Project Report: AI-Assisted Customer Support System

1. Project Overview

This project implements an intelligent Retrieval-Augmented Generation (RAG) system designed to assist customer support agents. By combining a specialized support dataset with Google’s Gemini LLM, the system retrieves relevant historical context to generate accurate, professional, and editable customer responses.

2. Technical Architecture

The system is divided into three functional layers: data management, backend logic, and the user interface.

A. Dataset & Knowledge Base

- **Data Source:** Uses a structured CSV file (Customer_Support_Training_Dataset.csv).
- **Data Schema:** The dataset includes critical fields such as instruction (the customer's request), intent (the categorized goal), and response (the historical correct answer).

B. Backend Utility Logic (helper.py)

This module serves as the engine for the RAG pipeline:

- **Embedding Engine:** Uses the models/text-embedding-004 model via the Gemini API to transform text into 768-dimensional vectors.
- **Vector Indexing:** Implements FAISS (L2 distance) to store and search these embeddings efficiently.
- **Contextual Retrieval:** The semantic_similarity function converts user queries into vectors and performs a top-k search against the local index.
- **LLM Integration:** The call_llm function utilizes Gemini models to synthesize a final response by injecting the retrieved historical data into a structured system prompt.

C. Frontend Application (demo.py)

- **Interactive Interface:** Built with Streamlit to allow agents to input queries and view AI-generated drafts.
- **Dynamic Refinement:** Enables users to iterate on the AI's response by providing specific feedback (e.g., "be more empathetic"), which triggers a localized regeneration loop.

3. Technology Stack

Category	**Component**
LLM & Embeddings	Google Gemini (models/gemini-1.5-flash, text-embedding-004)
Vector Search	FAISS (Facebook AI Similarity Search)
Frameworks	LangChain, Streamlit
Data Handling	Pandas, Numpy

4. Key Features

- **High Precision:** By using text-embedding-004, the system ensures high semantic accuracy during retrieval.
- **Free-Tier Optimization:** The system is configured to support Gemini's free-tier models, making it cost-effective for development and small-scale deployment.
- **Grounding:** AI responses are strictly grounded in the provided Customer_Support_Training_Dataset, reducing the risk of irrelevant information.

5. Conclusion

The AI-Assisted Customer Support System successfully bridges the gap between static historical data and dynamic AI generation. It provides a scalable solution for improving support response times while maintaining high consistency through retrieval-based grounding.