Phase 2 - Chatbot with Rule-Based + LLM Integration

1. Introduction

In Phase 2, we integrated the frontend and backend of the chatbot system. The backend is powered by FastAPI and Groq's LLM API, while the frontend is a custom-built HTML/CSS/JS interface. This phase allows the chatbot to handle basic rule-based responses (for greetings, farewells, thank you, help, etc.) and more advanced AI-driven responses through Groq's LLM.

2. Workflow Explanation

The chatbot workflow in Phase 2 can be summarized as follows:
1. User types a message in the frontend (HTML/JS interface).
2. The message is sent to the FastAPI backend (/chat endpoint).
3. The backend first checks if the message matches predefined rule-based responses.
4. If no rule matches, the message is forwarded to Groq LLM API for an AI-generated response.
5. The backend sends the final response (rule-based or LLM-generated) back to the frontend.
6. The frontend displays the chatbot's reply.

3. Technologies Used

- **Python (FastAPI):** Backend framework for handling requests.
- **Groq LLM API:** Provides AI-based responses when rule-based logic fails.
- **HTML/CSS/JavaScript:** Used to build the frontend UI for interaction.
- **Fetch API:** Sends user queries from frontend to backend.
- **Rule-Based Logic:** Provides quick responses for predefined queries.

4. Example Input/Output

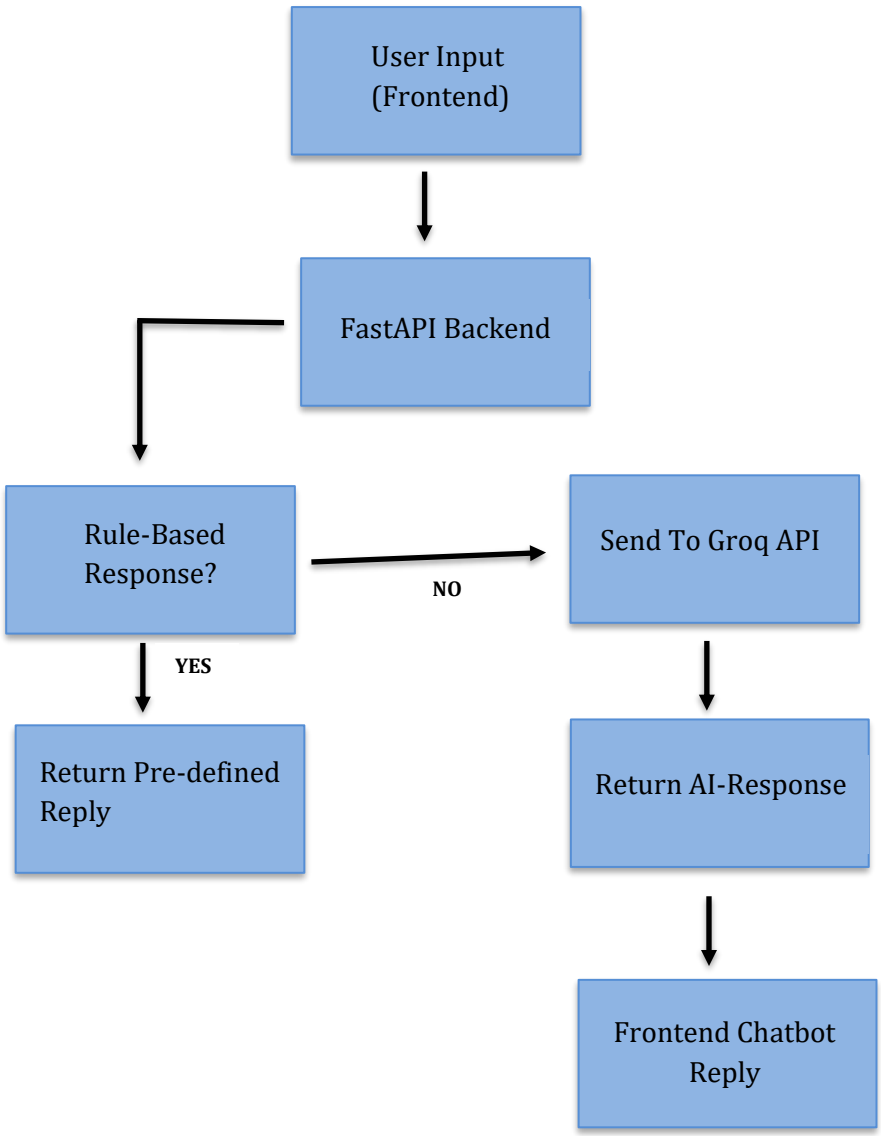
Example 1 (Rule-Based):

User: Hello
Chatbot: Hi there! How can I help you today?

Example 2 (AI-Powered): User: Explain quantum computing.
Chatbot: Quantum computing uses quantum mechanics principles...

5. System Flow Diagram

The flow diagram below shows how the chatbot processes user queries in Phase 2.



6. Final Outcome

By the end of Phase 2, the chatbot successfully integrates both rule-based and AI-driven responses. The frontend and backend communicate seamlessly, allowing users to interact with a functional chatbot that can handle simple queries instantly while relying on Groq's LLM for complex questions.