

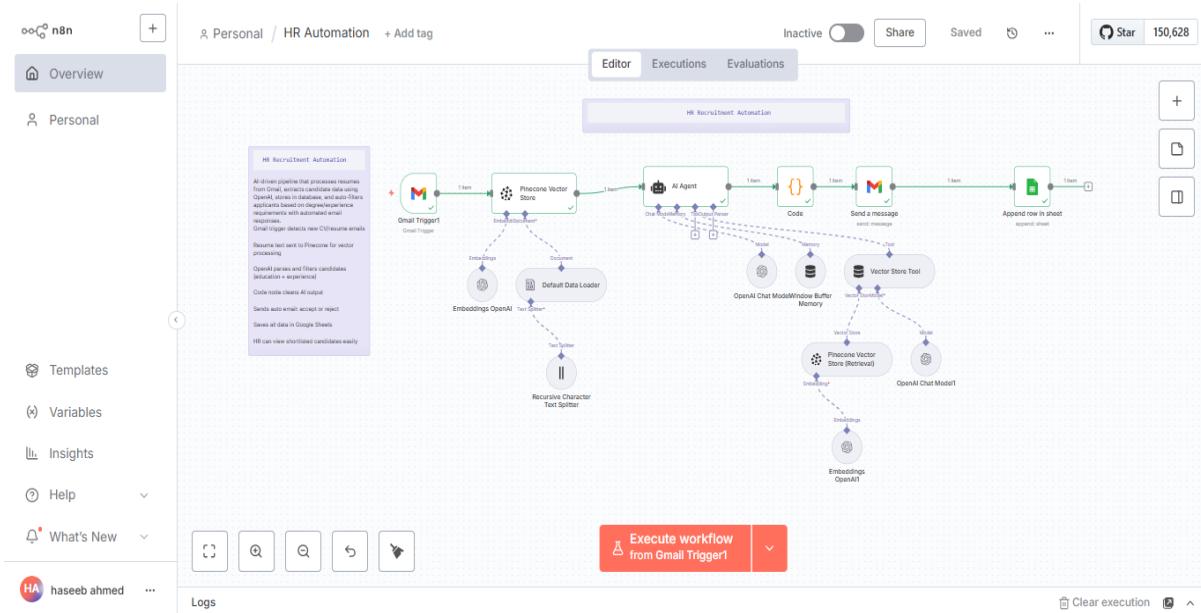
HR Recruitment Automation using n8n

Project Overview

This n8n automation project streamlines the HR recruitment process by automatically collecting, analysing, and filtering incoming resumes. The workflow eliminates manual screening, ensures consistent evaluation based on defined criteria, and maintains organized candidate records for easy HR review.

HR Recruitment Automation using n8n

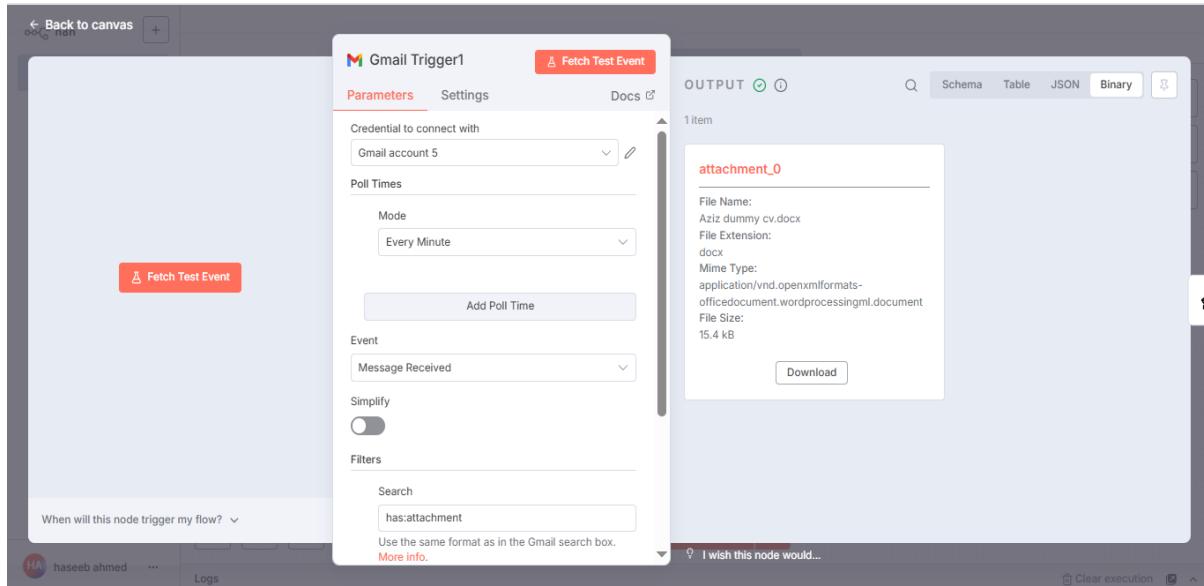
Summary Page



Gmail Node Monitoring

Automatically monitors the HR inbox for new emails containing attachments with the words “CV” or “Resume” in the subject or body.

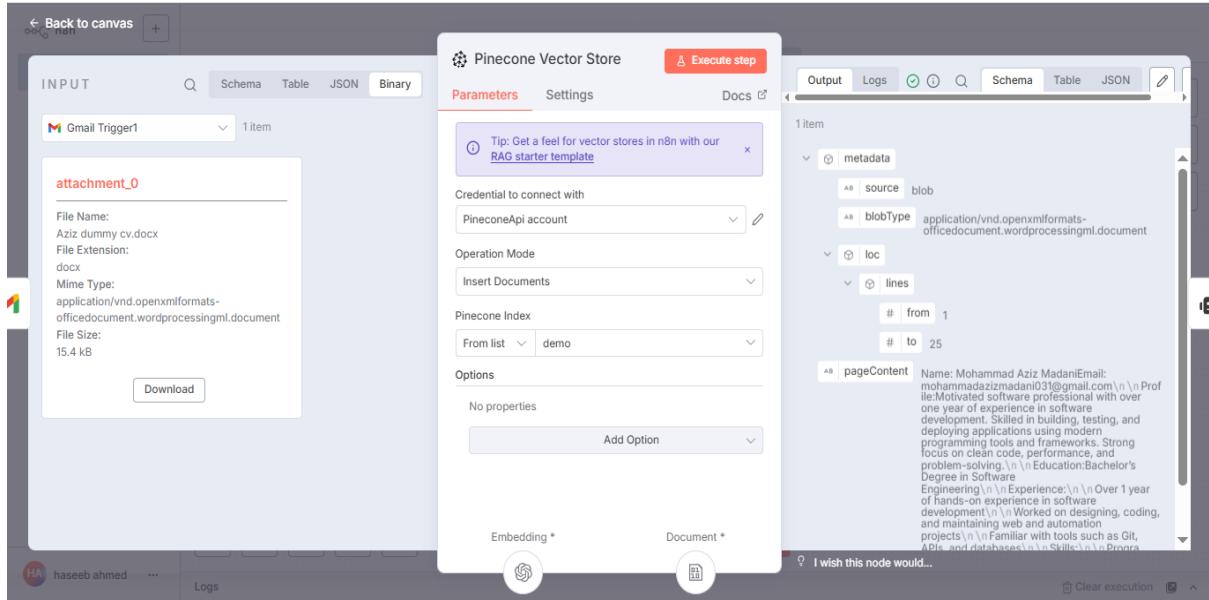
Purpose: To identify and capture all incoming job applications without manual checking.



Resume Processing

When a qualifying email arrives, its attachment (resume) is sent to **Pinecone Vector Store** for text splitting and vectorization.

Purpose: To convert the resume text into a structured format that can be processed by AI for data extraction.



Qualification Filtering

The AI agent reviews each resume to verify qualifications:

- **Education:** Bachelors in Software Engineering or any Finance degree
 - **Experience:** At least 1 year

Results:

- Application sent for further processing — qualified
 - Rejected — not qualified

Purpose: Automatically filters candidates based on set criteria.

The screenshot shows a user interface for an AI workflow. On the left, there is an "INPUT" panel displaying a Pinecone Vector Store with one item. This item contains metadata (blob and blobType), loc (lines from 1 to 25), and pageContent (a resume text). In the center, there is an "AI Agent" step configuration panel. The "Agent" dropdown is set to "Tools Agent". The "Source for Prompt (User Message)" field contains the instruction: "You are a strict resume parsing API. Extract information from the resume text and output ONLY valid JSON." Below this, under "Prompt (User Message)", is a code block with the heading "# EXTRATION RULES:" and a note: "You are a strict resume parsing API. Extract information fr...". A "Require Specific Output Format" toggle switch is turned on. At the bottom of this panel, there is a note: "Connect an output parser on the canvas to specify the output format you require". On the right, there is an "Output" panel showing the resulting JSON output: { "full_name": "Mohammad Aziz Madan", "email": "mohammadazizmadan03@gmail.com", "skills": "Python, JavaScript, C++, C, VS Code, nBn, Postman, MySQL, SQLite, automation, web development, data handling", "education_field": "Software Engineering", "years_of_experience": 1, "experience_summary": "Designed, coded, and maintained web and automation projects using tools such as Git, APIs, and databases. Strong focus on clean code, performance, and problem-solving.", "status": "Application sent for further processing" }. There are also tabs for "Logs" and "Clear execution" at the bottom.

Automated Email Response

If the candidate's resume meets company requirements, an automated email is sent to inform them that their application has been accepted and they will be contacted for an interview.

If not, a polite rejection email is sent thanking them for applying.

Purpose: To provide professional and timely communication with all candidates.

The screenshot shows a workflow editor interface with three main sections: INPUT, PROCESS, and OUTPUT.

INPUT: Contains a 'Code' node with the following fields:

- Full_Name: Mohammad Aziz Madani
- Email: mohammadazizmadani031@gmail.com
- Skills: Python, JavaScript, C++, Git, VS Code, n8n, Postman, MySQL, SQLite, automation, web development, data handling
- Education: Software Engineering
- Experience: 1
- Projects: Designed, coded, and maintained web and automation projects using tools such as Git, APIs, and databases. Strong focus on clean code, performance, and problem-solving.
- Status: Application sent for further processing

PROCESS: A central step titled 'Send a message' with the following configuration:

- Credential to connect with: Gmail account 5
- Resource: Message
- Operation: Send
- To: {{ \$json.Email }} (mohammadazizmadani031@gmail.com)
- Subject: Resume Application Status
- Email Type: HTML
- Message: {{ \$json.Status === "Application sent for further processing" ? "Your application has been received and is currently being reviewed. We will be in touch shortly." : "We regret to inform you that your application did not meet our current hiring needs at this time. Thank you for your interest in our company." }}

OUTPUT: Shows the resulting message details:

- 1 item
- id: 19a06fee6485bdc8
- threadId: 19a06fee6485bdc8
- labelIds:
 - labelId[0]: SENT

Data Logging

All candidate details, including extracted information and qualification status, are appended to a **Google Sheet** for HR reference.

Purpose: To maintain a centralized and organized database of all applicants for easy tracking and follow-up.

The screenshot shows a data integration tool's interface with two main sections: INPUT and OUTPUT.

INPUT: Contains a single node: "Send a message". It has fields for "id" (19a06fee6485bcd8) and "threadId" (19a06fee6485bcd8). A nested node "labelIds" is expanded, showing "labelIds[0]" with the value "SENT". Other collapsed nodes include "Code", "AI Agent", "Pinecone Vector Store", and "Gmail Trigger1".

Step Configuration: The central panel is titled "Append row in sheet". It includes tabs for "Parameters" (selected), "Settings", and "Docs". Under "Parameters", the "Credential to connect with" is set to "Google Sheets account 2". The "Resource" is "Sheet Within Document" and the "Operation" is "Append Row". The "Document" is "From list" (HR_Projects) and the "Sheet" is "From list" (Sheet1). The "Mapping Column Mode" is "Map Each Column Manually". The "Values to Send" field contains the expression: `Full_Name` `fx {{ ${('Code').item.json.Full_Name} }}`.

OUTPUT: Displays the appended data in a table. The columns are "Full_Name", "Email", "Education", "Skills", "Experience", "Projects", and "Status". The data rows are:

Full_Name	Email	Education	Skills	Experience	Projects	Status
Mohammad Aziz Madani	mohammadsizmedani03@gmail.com	Software Engineering	Python, JavaScript, C++, Git, VS Code, n8n, Postman, MySQL, SQLite, automation, web development, data handling	1	Designed, coded, and maintained web and automation projects using tools such as Git, APIs, and databases. Strong focus on clean code, performance, and problem-solving.	Application sent for further processing

Outcome Of HR Automation Project

- Automates the complete recruitment process from email detection to candidate evaluation and response
- Saves HR time by eliminating manual screening
- Reduces human errors in resume evaluation
- Improves overall efficiency and consistency in HR operations