

Department of Computer Science, DESJONQUERES Nicolas

Department of Computer Science, ROBIN Héloïse

Department of Computer Science, KHEYAR Adel

Department of Computer Science, ARM Jules



## Motivation

### Fragmented Job Search Process

Job seekers manually browse **multiple** platforms (LinkedIn, Indeed, emails) and **lose track** of opportunities. No unified view of relevant positions.

### Time-Consuming Application Preparation

Each application requires reading long descriptions, extracting key requirements, and writing personalized cover letters. **Hours spent** on repetitive tasks.

### Generic Automation Tools

Existing solutions like Zapier offer basic workflows but **lack job-specific** intelligence. No semantic understanding of job requirements or candidate profiles.

AI-orchestrated job search, from chaos to clarity,

EVIDI



## Project Highlights

01

### Unified Multi-Source Aggregation

Automated collection from RSS feeds and **general or customized APIs** (LinkedIn, Indeed) into a single dashboard.

02

### AI-Powered Matching & Summarization

Gemini 2.5 extracts skills from CVs, scores job relevance, and generates concise summaries. **No manual reading** of long descriptions.

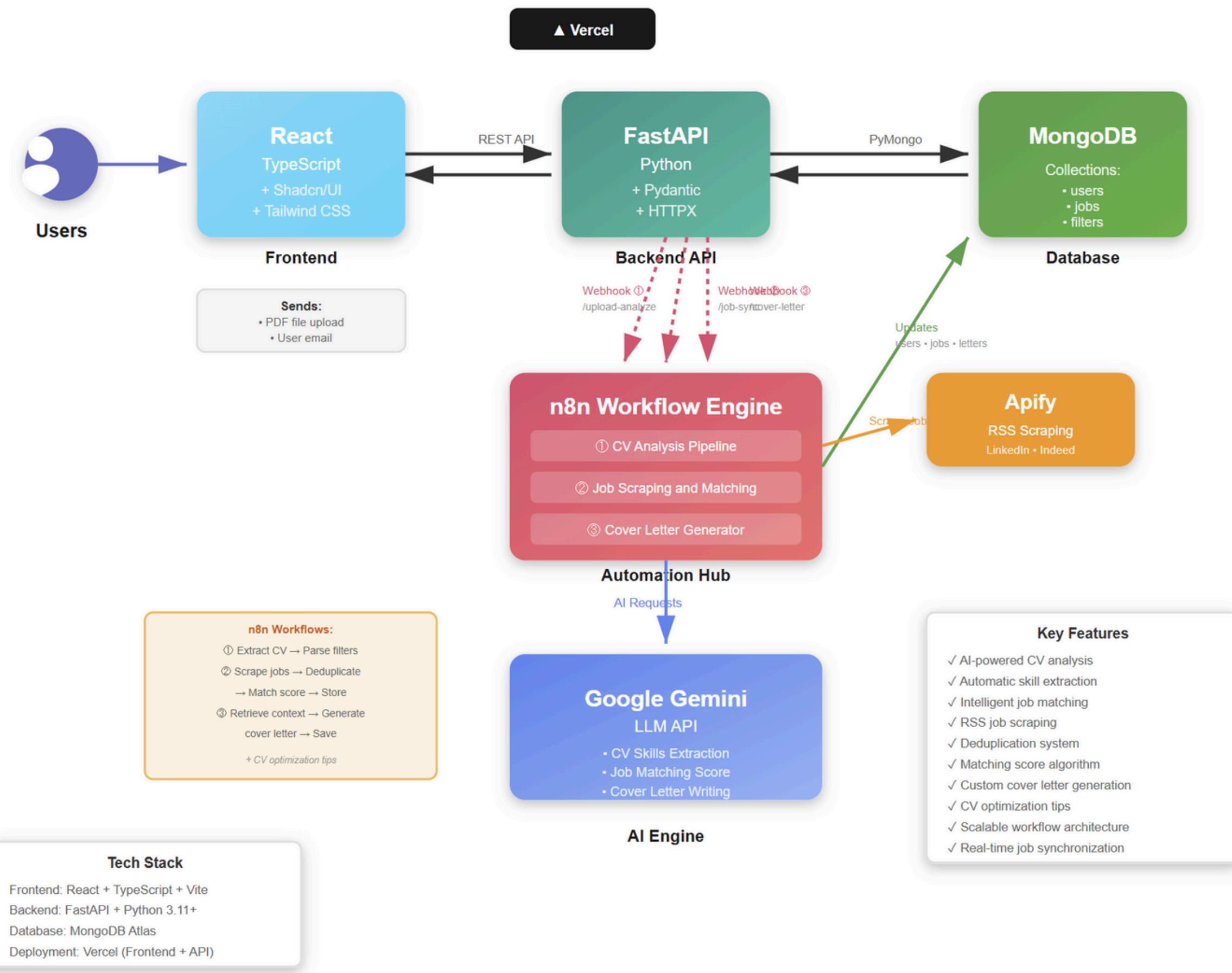
03

### One-Click Cover Letter Generation

Personalized draft letters combining your CV context with job requirements. Instant professional output **ready to edit**.



## Architecture





## GitHub Repository

Fragmented Job Search Process

Clear and modular repository structure  
for better maintenance and debugging.



```
evidi/
  |
  +-- backend/
  |    +-- public/
  |    +-- app/
  |    |    +-- routers/
  |    |    +-- __init__.py
  |    |    +-- db.py
  |    |    +-- models.py
  |    +-- tests/
  |    +-- main.py
  |    +-- pyproject.toml
  |    +-- requirements.txt
  |    +-- vercel.json
  |
  +-- frontend/
  |    +-- public/
  |    +-- src/
  |    +-- package.json
  |    +-- tsconfig.json
  |    +-- .env.example
  |
  +-- n8n/
  |    +-- workflows/
  |    |    +-- resume_analysis_pipeline.json
  |    |    +-- job_scraping_and_matching.json
  |    |    +-- cover_letter_generator.json
  |    +-- docs/
  |    |    +-- workflow_diagram.png
  |    +-- README.md
  |
  +-- docs/
  |
  +-- .github/
  |    +-- ISSUE_TEMPLATE.md
  |
  +-- CONTRIBUTING.md
  +-- README.md
```

# FastAPI application  
# Routes: auth, jobs, ai, criteria...  
# MongoDB client & collections  
# Data models  
# Entry point for Vercel  
  
# React + TypeScript web interface  
  
# Workflow automation layer  
# JSON exports of automation workflows  
  
# Architecture diagram  
# Explanation of workflow logic  
  
# Technical documentation (LaTeX, specs, diagrams)

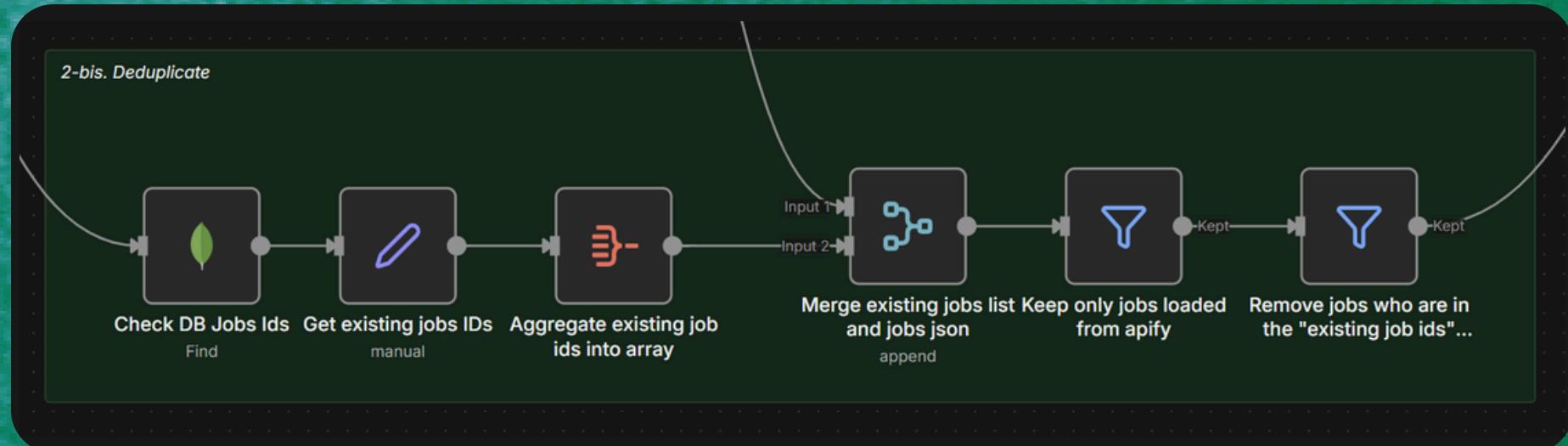


## Dataset & Sources

Source Type	Examples	Update Frequency	Volume/Week
RSS Feeds	TechJobs, RemoteOK	On 'Sync Now' click	~100 jobs
APIs	LinkedIn, Indeed, Glassdoor	On 'Sync Now' click	~100 jobs
Customized	(Depending on user entry)	On 'Sync Now' click	(Depending on source)
Total	10+ (expendable)	Discrete (for now)	~200+ (expendable)



## Deduplication



## Job Data Collection



## Data Standardization

All sources normalized to unified schema (title, company, skills, location, description, URL)



## AI Integration

Gemini 2.5 Flash API

# TECH STACK

Model

Gemini 2.5 Flash

Integration

n8n AI Agent node +  
FastAPI endpoints

Integration

CV parsing, job  
summarization, letter  
drafting

### # Overview

You are an expert technical recruiter specializing in [Execute previous nodes for preview] roles.

You have been given a job description and a candidate resume. Your task is to analyze the resume in relation to the job description and provide a detailed screening report to the candidate to help in understand why he matches the job or not.

Focus specifically on how well the candidate matches the core requirements and ideal profile outlined in the job description. Evaluate both technical skill alignment and business-content understanding. Use reasoning grounded in the actual content of the resume and job post -- avoid making assumptions.

You should only write in english throughout the entire process.

### ## Output

Your output should follow this exact format:

#### \*\*Requirements\*\*

List the 4 most important requirements (experience knowledge, business, etc.) mentionned in the job of description.

#### \*\*Required Skills\*\*

List the technical and business skills required given the job description.

#### \*\*Candidate Matched Skills\*\*

List the technical and business required skills that matches his skills from his resume and the stack from his filters as follows : [Execute previous nodes for preview].

#### \*\*Candidate Strengths:\*\*

List the top strengths or relevant qualifications the candidate

Performance

3-5 sec response time

Cost: ~\$5-10/month (prototype)



### What is n8n?

Workflow orchestration engine that manages complex multi-step processes across services, APIs, and AI models. Acts as an **intelligent middleware** between data sources and backend.

Aspect	FastAPI Backend	n8n Orchestration
Role	Data persistence, authentication, CRUD	Multi-service coordination, AI chains
Pattern	Request → Process → Response	Trigger → Transform → Route → Action
AI Integration	Single API call per endpoint	Multi-step AI agent workflows with context
Observability	Code-level logging	Real-time execution graph with data inspection

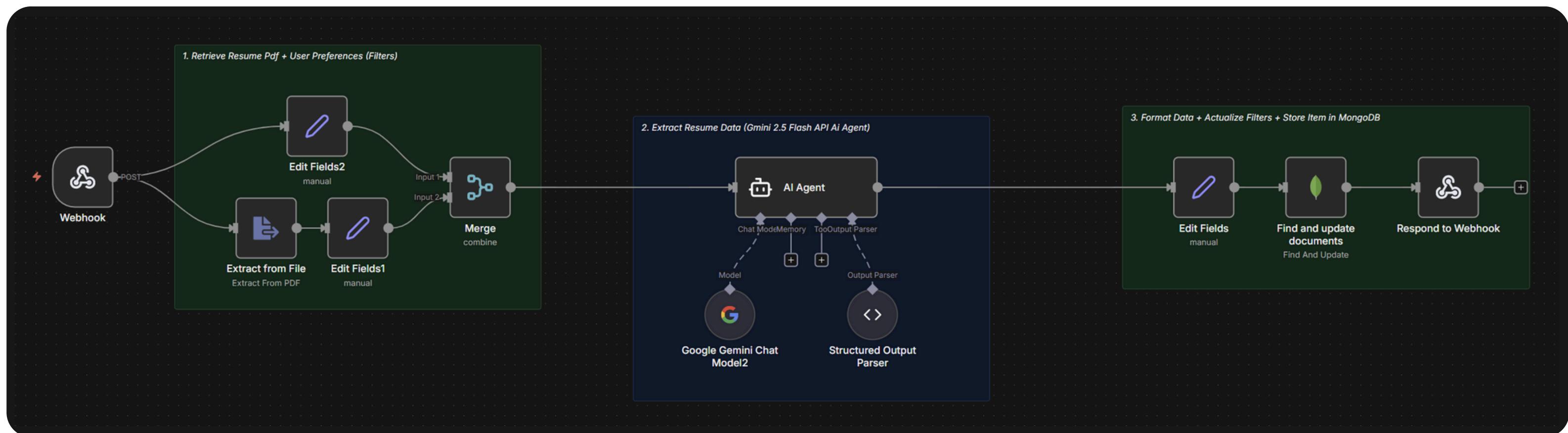
### Orchestration vs Traditional Backend

**Agentic AI Pattern**  
Sequential AI calls with context propagation. Example: CV skills → Job requirements → Personalized letter draft.  
Each step feeds the next, creating intelligent decision chains impossible with isolated API calls.



PDF Upload → Text Extraction → Gemini Skill Parsing  
→ JSON Validation → MongoDB Storage → User Filters Update

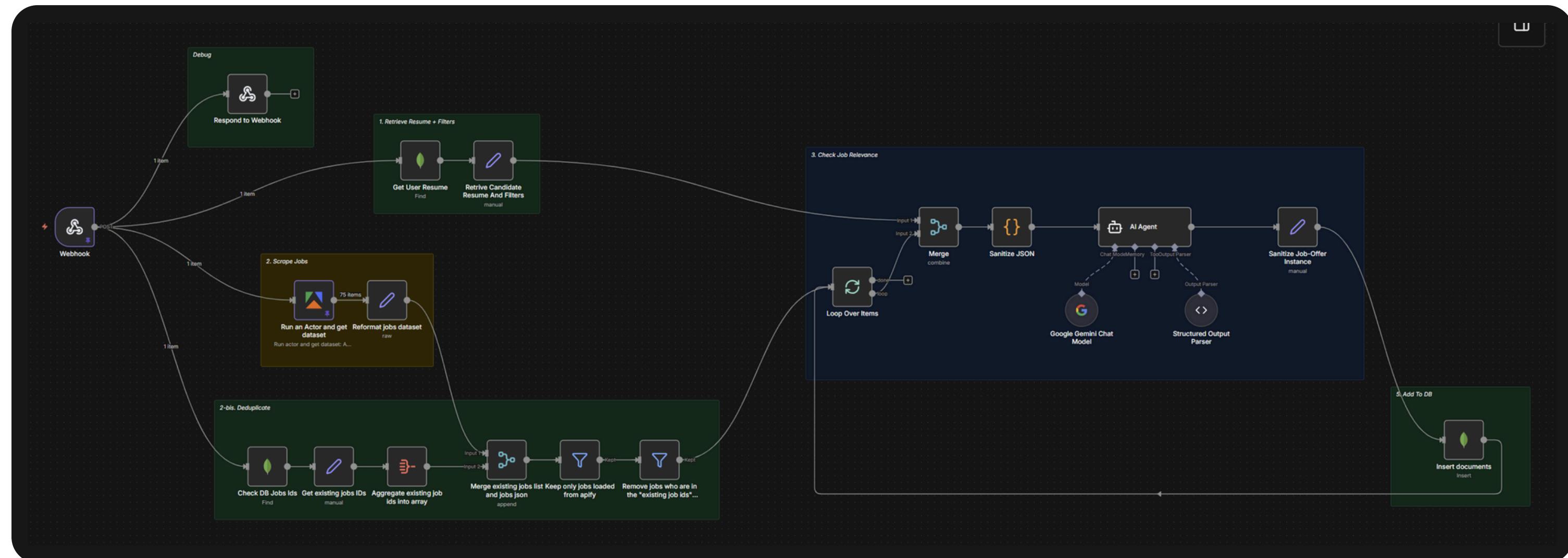
## Resume Analysis Pipeline





Apify Trigger (every 3h) → Normalize Data → Match Criteria  
→ Gemini Summarization → Relevance Scoring → Database Insert

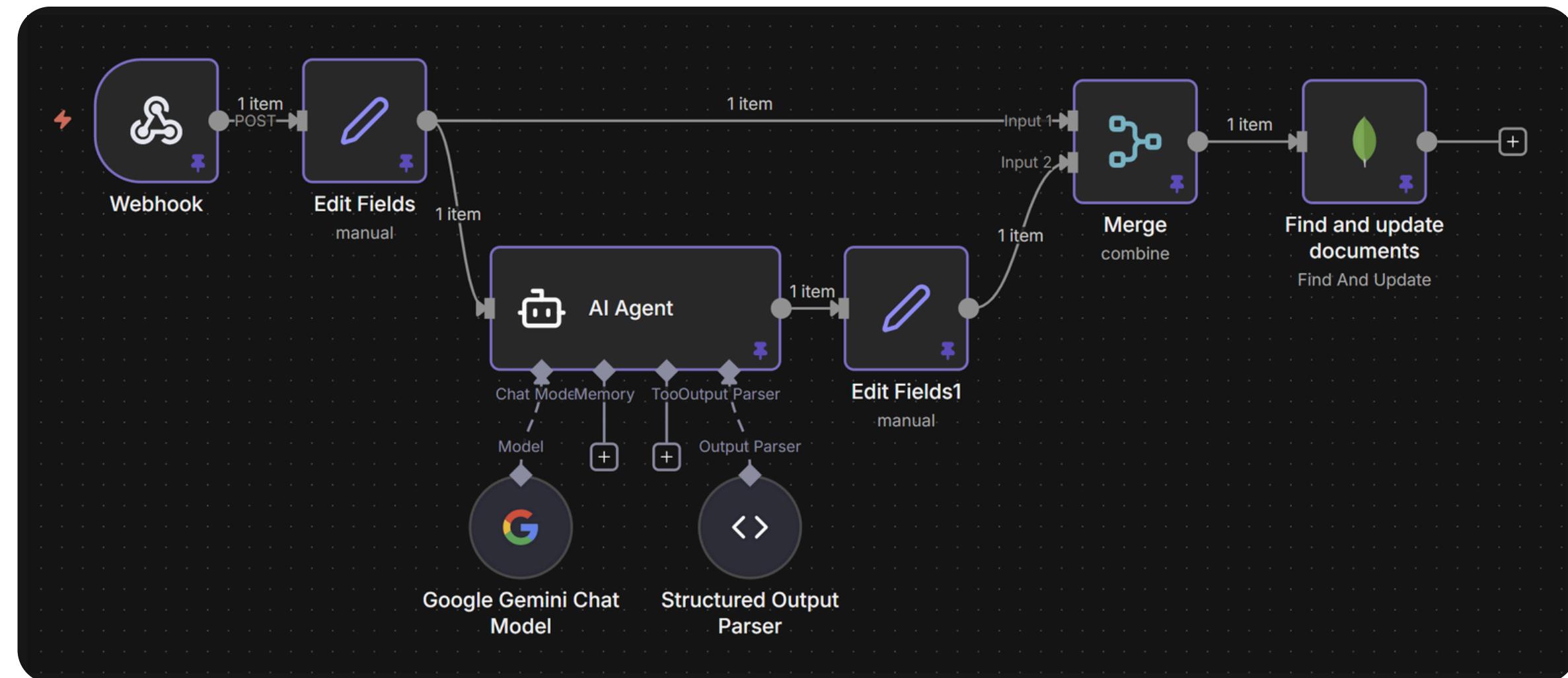
## Job Scraping & Matching

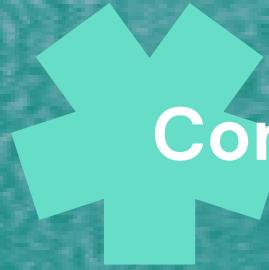


Job Selection → Fetch CV Context → Retrieve Job Details  
→ Gemini Draft Generation → Template Formatting → User Delivery

## Cover Letter Generator





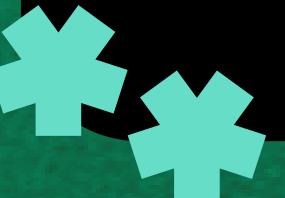


## Conclusion: Why Evidi matters?



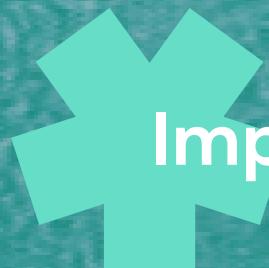
### Innovation Angle

Unlike closed commercial tools (Simplify, LazyApply), Evidi provides a **transparent, modular, and extensible open-source alternative**. Students and researchers can inspect, modify, and learn from the entire pipeline.



### Practical Impact

**Saves 5-10 hours/week** for active job seekers by automating discovery, analysis, and initial response generation.



## Improvements

01

### Enhanced AI Models

Test GPT-4o, Claude, and Gemini 3 for quality comparison.  
Implement user feedback loop and fine-tuning.

02

### Advanced Matching Algorithm

Add semantic embeddings (sentence-transformers) and historical pattern analysis for better relevance scoring.

03

### Expanded Integrations

Direct company career page scraping, GitHub job boards, and social media monitoring.





## Appendix

# TECH STACK

## Frontend

React 18 + TypeScript +  
Tailwind CSS

## Backend

FastAPI + Python 3.11 + Pydantic

## Database

MongoDB Atlas (NoSQL)

## Automation

n8n.cloud (Workflow  
Orchestration)

## AI Engine

Gemini 2.5 Flash API

## Deployment

Vercel + Railway

# TEAM

Documentation, Presentation  
KHEYAR Adel

Documentation, Presentation  
ARM Jules

Software Developer, Documentation  
DESJONQUERES Nicolas

AI, n8n Orchestration, Documentation,  
Presentation  
ROBIN Héloïse