Exercise – Conversational AI Back-End Service

Scenario

A healthcare provider is developing a back-end service in Python using FastAPI. This service will expose an endpoint that leverages multi-agent AI frameworks to help patients manage their appointments through a conversational interface.

Your Task

Implement a conversational AI agent that supports the following interaction flow:

1. User Verification: The assistant must first verify the patient's identity using their full name, phone number, and date of birth;
2. List Appointments: This action should be available only after successful user verification;
3. Confirm Appointment: This action should be available only after successful user verification;
4. Cancel Appointment: This action should be available only after successful user verification;
5. Routing/re-routing: The assistant must allow the patient to freely navigate between actions—for example, listing appointments, then confirming one, and returning to the list again. Transitions between actions should feel natural and conversational.

Requirements

- The endpoint must simulate a conversational experience between the patient and the AI assistant;
- Access to appointment-related actions (listing, confirming, canceling) must be strictly gated behind successful identity verification;
- You may use any conversational AI framework you prefer, such as LangChain, LangGraph, or LlamaIndex, to implement the agent logic.

Guidance

Consider the person interacting with the endpoint as a patient of a clinic. The assistant should guide them through the process step by step—starting with verifying their identity—before granting access to appointment management features.