# 📘 Project Documentation: MaliziaSign Chatbot with RAG (Retrieval-Augmented Generation)

## 🔍 Overview

This project is a real-time chatbot web application built using Streamlit, designed to assist users by answering queries using a dataset of instructions and responses. The chatbot utilizes a Retrieval-Augmented Generation (RAG) approach with sentence embeddings for semantic search and Flan-T5 for natural language generation.

## 🚀 Features

• Retrieval-based chatbot with contextual awareness.

• Utilizes SentenceTransformer for semantic search (top-k similar questions).

• Flan-T5-small model for fast and instruction-tuned responses.

• Clickable suggested questions for enhanced UX.

• Maintains chat history and dynamically updates suggestions.

## 🧰 Technologies & Libraries Used

|  |  |
| --- | --- |
| Tool | Purpose |
| Streamlit | Web app framework |
| Transformers (Hugging Face) | Loading the Flan-T5 model |
| SentenceTransformer | Semantic similarity search using sentence embeddings |
| PyTorch | Underlying tensor computations |
| Pandas | Data handling (CSV loading) |
| NumPy | Utility operations |
| random | Suggestion sampling |

## 🗃️ Dataset

File: malizia\_sign\_complete\_dataset.csv

Columns:

• Instruction: Sample user instructions/questions.

• Response: Corresponding ideal chatbot answers.

## 🧠 Model & Embedding Details

### ✅ Flan-T5-small

• Pre-trained model from Google.  
• Efficient for instruction-following tasks.  
• Used to generate responses based on a prompt with retrieved context.

### ✅ SentenceTransformer (all-MiniLM-L6-v2)

• Lightweight embedding model for fast similarity search.  
• Encodes each instruction into a high-dimensional embedding for cosine similarity matching.

## 🔁 Workflow

### 1. Data and Model Loading

Cached function `load\_data\_and\_models`:  
• Loads CSV.  
• Tokenizes using Flan-T5.  
• Embeds corpus with SentenceTransformer.  
• Computes and caches embeddings for quick lookup.

### 2. User Query Handling (RAG)

Function: get\_rag\_response(user\_query, ...)  
• Encodes the user query into embedding.  
• Computes cosine similarity with precomputed corpus embeddings.  
• Retrieves top-k most relevant instructions & associated responses.  
• Constructs a prompt with retrieved responses.  
• Generates a response using Flan-T5.

### 3. Streamlit Interface • Chat history is maintained across messages. • Random suggested questions shown as buttons. • Users can type in custom questions. • Option to refresh suggestions.

## 💬 Example Prompt to Flan-T5

chat with context: [Top 3 similar responses]  
Question: How do I register my business?

## 🖥️ Usage Instructions

1. Place malizia\_sign\_complete\_dataset.csv in the working directory.

2. Install required packages:

pip install streamlit transformers sentence-transformers pandas torch

3. Run the app:

streamlit run app.py

## ⚙️ Possible Improvements

• Use larger language models (e.g., Flan-T5-base) for improved accuracy.

• Integrate feedback-based ranking to improve relevance.

• Store chat logs for analytics.

• Add voice input/output for accessibility.