Coding Challenge // Scaling the Chatbot's Knowledge Base While Preserving Quality

We do not expect you to tackle every single detail. Instead, we want to understand how you think, which aspects are important to you, and how you structure your work. Please do not invest more than 2 hours into the preparation of the challenge.
The way you present your work is completely up to you.

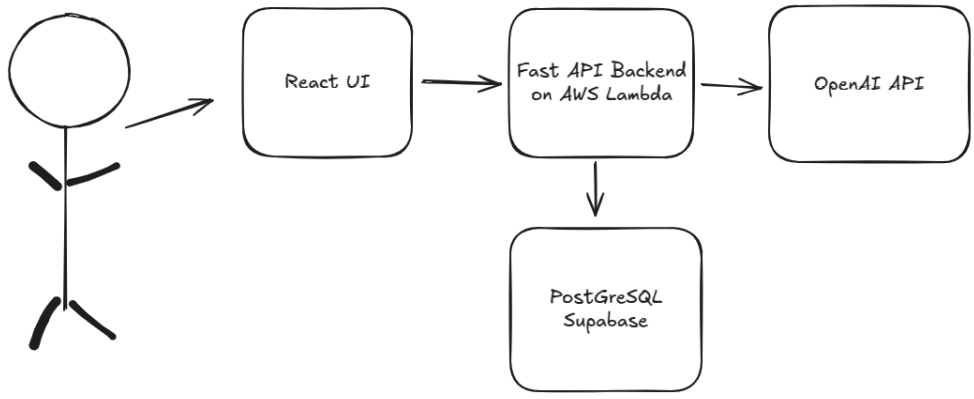
You are responsible for extending and refining the RAG capabilities of an existing OpenAI-powered chatbot deployed on AWS. The chatbot is in an MVP stage and the current answers have received good feedback in beta tests. We now want to add our blog posts (https://checkmk.com/blog) as a new knowledge source.

Your task:

1. Propose a strategy for integrating the new knowledge source.
2. Identify challenges this integration might introduce.
3. Suggest solutions to prevent potential issues.
4. The chatbot continues to provide **accurate and relevant answers**.
5. Adding the blog posts to the knowledgebase does not **introduce inconsistencies** or lower response quality.

Background:

The following diagram gives you a high level overview about the current retrieval architecture of the chatbot:



High level description:

- React UI: Frontend for the chatbot

- FastAPI Backend: Where the chatbot logic lives
- PostgreSQL DB: used to store the vector for additional resources and other chat related data.
- OpenAI API: Used to interact with LLM

Additional Information:

- Our current setup includes 2 public data sources from Checkmk:
    - Documentation (https://docs.checkmk.com/latest/en/)
    - Vendor integrations (https://checkmk.com/product/integrations)
- The system performs a full rebuild of the entire knowledge base once an hour