Documentation
Microservice-based ChatBot Q&A on Medical Services

Adi Prager

April 18, 2025

Overview

An AI-powered, multilingual (Hebrew and English) **stateless microservice chatbot** that collects personal medical information and provides **personalized Q&A** about services offered by Israeli HMOs: **Maccabi**, **Meuhedet**, and **Clalit**. All user session data is managed on the client side (frontend).

Built using **FastAPI**, **Streamlit**, and **Azure OpenAI**, connected to a local database of medical service benefits extracted from HTML files.

Project Structure

```
ChatBot_Q-A_on_Medical_Services/
        backend/
            app/
                data/
                    phase2_data/
                        alternative_services.html
                        communication_clinic_services.html
                        dentel_services.html
                        optometry_services.html
                        pragrency_services.html
                        workshops_services.html
                services/
                    confirm_classifier.py
                    info_collector.py
                    llm_client.py
                    qa_handler.py
                    user_info_extractor.py
                utils/
                    html_loader.py
                    logger.py
                    test_concurrency.py
                    test_html_read.py
                api.py
                main.py
            requirements.txt
        frontend/
            app.py
            requirements.txt
        README.md
        .env
```

How It Works

1. User starts a chat via Streamlit.
2. **Info Collection Phase** (`phase = info_collection`)
   - Bot asks for personal details (name, ID, gender, etc.).
   - Bot prints a clear confirmation summary.
   - If user confirms, switch to QA phase.
3. **Question Answering Phase** (`phase = qa`)
   - Bot answers based on user HMO/tier and knowledge base.
4. **Logging**: All interactions are saved in `chatbot.log`.
5. **Data Source**: Local HTML files parsed with `BeautifulSoup`.

Technologies Used

- **FastAPI** – API server.
- **Streamlit** – Web UI.
- **Azure OpenAI** – LLM completion service.
- **BeautifulSoup** – HTML parsing.
- **httpx / asyncio** – Async concurrency.
- **Pydantic** – Data validation.

Installation

1. Clone the project

```
git clone https://github.com/adidereviani/ChatBot_Q-A_on_Medical_Services.git
cd ChatBot_Q-A_on_Medical_Services
```

2. Install Backend

```
cd backend
pip install -r requirements.txt
```

3. Install Frontend

```
cd ../frontend
pip install -r requirements.txt
```

4. Set Environment Variables

Create a `.env` file inside the `backend/` folder:

```
AZURE_OPENAI_API_KEY=your-key-here
AZURE_OPENAI_API_VERSION=your-api-version
AZURE_OPENAI_API_BASE=https://your-resource-name.openai.azure.com/
AZURE_DEPLOYMENT_NAME=your-deployment-name
```

Running Locally

Start backend server

```
cd backend
uvicorn app.main:app --reload
```

Start frontend (Streamlit app)

```
cd frontend
streamlit run app.py
```

Visit `http://localhost:8501` to interact with the chatbot.

Language Support

- English
- Hebrew

Features

- Multilingual conversation (Hebrew / English).
- Phase switching: `info_collection` → `qa`.
- Integration with Azure OpenAI GPT.
- Auto-recovery if user info is missing or incomplete.
- Fully logged conversation history.
- Local database parsed from HTML files.

Author

Developed by Adi Prager.