Task Title: RAG ChatBot

Live Coding Interview
Duration: 1.5 hours

---

1. Objective

Design and implement a fully functional RAG-powered chatbot:

- Parse, chunk and embed given document
- Perform retrieval over a supplied document.
- Re-rank and optimize retrieved passages.
- Synthesize coherent answers to user questions.
- Suggest relevant follow-up questions.
- Maintain full conversational context across multiple turns.
- Optionally answers with relevant images (Multimodal RAG is optional)

Here is a link to sample document

https://drive.google.com/file/d/1o7m_pbp70mWxeS6zCjRytoxjAJ8uf7Xy/view?usp=sharing

2. Scope & Requirements

- **Backend Agent (Python):** Create a service (FastAPI etc.) that handles retrieval, re-ranking, query optimization, and answer synthesis using any libraries or AI tools of your choice.
- **Retrieval Logic:** Implement both semantic search and vector search; combine results for best coverage.
- **Context Management:** Track the entire chat history and determine when to invoke retrieval versus using existing context to answer.

3. Optional Implementations

These are things that are nice to have if implemented, and will be given as bonus points

- **Chat UI**: Build a clean, responsive chat interface that lets users type questions, displays bot replies, and shows follow-up suggestions as tappable chips.
- **Answering with relevant images:** Optionally agents might show images while explaining the corresponding content. (This task is optional and nice to have it will not be main eval criteria)
- **Evaluation**: Optionally provide evaluation of each step to improve further operations (RAGAS metrics etc.)

3. Deliverables

1. **Optional UI** with chat interface and follow-up suggestion chips.
2. **Python service** implementing RAG, re-ranking, and query-optimization logic.
3. **Documentation:** README explaining setup, architectural decisions, and how technical implementations are handled.

4. Evaluation Criteria

- Quality of retrieval: relevance and accuracy of answers.
- Efficiency: proper use of semantic vs. vector search and re-ranking.
- Context-awareness: correct handling of multi-turn dialogue.
- UI/UX: responsiveness, clarity of message flow, and follow-up suggestions.
- Code structure, readability, and use of best practices.

Notes

- *You are free to prototype or implement agent logic using ChatGPT, Cursor, Windsurf, Google Search or similar AI development tools.*
- *OpenAI API Key:*

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

- *Google GenAI API Key:*

AIzaSyCoqz8ANvFz-XC-n7iQEhh_cmOyHStb-Io

- *Anthrophic API Key:*

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

- *If you need any other api key, you can request it during the interview*

Good luck!