### Internship Report(N8N)

### Mentor Name- Cladius Fernando, Yash Chavan

### HR Name- Ketaki Gokhale

### Trainee Name- Shubh Marwadi

### Topic- Gen-AI Workflow through N8N

### 1. Comprehensive Report on the Development of n8n-Based AI Agentic Systems

#### 1.1. Introduction

This report synthesizes the findings from a series of progressively complex AI agent development projects, all architected on the n8n workflow automation platform. The projects range from a foundational "Hello World" conversational agent to a sophisticated, multi-agent "NotebookLM" framework, demonstrating a clear roadmap for building complex AI systems. The core objective across all tasks was to explore, implement, and evaluate the creation of robust, agentic AI systems using a low-code, visual framework. These projects demonstrate the integration of various technologies, including Large Language Models (LLMs) via OpenRouter, specialized APIs for embeddings (Cohere), real-time web search (SerpAPI, Tavily), and a persistent vector database (Supabase) for long-term memory.

#### 1.2. Purpose of this Report

The purpose of this document is to provide a holistic analysis of the frameworks, tools, and methodologies employed throughout these projects. It will document the design decisions, compare the integrated technologies, and evaluate the overall effectiveness of n8n as a platform for building complex, multi-step AI solutions. This report will cover memory management, tool integration, workflow architecture, reasoning capabilities, and provide a detailed summary of the pros and cons of each major component, serving as a technical reference and a guide for future development.

### 2. The n8n Framework: A Foundational Analysis

Across all projects, n8n was used as the central nervous system for orchestration and automation. Its capabilities were pivotal to the success of each task.

* **GUI and Visualization:** n8n’s primary strength is its visual, node-based graphical user interface (GUI), which allows for the intuitive drag-and-drop design of agentic workflows. This visual paradigm was essential for managing the multi-step logic of the RAG and "NotebookLM" systems, with real-time feedback on data flow simplifying the debugging process.
* **Workflow Design and Modularity:** The framework inherently supports modular workflow design. This was demonstrated by creating separate, specialized sub-workflows for each agent (Preprocessing, Summarization, Storage, etc.) in the "NotebookLM" project, which were then orchestrated by a central

Main Workflow. This hub-and-spoke architecture allows each component to be developed, tested, and maintained independently.

* **Hosting and Deployment:** n8n offers both a fully managed cloud-hosted version and a Dockerized container for self-hosting. This provides flexibility, allowing developers to choose between ease of use (cloud) and full control over data and environment (self-hosted).
* **Tool Integration:** n8n excels as a universal API orchestrator. The projects successfully integrated a wide array of triggers (Telegram, Chat), APIs (OpenRouter, Cohere, Supabase, SerpAPI), and logic nodes (Code, Switch), showcasing its power in building cohesive systems from disparate services.

### 3. Task Analysis and Key Learnings

#### 3.1. Task 1 & 2: Foundational Chatbots ("Hello World Agent")

* **Objective:** To build a simple, cost-effective conversational agent to establish a baseline understanding of n8n's AI capabilities.
* **Key Components:** The workflow consisted of a Chat Trigger and an AI Agent node connected to the GPT-3.5 Turbo model via **OpenRouter**. OpenRouter was specifically chosen for its free access to high-quality models, flexibility, and ease of integration.
* **Memory Management:** This agent utilized n8n's built-in **Simple Memory**, which stores conversation history for a single session but is not persistent. While ideal for rapid prototyping, a plan was immediately formulated to integrate persistent memory solutions like MongoDB or Google Sheets for future iterations.
* **Conclusion:** The project successfully demonstrated that n8n and OpenRouter can be used to create functional chatbots quickly. The primary limitation identified was the ephemeral nature of the built-in memory.

#### 3.2. Task 3: Real-Time Web Searching

* **Objective:** To integrate live web search capabilities, mimicking platforms like Perplexity.ai by providing summarized answers based on real-time data.
* **Key Components & Comparison:**
  + **SerpAPI:** This tool connects to traditional search engines and returns raw, structured data (URLs, snippets). It is highly configurable but requires a subsequent LLM call to summarize the results for the user. Its free trial is very limited.
  + **Tavily:** This is a search API designed specifically for AI, returning summarized, human-readable answers directly. It requires minimal configuration and has a generous free tier, making it more efficient for Q&A workflows.
* **Conclusion:** **Tavily is superior for Perplexity-style flows** where a direct, summarized answer is the goal.

**SerpAPI is better for power users** who need granular, raw search data for more complex tasks.

#### 3.3. Task 4: Resume-based RAG Chatbot

* **Objective:** To build a chatbot that could ingest .docx resumes and answer questions about their content using a Retrieval-Augmented Generation (RAG) framework.
* **Key Components:**
  + **Supabase (Vector Store):** Chosen over Pinecone for its open-source flexibility, integrated PostgreSQL database, and developer-friendly free tier.
  + **Cohere (Embeddings):** Used to convert text chunks from resumes into high-quality vector embeddings for semantic search.
* **Memory Management:** Utilized Supabase and PostgreSQL for robust, persistent storage of both chat history and the embedded resume content, enabling user-specific memory across sessions.
* **Conclusion:** The project successfully demonstrated an effective RAG pipeline for document-based Q&A, proving the combination of Supabase and Cohere to be a strong foundation.

#### 3.4. Task 5: The "NotebookLM" Multi-Agent System

* **Objective:** To create a comprehensive, multi-agent system combining all previous concepts into a unified application.
* **Architecture:** A modular hub-and-spoke model with a Main Workflow orchestrating six specialized agents: Preprocessing, Summarization, Mindmap, Podcast, Storage, and Retrieval.
* **Reasoning:** The core reasoning is handled by the main AI Agent, whose **System Prompt** dictates the logic for choosing between a **Content Ingestion Path** and a **Retrieval Path** based on user intent. The agent demonstrates dynamic tool selection, for example, by calling the

Mindmap Agent when it detects relevant keywords.

* **Conclusion:** This final project confirmed that n8n is a highly capable platform for orchestrating complex, multi-agent AI systems, successfully integrating memory, reasoning, and a diverse set of tools into a single, functional workflow.

### 4. Technology Deep Dive: Pros, Cons, and API Keys

This section details the key third-party services integrated throughout the projects.

* **4.1. OpenRouter**
  + **API Key Location/Format:** User Account -> Keys (sk-or-...)
  + **Pros:** It offers a cost-effective and flexible way to access a wide variety of LLMs, which is ideal for experimentation and development. The integration with n8n is straightforward. The reasoning capability of models accessed via OpenRouter was strong for general conversational tasks.
  + **Cons:** The service may impose rate limits or require credit top-ups depending on usage. Documentation is not as extensive as some other providers, which may require more trial and error.
* **4.2. Cohere**
  + **API Key Location/Format:** User Dashboard (...)
  + **Pros:** Provides high-quality, fast, and multilingual embeddings, making it a strong choice for the embedding component of a RAG system. The API is easy to use and integrate into workflows.
  + **Cons:** The free tier has API rate limits that can be a bottleneck. It offers limited control over the dimensions of the output vectors and does not have an open-source variant.
* **4.3. Supabase**
  + **API Key Location/Format:** Project Settings -> API (URL & Anon Key)
  + **Pros:** As an open-source platform, it offers great flexibility, including the ability to self-host, avoiding vendor lock-in and allowing for easy data migration. It combines a traditional relational database with vector storage capabilities, streamlining the tech stack. Its free tier is generous and suitable for development.
  + **Cons:** Free projects are subject to being paused after a period of inactivity.
* **4.4. Tavily**
  + **API Key Location/Format:** User Dashboard (tvly-...)
  + **Pros:** It has a generous free tier, making it highly accessible. It is designed specifically for RAG and directly returns summarized, concise, and human-readable answers with minimal setup.
  + **Cons:** It requires manual setup to integrate with agent memory tools in n8n and is less configurable than more traditional search APIs like SerpAPI.
* **4.5. SerpAPI**
  + **API Key Location/Format:** User Dashboard (...)
  + **Pros:** It returns raw search engine results, including URLs and snippets, offering high flexibility and control. It is highly configurable and integrates well with n8n's agent memory tools.
  + **Cons:** The free trial is very limited with a low usage quota. It lacks built-in summarization, which requires an additional LLM call to process the raw data into a readable answer.
* **4.6. ElevenLabs**
  + **API Key Location/Format:** Profile Settings (xi-api-key)
  + **Pros:** (Inferred from its purpose in the project) It is renowned for its extremely high-quality, human-like text-to-speech audio.
  + **Cons:** The free tier is subject to strict abuse detection that can block users on VPNs or certain cloud IP addresses. Paid plans are required for commercial use and to bypass these network restrictions.

### 5. Final Conclusion and Recommendations

This series of projects successfully demonstrates the power and flexibility of using n8n to build sophisticated AI agentic systems. Starting from a simple chatbot and culminating in a multi-agent framework with persistent memory and diverse capabilities, the journey highlights key principles of modern AI application development.

**Key Findings:**

* **n8n as a Powerful Orchestrator:** Its visual interface, modular design, and extensive integration capabilities make it an ideal platform for managing complex, multi-step AI processes.
* **The Importance of Specialized APIs:** A successful system relies on integrating the best tool for each specific job, whether it's Cohere for embeddings, ElevenLabs for voice, or Tavily for RAG-focused search.
* **Persistent Memory as a Core Component:** For an agent to be truly useful and provide a personalized experience, a persistent long-term memory solution, such as the Supabase vector store, is crucial.

**Recommendations for Further Exploration:**

* **Integrate General Web Search:** Add a tool like SerpAPI or Tavily to the final "NotebookLM" project to allow the agent to answer questions about topics not contained in its stored knowledge base.
* **Enhance Error Handling:** Build automated error-handling branches within the workflows to gracefully manage external API failures and provide informative feedback to the user.
* **Upgrade to Paid API Tiers:** For production use, upgrading key services would remove free-tier limitations, increase reliability, and unlock more powerful features.
* **Implement Dynamic Routing Logic:** Create more sophisticated logic in the Main Workflow to dynamically choose between different LLMs or tools based on the perceived complexity or category of the user's request.
* **Integrate Observability Tools:** For production monitoring, integrate dedicated observability platforms to track performance, latency, and API costs across the entire system.

Thank You