****

**PROGRAMMING FOR AI (LAB)**

**Lab Task No 12**

**Submitted To :- Sir Rasikh Ali**

**Submitted By :- Aiman Ijaz**

**Roll No :- SU92-BSAIM-F23-010**

**Department :- Software Engineering**

**Section :- BSAI-4A**

**Lab 12 task : Run the "HadithBot.ipynb" file and similarly implement any QnA bot which you made from "Lab 10"**

**Medical Center QnA Chatbot**

This project is a Question-and-Answer (QnA) chatbot for a medical center. It uses Flask for the web interface, FAISS for similarity search, and Sentence Transformers for generating embeddings.

**Project Overview**

The chatbot answers common patient queries such as:

* Booking appointments
* Business hours
* Specialist availability
* Emergency procedures
* Clinic address
* Test pricing

It uses a semantic search approach to find the most relevant response based on the user's input.

**Folder Structure**

Qna/

│

├── app.py # Flask web server

├── medical\_data.py # Contains questions and answers

├── faiss\_index.py # Embedding and FAISS indexing logic

├── requirements.txt # Dependencies

├── templates/

│ └── index.html # HTML frontend for the chatbot

└── static/

└── style.css (optional) # Optional styling

**How It Works**

* The chatbot stores a predefined list of questions and answers in medical\_data.py.
* These questions are embedded into vectors using a pre-trained sentence transformer model.
* FAISS creates an index of these vectors for fast similarity search.
* When the user enters a query:
  + The query is converted to a vector.
  + FAISS searches for the most similar stored question.
  + The answer corresponding to the closest match is returned if the similarity is above a certain threshold.

**Key Files**

* app.py: Handles the web interface using Flask.
* medical\_data.py: Contains hardcoded question-answer pairs.
* faiss\_index.py: Handles vectorization and similarity search.
* index.html: Provides a simple web-based UI for interacting with the chatbot.

**Example Questions**

Here are some sample queries the chatbot can respond to:

* I want an appointment tomorrow
* What are your business hours?
* Do you have specialists?
* Where are you located?
* How much do tests cost?
* What should I do in an emergency?

**Dependencies**

The project uses the following libraries:

* Flask
* faiss-cpu
* sentence-transformers

These are listed in the requirements.txt file.

**Customization**

* To add more responses, edit or expand the list in medical\_data.py.
* The sensitivity of the response can be adjusted in faiss\_index.py by changing the FAISS distance threshold.
* You can customize the frontend in index.html and add styling via CSS.