New Hire Challenge

Quality Assurance Engineer – AI Product Line
Agentic RAG Chatbot Challenge with Crew AI, Chonkie, and DeepEval

Objective:
Candidates will develop a Retrieval-Augmented Generation (RAG) chatbot capable of "agentic" behavior. The chatbot should leverage **Crew AI** (or similar agent framework) to handle agentic tasks and incorporate **Chonkie** for chunking source documents. The solution must also include an evaluation step using **DeepEval** or a comparable framework to measure retrieval and generation performance.

Deadline:
You have 66 hours from receiving this message to submit a link to a GitHub repository containing your full solution. Any code commits beyond this time will not be considered. A non-compiling solution does not necessarily disqualify you—clear explanations of your design and problem-solving approach are highly valuable.

Getting Started:
Feel free to explore existing tutorials, guides, or code snippets on agentic RAG, Crew AI, Chonkie, and DeepEval. Your solution will be assessed on design clarity, rationale, and documentation as much as on functionality. In particular, detailing why and how you employ Chonkie, DeepEval, and other tools is highly recommended.

Evaluation Criteria

1. End-to-End RAG Chatbot

- **Agentic Behavior:** Demonstrate how your choice of agent framework is integrated to manage dynamic tasks (e.g., deciding whether to retrieve more context, refine the prompt, or request clarifications).
- **Chunking Implementation:** Use Chonkie to partition documents into optimally sized chunks. Clearly document your chunking strategy and explain its impact on retrieval (use some advance semantic chunking for bonus points).
- **Architecture:** Choose a vector database (e.g., Qdrant, Milvus, Pinecone), agent framework, and evaluation tool that best suit your project. Provide reasoning behind each choice and how you integrated them, highlighting factors such as scalability, ease of setup, or unique features.

2. RAG Evaluation with DeepEval (or Similar)

Your assessment should show both retrieval and generation quality:

1. **Context Retrieval**
    - **Contextual Precision:** How often the system retrieves highly relevant chunks.
    - **Contextual Recall:** Ensures that the system captures all necessary context from the database.
    - **Contextual Relevancy:** Measures how closely the returned context aligns with the user query.
2. **Content Generation**
    - **Answer Relevancy:** Checks if the chatbot's generated response truly addresses user inquiries.
    - **Faithfulness:** Confirms the response remains grounded in the retrieved context, avoiding extraneous or false details.

You are welcome to add extra metrics or methods for evaluation if they offer additional insights (e.g., side-by-side comparisons, user testing).

HAPPY CODING!!! 👷