Intern Assignment: AI Voice Agent for Hospital Network

Assignment Overview:

The task is to create a conversational AI voice agent that can answer user queries about a large network of hospitals. The agent will be accessed through a simple web interface. The core challenge involves enabling the agent to accurately retrieve information from a dataset that is assumed to be too large to fit into a standard AI model's context window.

This project is designed to evaluate your skills in integrating third-party APIs, overcoming data retrieval challenges, and developing a functional, end-to-end application.

Data Source: A large CSV file containing a list of several thousand network hospitals and their addresses is attached with this assignment.

Technology Suggestions:

- **Backend:** You may use any technology of your choice. Common options include MERN stack (Node.js/Express) or Python frameworks such as Flask or FastAPI.
- **Voice AI API:** We recommend using free trial credits for audio-to-audio API from services like Gemini, OpenAI, Sarvam or ElevenLabs. You are free to use any other service. You should activate a free trial to get the necessary API keys.

Assignment Details:

This assignment consists of three parts. The final product should be a simple web page with a microphone button to initiate a voice conversation with the "Loop AI agent".

UI Example:
A minimal web page that includes:

- A title such as “Loop AI Hospital Network Assistant”
- A large, centered microphone icon button
- A label under the button that says “Start Conversation”

Part 1: API Integration and Data Loading

- Create a simple interface with a single button to start and stop the voice conversation.

- Integrate your chosen voice-to-voice API to handle user speech input and generate a spoken response.
- Load the provided hospital data file (CSV). Since it is assumed to be very large files, you must devise a strategy to access and search this data efficiently without sending the entire file to the AI model.
- Implement the core logic to answer user queries. Your solution will be tested specifically on its ability to answer:
    1. "Tell me 3 hospitals around Bangalore."
    2. "Can you confirm if Manipal Sarjapur in Bangalore is in my network?."
- **Hint:** To handle the large dataset, we strongly encourage exploring modern techniques like creating an Retrieval-Augmented Generation (RAG) vector database of the hospitals list or extract search keywords from user query
- using function calling/structured outputs for doing an ‘exact match’ search on the hospital database.

Part 2: Introduction and followups

- The voice agent must introduce itself as a “Loop AI" at the beginning of the conversation.
- The agent should be able to handle simple follow-up questions after providing an initial answer.
- If the user query is insufficient, it should ask clarifying questions like “I have found several hospitals with this name, In which city are you looking for Apollo hospital?”

Part 3: Error Handling and connecting with Twilio phone number (optional for Brownie points)

- Implement a mechanism to detect when a user's question is out of scope (i.e., not related to finding hospitals in the provided list).
- When an out-of-scope question is detected, the agent must politely respond with, "I'm sorry, I can't help with that. I am forwarding this to a human agent," and then end the interaction.
- Connect this API with an actual phone number on a Twilio trial account.

Part 1 of this assignment is compulsory.

Submission Requirements:

- **Loom Video Demo:** Submit a link to a short (approx. 1-2 minute) Loom video demonstrating a live conversation with your voice agent. The video must show you asking the two test questions mentioned.

- **Public GitHub Link:** Provide a link to a public GitHub repository containing your source code.

Evaluation:

- **Functionality First**: Your assignment will be primarily evaluated on functionality. The agent must successfully answer the test queries in a voice conversation with minimum latency.
- **Use of AI Tools**: The use of AI tools (e.g., Cursor, ChatGPT, Claude Code) to assist in writing code is encouraged. However, an understanding of your code and workflow APIs you are using is required, as this will be evaluated in the technical interview.
- **No ML Expertise Needed**: You do not need to build or understand the internal workings of AI models. You only need to know how to effectively call and utilize the provided APIs.
- **Code Quality**: Our evaluation focuses on functionality and correct API integration. Clean code is appreciated but not required.

Deadline:

- You are expected to spend **24 hours** to complete this assignment.