Fabian: Personal RAG-based Chatbot

1. Project Description

To design and develop local opensource LLM powered end to end personal chatbot. Chatbot will be RAG-based, admin can upload documents to vector database, chatbot will produce responses as per the provided knowledge.

Core Features

Local LLM powered RAB-based chatbot:

- Chatbot is powered by Llama3.1 (8B).
- Vector database (Chroma DB) is hooked with the LLM so it will produce response based on the custom knowledge.
- Hallucination of the LLM is avoided using proper prompting and custom factual knowledge.
- Admin can add and remove any document from the vector database.
- This whole system will run locally without any internet connection and no paid API.

2. Technology Stack

- **UI/UX:**
    - Figma Designing Wireframes for UI/UX design and prototyping of frontend.
- **Frontend:**
    - React.js for creating a responsive frontend.
- **Backend:**
    - Python for server-side logic and AI integration.
    - Fast API Framework for backend development.
    - Llama3.1 (8B) as large language model.
    - Langchain framework to integrate LLM with vector database and make RAG-based chatbot.
- **Databases:**
    - Chroma DB as vector database for integrating custom knowledge.
    - Postgres database for structured data (e.g. user information i.e. username, password etc.).

    - MongoDB for unstructured data (e.g. user conversation sessions and history).

- **Hosting:**
    - All the software will be hosted and run on local machines for this prototype.
    - No internet, no paid APIs or external paid tool are required for hosting or development.
    - A minimum of 16GB of RAM will be required for this prototype to run normally for single user.
    - NVIDIA GPU (specs depends on your budget, better the GPU, faster would be the chatbot) would be required if you want this solution to scale up for multiple users.

3. Development Milestones

Milestone 1: Proof of Concept (POC)

- Simple POC will be prepared in Streamlit for the Chatbot.
- POC will demonstrate the RAG capabilities where users will upload documents and the chatbot will produce answers as per the provided knowledge.
- No advanced login or admin authentication would be done in POC. It will be done in Final product for production to save cost, time and resources.
- POC will run 100% locally without any internet connection or paid APIs
- Demo of POC will be given to client for approval to work on final product.

Milestone 2: Chatbot Prototype and Frontend Development

- Frontend will be implemented in React.js for the smooth user interface.
- Implement backend logic using Python and Fast API for the AI chatbot workflow.
- Use Langchain to connect the RAG-based chatbot with Chroma DB vector database.
- Postgres database will store the user's information like username and password.
- MongoDB will be used to store the users' sessions and conversations history.
- Integrate all the backend with the frontend.

Milestone 3: Testing and Quality Assurance (QA)

- Conduct unit testing to test individual features in isolation.
- Conduct integration testing to check and test the system as a whole.
- After thorough testing, bugs and errors would be fixed if any.

Milestone 4: Deployment and handover

- Deploy the system on local machine on your side.

- Perform final checks and launch the system for production.
- Documentation will be provided on how to install all the software on local machines.
- I will be available for support, or any help required over the whole process.

4. User's Roles

- **Admin User:** Full backend and admin access to CRUD (Create, Read, Update and Delete) operation. Admin will be super-user, can add or remove any document or user or their data.
- **Normal Users:** Access to Chatbot and their own conversation history inside the chatbot like ChatGPT on the sidebar.

5. Client communication, requirements and support

Communication:

- The client will be updated almost daily on the progress of development via communication channel (I prefer slack).
- If there is any blockage or confusion that requires the client's involvement, the client is expected to communicate with me, I expect and appreciate prompt communication so that I deliver results on promised or before deadline.
- A weekly meeting would be arranged with the client (I will choose the time that is suitable for you), all the demos and progress would be shown in that meeting.
- I will communicate very often to remain transparent and clarify any confusion and misunderstandings.

Requirements from Client:

- During development of this chatbot, nothing is required from the Client however the client must have the required local machine (at least 16 GB RAM) so that I am able to deploy and run this software on his machine.

Extra Support:

- I will provide support for the software developed for up to 10 days from the date of launch if required.
- If support is required after 10 days, an extra per hour cost will be charged for it.

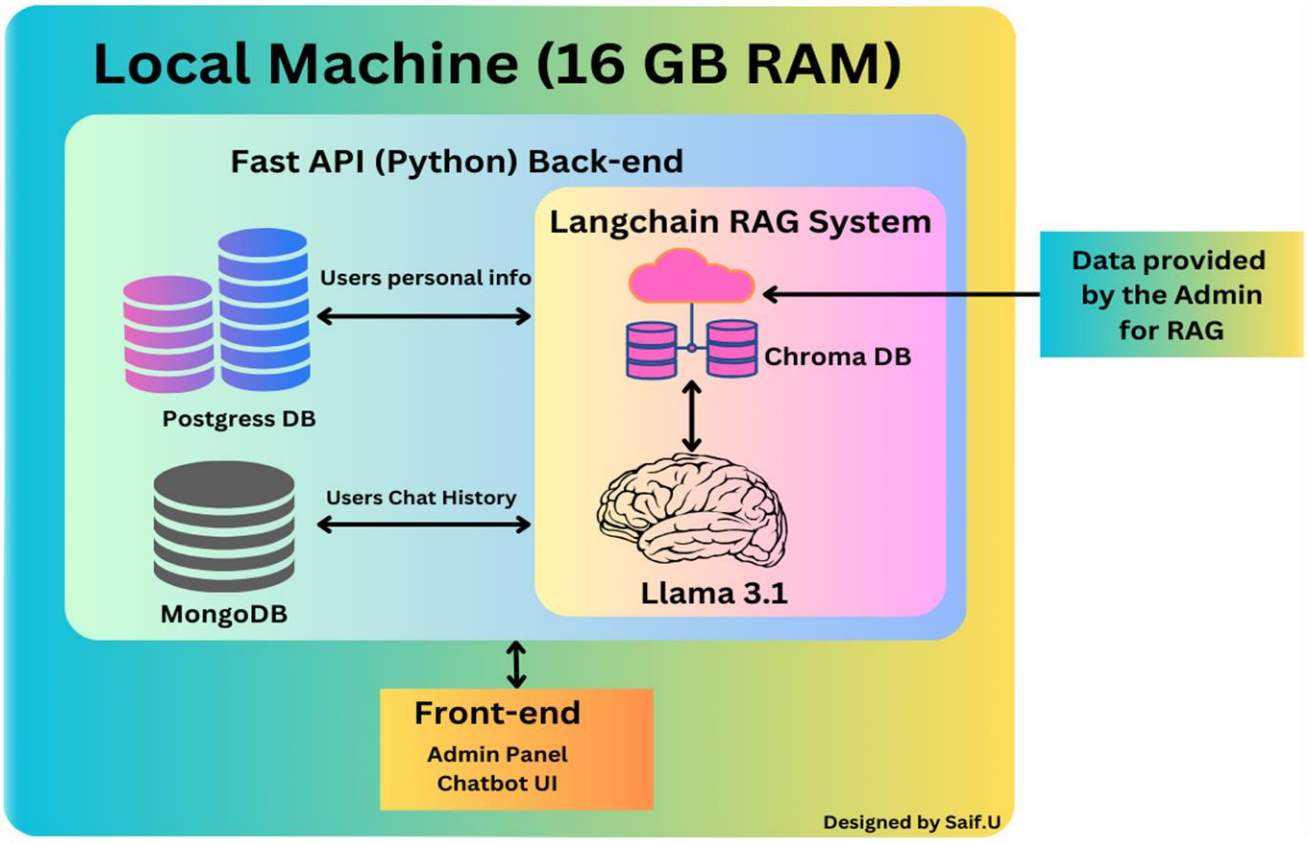
6. Software Flow

Login Page:

- **Admin Panel Redirection:**
    - After login, admins are redirected to the Fast API Admin Panel.
    - **CRUD Operations:** Admins can perform Create, Read, Update, and Delete operations on the database, manage user accounts, upload and remove documents.
    - Admins will be provided secret keys to be used during signup for the first time.

- **Chatbot Redirection:**
    - Users will be redirected to the Chatbot user interface.
    - Normal users must signup for the first time with their email, username and password.
    - After login, normal users will be redirected toward chatbot UI created in React.js.
    - Users can chat with the chatbot, have different conversation sessions.

Software Flow Diagram:


Local Machine (16 GB RAM)
Fast API (Python) Back-end
Langchain RAG System
Users personal info
Postgress DB
Chroma DB
Data provided by the Admin for RAG
Users Chat History
MongoDB
Llama 3.1
Front-end
Admin Panel
Chatbot UI
Designed by Saif.U