpose: Prevents errors due to malformed or partial outputs.

**2.7 Match Scoring via Gemini**

* Node: Message a model1
* Function: Leverages Gemini again to compare your resume/profile and the job’s requirements, outputting a numeric score reflecting compatibility.
* Purpose: Quantifies the fit, allowing you to prioritize top matches.

**2.8 Parse & Store Score**

* Node: Code1
* Function: Ensures the match score is parsed, cleaned, and formatted as needed.
* Purpose: Makes the data immediately usable for filtering and tracking.

**2.9 Generate Cover Letter (Gemini)**

* Node: Message a model2
* Function: Creates a custom, concise cover letter for the job based on both the cleaned job data and your resume.
* Purpose: Saves time and increases the quality of every application.

**2.10 Parse Cover Letter Output**

* Node: Code2
* Function: Extracts and cleans the cover letter text for storage and emailing.
* Purpose: Ensures proper formatting and storage consistency.

**2.11 Store in Google Sheets**

* Node: Append or update row in sheet
* Function: Updates a Google Sheet with all key job listing details, match scores, and generated cover letters.
* Purpose: Centralizes job tracking and application history.

**2.12 Email Notification**

* Node: Send a message
* Function: Emails a summary (e.g., top matches) or the generated cover letter using Gmail.
* Purpose: Keeps you immediately informed and ready to act on daily job opportunities.

**3. API & Prompt Usage Explanation**

**Job Source API (RSS, SerpAPI, JSearch/RapidAPI)**

* *Role:* Fetches current job openings based on targeted queries and filters (e.g., position, location).
* *Normalization:* RSS and search APIs are parsed for consistency in fields (title, company, etc.).

**Google Gemini (via LangChain n8n custom node)**

* *For Job Details Extraction:* Structured prompt tells Gemini to output only critical job attributes as JSON.
* *For Match Scoring:* Custom prompt compares job description against a static resume/profile and outputs a single numeric score.
* *For Cover Letter Generation:* Well-crafted prompt asks Gemini to create a tailored cover letter in strict JSON format, referencing specific skills and experience.

**Google Sheets**

* *Integration:* The n8n Google Sheets node appends new or updates existing rows, mapping all custom fields (job URL, score, company, benefits, etc.).
* *Matching/Update Strategy:* Uses unique fields (e.g., job URL/link) to prevent duplicate entries and ensure up-to-date tracking.

**Gmail/Email**

* *Output:* Sends summary of matches (or individual cover letters) to a chosen email—configurable for frequency and format.

**4. Challenges & Solutions**

| **Challenge** | **Solution** |
| --- | --- |
| API Rate Limits | Added Limit node to restrict number of API calls per run. Staggered automation timing. |
| Job Data Parsing/Extraction Errors | Multiple validation and clean-up (Code) nodes ensure consistent, parseable JSON output. |
| Gemini Output Variance | Strict prompts requesting output strictly in JSON with field-by-field instructions. |
| Skill Matching Accuracy | Embedded static resume within matching prompt, using explicit comparison rubric. |
| Monitoring Duplicates | Used job URL as unique ID in Google Sheets appending/updating logic. |
| Cover Letter Personalization | Integrated resume and job attributes into cover letter prompts for higher relevance. |
| Email Deliverability | Handles email via authenticated Gmail node in n8n, avoiding common spam traps. |

**5. Summary of Learnings**

* Automating end-to-end job search/application using n8n and LLMs can save significant time and greatly improve the efficiency and quality of job applications.
* Thorough prompt engineering with Gemini/LLMs is essential—prompts must be clear on output format and reference relevant user experience.
* Error handling and result cleaning in workflows are key when dealing with dynamic/inconsistent API and LLM responses.
* Data pipeline normalization (through clean-up nodes and careful field mapping) ensures smooth integration with external tools such as Google Sheets and email.
* API rate limits and duplication control are practical engineering requirements that must be considered in production-ready workflows.
* Personalization at scale (dynamic cover letters, auto-match scoring) is achievable with careful template and prompt design, providing a competitive edge in high-volume job searches.