Agentic AI Internship: Evaluation Task

This assignment is designed to evaluate your understanding of Agentic AI, multi-agent system design, and full-stack application development. You are required to develop, host, and present a multi-user AI chatbot using LangChain and LangGraph that simulates patient behaviour. The chatbot must interact naturally with users, describe symptoms, and accept prescribed treatments.

1. Project Description

Develop a multi-user AI chatbot application that simulates a patient interacting with users (acting as doctors). The chatbot should describe its symptoms progressively, answer relevant questions, and eventually accept a prescribed treatment from the user. Your system must be built using LangChain and LangGraph, with Flask as the backend API, and deployed on Vercel. The front-end can be built using React/JavaScript or pure HTML/JavaScript.

2. System Requirements

Your project should include the following components:

• **Frontend**: A visually appealing interface developed in React.js or HTML/JavaScript.
• **Backend**: A Flask-based API that communicates with the LangChain/LangGraph logic.
• **AI Logic**: Use LangChain and LangGraph to create a conversational flow where the chatbot simulates patient behaviour.
• **Hosting**: The entire app should be deployed on Vercel.
• **Multi-user Capability**: Ensure that multiple users can interact with independent chatbot sessions simultaneously.

3. Technical Specifications

- Use Python (Flask or FastAPI) for backend development.
- Use LangChain and LangGraph for chatbot flow, state management, etc.
- Frontend should consume backend APIs for chatbot communication.
- Deploy backend on Vercel or a compatible service with Flask support.
- Maintain session isolation for each user to ensure independent interactions.
- Integrate minimal logging for tracking conversations (without storing sensitive data).

4. Expected Deliverables

1. **Working Application**: Hosted on Vercel and accessible via a public link.
2. **Source Code**: Complete frontend and backend code, uploaded on ICAPP
3. **Presentation Deck**: A short slide deck (max 6 slides) explaining:
   - Architecture
   - Technologies used
   - Key design choices
   - Demonstration of chatbot functionality
4. **Recorded Video Presentation**: 5–7 minutes, on ICAPP, demonstrating:
   - Presentation of your deck, explaining your code and development process
   - Live chatbot demo with multiple users
   - Discussion of challenges and learnings