Structure of Implementation of Multiple Agentic RAG System

Creating a Multiple Agentic RAG (Retrieval-Augmented Generation) System involves designing a set of collaborating agents with well-defined roles, each equipped with specific tools, to enhance reasoning and retrieval for complex tasks. Below is the full implementation structure.

- Agents are role-specialized (e.g., Retriever, Researcher, Synthesizer).
- Tools are assigned per-agent based on their function.
- Communication follows a GroupChat or Coordinator pattern.

AGENTS & THEIR RESPONSIBILITIES

1. UserProxyAgent

- **Role**: Acts as the user interface. Receives user prompts and sends them to the system.
- **Why Needed**: Helps simulate user interaction in AutoGen.

2. ResearcherAgent

- **Role**: Understands the query, refines it for better search, and initiates the retrieval process.
- **Tools Used**:
  - QueryRefiner: Improves vague or broad user queries.
  - RetrieverInterface: Passes refined query to the RetrieverAgent.
- **Output**: Refined query and fetched documents.

3. RetrieverAgent

- **Role**: Connects with the vector database (e.g., ChromaDB) to fetch relevant information.
- **Tools Used**:
  - ChromaDBRetrievalTool: Executes the actual query against the DB and returns top-k results.
- **Output**: List of document chunks with metadata.

4. SummarizerAgent

- **Role**: Synthesizes retrieved information into a final, readable, contextual answer.
- **Tools Used**:
  - SummarizerTool: Uses LLM (Gemini/GPT-4) to create human-like answers.

- **Output**: Final answer draft.

5. CriticAgent (Optional)

- **Role**: Reviews the summarizer’s response and checks for errors, omissions, or improvements.
- **Tools Used**:
    - EvaluationTool: Uses prompting or rules to assess quality.
    - AnswerImprover: Suggests or applies edits.
- **Output**: Refined, polished answer.

Tools

Retrieval Tools

- **QueryRefiner**: Enhances vague queries.
- **ChromaDBRetrievalTool**: Queries vector database for relevant chunks.
- **RetrieverInterface**: Allows agent-to-agent document retrieval.

Generation Tools

- **SummarizerTool**: Creates human-readable answers.
- **AnswerImprover**: Refines tone, clarity, completeness.

Evaluation Tools

- **EvaluationTool**: Scores answer quality, fact-checks.

Backend Tools

- **Chunker**: Splits unstructured data into retrievable chunks.
- **Embedder**: Converts chunks to embeddings for ChromaDB.
- **DatabaseManager**: Manages course metadata in SQLite/MongoDB.

System Workflow

Frontend Workflow

1. **User Query** sent to UserProxyAgent.
2. **ResearcherAgent** refines and sends it to the Retriever.
3. **RetrieverAgent** fetches documents from ChromaDB.
4. **SummarizerAgent** generates an answer.
5. **CriticAgent** improves the answer (optional).
6. User receives a final personalized response.

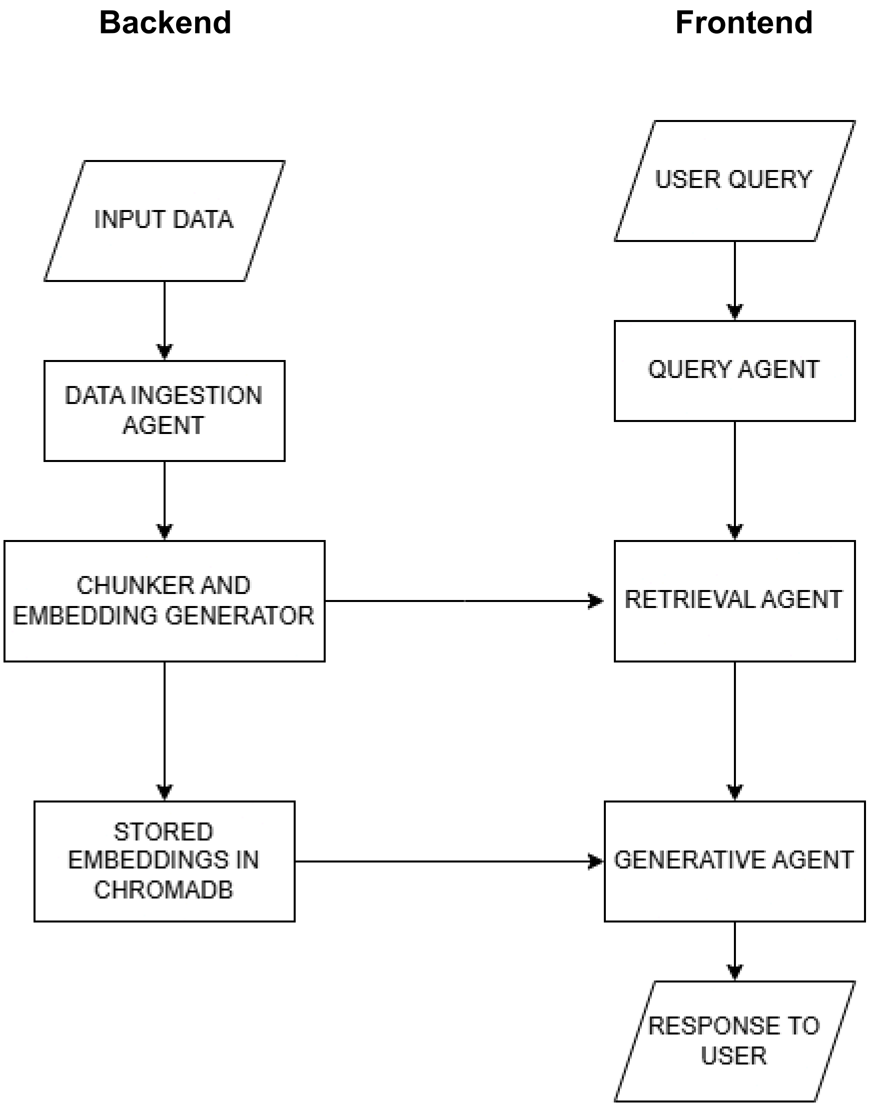
Backend Workflow

1. Admin inputs course data (course info, outcomes, syllabus).
2. Data saved in SQLite/MongoDB using DatabaseManager.
3. Unstructured data chunked and embedded.
4. Embeddings stored in ChromaDB for retrieval

Output Formation

- **Admin Data Ingestion:**
  Stores structured data in SQLite3 or MongoDB using the Data Ingestion Agent.
- **Data Chunking & Embedding**:
  Generates chunks and embeddings stored in ChromaDB using the Chunker and Embedding Generator.
- **User Question Processing:**
  Retrieves relevant chunks from ChromaDB via User Query Agent and Retrieval Tool.
- **Answer Generation:**
  Produces a syllabus-aligned, personalized final answer using the Response Generator and Personalizer.
- **User Feedback Processing**:
  Enhances future answers via Feedback Analysis Agent for refined retrieval and personalization.

FLOWCHART


Backend
Frontend
INPUT DATA
DATA INGESTION AGENT
CHUNKER AND EMBEDDING GENERATOR
STORED EMBEDDINGS IN CHROMADB
USER QUERY
QUERY AGENT
RETRIEVAL AGENT
GENERATIVE AGENT
RESPONSE TO USER