

Requirement Document: Generative AI Chatbot for Customer Support

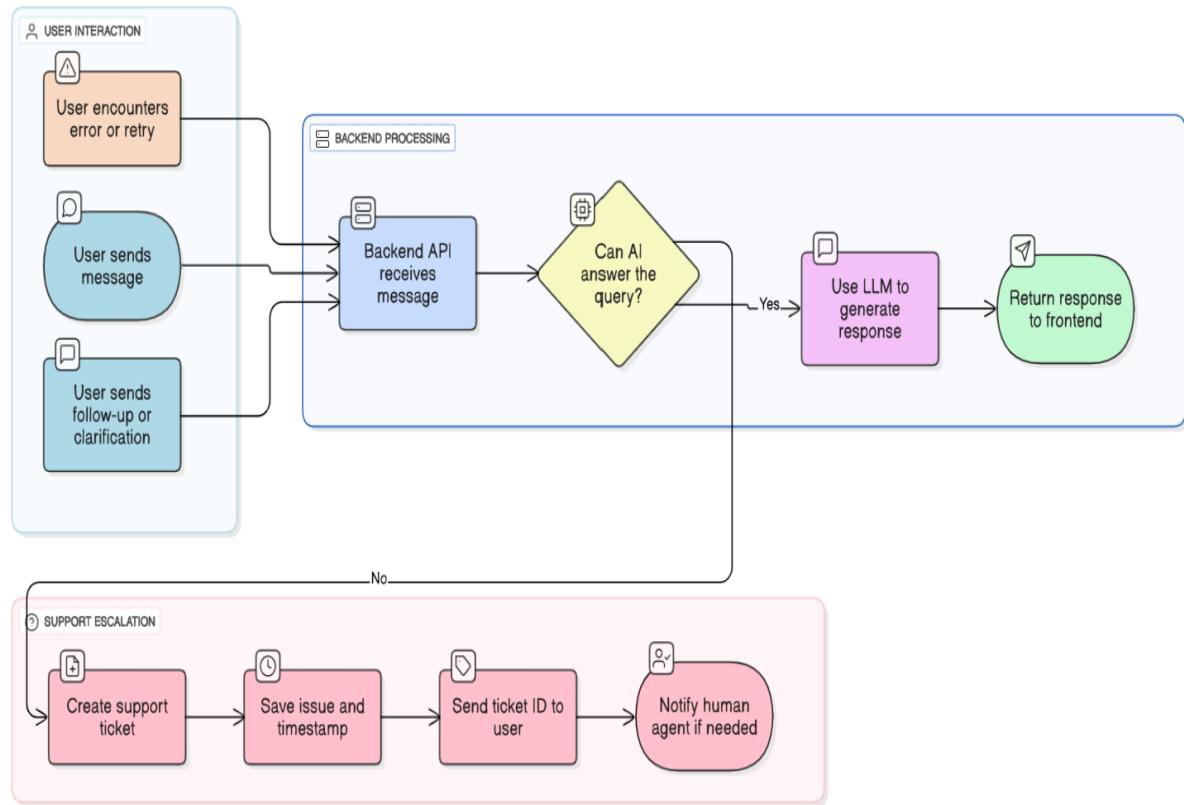
1. Objective

Build a basic chatbot using a Generative AI model (e.g., OpenAI GPT-4) that can converse naturally with users. The chatbot will be deployed as a web application with a REST API backend.

The chatbot will serve as a first-line assistant for customer support, capable of responding to frequently asked questions, assisting users in generating support tickets, and escalating to human agents when necessary. The system is aimed at reducing the load on human agents by handling simple queries and ensuring seamless ticket creation for complex issues. Key features include:

- FAQ Handling: Respond to frequent user questions (e.g., shipping, refund, return policy).
- Ticket Creation: If the chatbot cannot resolve a user's issue, it will create a support ticket containing the user's message and a timestamp.
- Ticket Acknowledgment: The chatbot will confirm ticket creation and share a ticket ID with the user.
- Human Escalation: If needed, the chatbot can inform the user that their issue will be reviewed by a human support agent.

2. High Level Flow of the Chatbot



3. Functional Requirements

1. Use Case 1: Ask Information About a Support Ticket

Input: Ticket ID from user

Output: Return status message (stubbed/static for MVP)

2. Use Case 2: Create a New Support Ticket

Triggered when AI cannot resolve issue

Generates ticket ID, stores issue message, and returns ID to user

3. Use Case 3: View Chat History

For MVP, basic session-level chat log shown (not persisted across refresh)

In the Minimum Viable Product (MVP) version:

- The chatbot will maintain the chat messages only for the current session.

- The conversation is not stored in a database or file, so once the session ends (e.g., user refreshes the page, closes the browser, or times out), the chat history is lost.

4. Sample Chat Conversation

Use Case 1: Asking Information About a Support Ticket

User: Hi, I want to check the status of my support ticket.

Bot: Sure! Please provide your ticket ID.

User: It's TCKT-20250722-008.

Bot: Thank you. Your ticket TCKT-20250722-008 is currently being reviewed by our support team. You'll receive an update shortly.

Use Case 2: Request for Creating a New Support Ticket

User: I'm unable to log into my account.

Bot: I'm sorry to hear that. Would you like me to create a support ticket for you?

User: Yes, please.

Bot: Got it. I've created a support ticket for your issue. Your reference number is **TCKT-20250723-001**. Our support team will contact you soon.

Use Case 3: View Chat History (Session-Based)

User: Hi

Bot: Hello! How can I help you today?

User: I want to return the product.

Bot: Sure, could you please share the product ID or describe the issue?

User: It's defective.

Bot: I've logged a ticket for your return request. Ticket ID: **TCKT-20250723-002**. You'll receive an update soon.

(User scrolls up and sees the full session history above — messages are retained as long as the page remains open.)

5. User Interface

- Input box for user request
- Submit button

- Output box for chat response
- Display chat history

6. Recommended Tech Stack

- Frontend: React.js or Angular or plain HTML/JS or Streamlit
- Backend: Python
- LLM Provider: Groq(llama-3.3-70b-versatile)
- Hosting: Local or EC2 (Backend)
- Backend Storage: JSON files or DB
- Deployment: GitHub + CI/CD – Recommended

7. Demo

 Support Assistant App - [Streamlit](#)