## 🎓 University Student Support RAG Agent – Workflow Documentation

﻿

### Overview

The **University Student Support RAG Agent** is an intelligent, AI-powered assistant designed to help students get instant, accurate answers to university-related questions such as admission details, course requirements, deadlines, and campus support services.

It uses a **Retrieval-Augmented Generation (RAG)** approach — combining **real-time data retrieval from a vector database (Supabase)** and **natural language generation via Google Gemini**.

This system consists of two main parts:

**Chat Agent Workflow** – Handles student interactions in real-time.

**Vector Database Builder Workflow** – Automates knowledge base creation and updates from Google Drive documents.

﻿

## 🟨 Section 1: Student Chat Agent Workflow

### Purpose

This part manages **real-time student communication**. It takes user queries, retrieves relevant information from Supabase, and generates a contextual, human-like answer using Google Gemini.

﻿

### Step-by-Step Flow Description

1️⃣ **When Chat Message (Trigger Node)** **Function:** Starts the workflow whenever a student sends a new chat message. **Purpose:** Initiates the pipeline that processes the question. **Example:** Student types “What are the admission requirements?”

2️⃣ **Google Gemini Embeddings (Query Embedding)** **Function:** Converts the student’s query into a **semantic embedding (vector)**. **Purpose:** Enables similarity-based search in Supabase. **Example:** “Admission details” and “application info” produce similar embeddings.

3️⃣ **Supabase Vector Store (Retrieve Documents)** **Function:** Searches Supabase for the most relevant text chunks related to the student’s question. **Query Function:** `match\_university\_student\_support\_agent\_data()` **Output:** Returns top results with similarity scores and metadata. **Purpose:** Finds the most relevant documents or data chunks.

4️⃣ **AI Agent (Gemini Chat Model)** **Function:** Combines retrieved context + conversation history + query to generate an accurate response. **Model:** Google Gemini Chat **Purpose:** Acts as the reasoning and generation engine. **Output:** Natural, contextual answer. **Example:** **Input:** “What are the admission requirements?” **Response:** “To apply for admission, you must submit transcripts, ID, and fill out the online form by March 1st.”

5️⃣ **Simple Memory** **Function:** Stores previous messages and answers to maintain conversation continuity. **Purpose:** Enables follow-up questions and human-like interactions. **Example:**

Student: “When is the deadline?”

Agent: “March 1st.”

Student: “Can I submit late?” → Context remembered from previous response.

﻿

### Outcome

The chatbot delivers relevant and reliable answers in real time using up-to-date data from the knowledge base — providing a seamless, interactive student support experience.

﻿

## 🟩 Section 2: Vector Database Builder (Knowledge Base)

### Purpose

This section automatically updates the **Supabase vector database** when new university documents are uploaded to **Google Drive**. It ensures the chatbot’s knowledge base stays accurate and current without manual intervention.

﻿

### Step-by-Step Flow Description

1️⃣ **Google Drive Trigger** **Function:** Monitors a specific Google Drive folder for new or updated files. **Purpose:** Starts the workflow automatically whenever new content is added. **Example:** “University\_Admission\_Info.pdf” uploaded → triggers workflow.

2️⃣ **Set Created Values (Optional)** **Function:** Defines variables or metadata related to uploaded files (like file name, folder, or ID). **Purpose:** Ensures consistent file handling in the next steps.

3️⃣ **Download File** **Function:** Downloads the detected file from Google Drive. **Purpose:** Makes the file accessible for text extraction.

4️⃣ **Switch (File Type Detection)** **Function:** Checks the file format (PDF, TXT, CSV, etc.). **Purpose:** Routes the file to the correct extraction process.

5️⃣ **Extract from File / Extract from CSV** **Function:** Extracts readable text from PDF, TXT, or CSV documents. **Purpose:** Converts file content into a clean text format for processing. **Output:** Raw document text.

6️⃣ **Character Text Splitter** **Function:** Splits long text into smaller “chunks” (usually 500–1,000 characters). **Purpose:** Smaller chunks improve search accuracy and reduce token costs. **Example:** A 10-page document → 50 small text chunks.

7️⃣ **Google Gemini Embeddings (Chunk Embeddings)** **Function:** Converts each text chunk into a vector embedding. **Purpose:** Allows semantic similarity comparison between queries and document content.

8️⃣ **Supabase Vector Store (Store Embeddings)** **Function:** Inserts embeddings, content, and metadata into Supabase. **Purpose:** Builds and maintains the knowledge base. **Example Data Stored:**

Chunk text

Embedding vector

File name (metadata)

﻿

### Outcome

The system continuously builds and updates the knowledge base automatically — ensuring the AI agent always has access to the latest and most accurate information.

﻿

## 🧠 How RAG Works in This Workflow

Retrieval-Augmented Generation (RAG) = Retrieval + Generation

**1. Retrieval:** When a question is asked, the system retrieves the most relevant information from Supabase based on vector similarity.

**2. Augmentation:** The retrieved content is appended to the prompt sent to Gemini.

**3. Generation:** Gemini reads both the user question + retrieved content and generates a coherent, accurate answer.

This makes the chatbot more factual, contextual, and trustworthy.

﻿

## 🧩 Key Components

|  |  |  |
| --- | --- | --- |
| Component | Technology | Purpose |
| Chat Agent | Google Gemini | Generates natural responses |
| Vector Database | Supabase | Stores text embeddings for retrieval |
| Embeddings Generator | Gemini API | Converts text → numerical vectors |
| Memory Node | n8n Simple Memory | Maintains conversation context |
| Trigger | Google Drive | Automates database updates |
| Storage | Supabase | Saves embeddings and metadata |

﻿

## 🖼️ Recommended Diagram Structure (for your Word file)

You can add a visual layout like this:

+-------------------------------+| Chat Agent Section ||-------------------------------|| 1. Chat Trigger || 2. Gemini Embedding (Query) || 3. Supabase Search || 4. Gemini AI Agent || 5. Simple Memory |+-------------------------------+﻿ ↓﻿+-------------------------------+| Vector Database Builder ||-------------------------------|| 1. Google Drive Trigger || 2. Download File || 3. Extract Content (File/CSV) || 4. Split Text into Chunks || 5. Gemini Embeddings || 6. Store in Supabase |+-------------------------------+

﻿

## 🏁 Final Outcome

**For Students:** They get instant, contextual, and accurate answers to their questions — no waiting for email responses.

**For the University:** Automation reduces support workload and ensures 24/7 availability.

**For the Developer/Admin:** Easy maintenance — simply upload or update files in Google Drive, and the AI knowledge base updates automatically.

﻿

## 🧱 Tech Stack Summary

|  |  |
| --- | --- |
| Layer | Tools/Tech |
| AI Engine | Google Gemini (Chat + Embeddings) |
| Database | Supabase (PostgreSQL + pgvector) |
| Orchestration | n8n Workflow Automation |
| Source Storage | Google Drive |
| Knowledge Base Type | Vector Store (768-dimension embeddings) |
| Integration Function | match\_university\_student\_support\_agent\_data() |

﻿