**Resume-based RAG (Retrbatanaieval-Augmented Generation) Chatbot with Supabase and Cohere**

**Introduction**

This report presents a detailed analysis of a Resume-QA chatbot system built on a Retrieval-Augmented Generation (RAG) framework. The chatbot is capable of processing resumes uploaded in .docx format, converting them into text, and enabling question-answering via chat. It leverages Supabase as a vector store, Cohere for embeddings, and OpenRouter for conversational AI. The main goal is to explore the capabilities and implementation of this RAG pipeline and compare it with alternative technologies like Pinecone.

**Brief Overview of Framework**

The architecture consists of two primary workflows:

1. **Resume Ingestion & Embedding**: Extracts text from resume files, splits it, generates embeddings via Cohere, and stores them in Supabase Vector Store.
2. **Chat-based Question Answering**: Uses AI Agent memory stored in PostgreSQL and semantic search from Supabase Vector Store to answer queries based on resume content.

**Purpose of the Report**

The purpose of this report is to document the implementation of the Resume-QA chatbot, explain the design decisions (e.g., Supabase over Pinecone), and evaluate the framework based on critical features including memory management, observability, web search tools, and reasoning capabilities.

**Features Comparison**

| **Feature** | **Supabase** | **Pinecone** |
| --- | --- | --- |
| Cloud-hosted version | Yes | Yes |
| Dockerized container version | Yes (via self-hosting) | No (fully managed) |
| GUI and visualization | Supabase Dashboard | None |
| Commercial license | Open Source (MIT) | Proprietary (paid tiers) |

**Why Supabase over Pinecone?**

* **Open-source flexibility**: Supabase can be self-hosted.
* **Integrated DB**: Combines vector storage with relational PostgreSQL.
* **No vendor lock-in**: Open architecture allows export/migration.
* **Free tier friendly**: Suitable for testing and development phases.

**Memory Management**

| **Feature** | **Available? (Our Build)** |
| --- | --- |
| Built-in Memory | Yes (PostgreSQL) |
| Session Storage | Yes |
| User-specific Memory | Configurable via Sessions |

Supabase and PostgreSQL offer robust support for maintaining chat history and context between sessions.

**Web Search Tools**

| **Feature** | **Details** |
| --- | --- |
| Tools Library | Integrated in OpenRouter & AI Agent |
| Configuration Options | Vector store selection, API endpoints |
| Performance | Good (depending on Cohere model used) |
| Accuracy | High for factual retrieval tasks |

**Toolkits and APIs**

| **Feature** | **Description** |
| --- | --- |
| Prebuilt Toolkits | Google Drive, Text Extract, Chat Agent |
| API Support | Supabase, Cohere, OpenRouter |
| Observability Compatibility | Logs, vector index tracking via Supabase UI |

**Workflow Analysis**

The framework supports modular workflows:

* Drag-and-drop UI for AI agent design.
* Clear task pipelines for ingestion and question-answering.
* Support for multiple agents and memory types.

**Reasoning and Thinking Tools**

* **RAG Design**: Blends semantic search with generative LLM output.
* **Memory Context**: Maintains context across sessions.
* **Cohere Embeddings**: Provide high-quality semantic vector space.

**Pros and Cons of Cohere**

| **Pros** | **Cons** |
| --- | --- |
| Easy to use and integrate | API rate limits in free tier |
| High-quality multilingual support | Limited control on vector dimension |
| Fast inference speed | No open-source variant |

**Conclusion**

The Resume-QA Chatbot demonstrates an effective application of Retrieval-Augmented Generation using Supabase, Cohere, and OpenRouter. The system provides a strong foundation for document-based question answering.

**Recommendations**

* Explore hybrid vector stores (e.g., Weaviate + Supabase).
* Integrate web search tools for real-time data augmentation.
* Test other embedding providers like OpenAI or HuggingFace.
* Implement role-based access for secure memory storage.