**Mental Health Support Chatbot: Final Report**

**Author**: Nazokat Gayimova & Farkhodjon Nematov

**Course**: Machine Learning (Group Project)

**Institution**: IT Park University

**Date**: July 2025

**1. Introduction**

In today’s fast-paced digital world, mental health has become an increasingly critical concern. The prevalence of anxiety, depression, and emotional burnout has made it essential to explore innovative, scalable support systems. Traditional mental health services often struggle to meet the rising demand, especially in remote or underserved areas. This project addresses that gap by building a simple yet effective retrieval-based mental health chatbot capable of providing supportive, empathetic, and context-aware responses to users based on real conversations.

This report presents a comprehensive overview of the chatbot system developed as part of the Master’s program capstone project. The chatbot uses a retrieval mechanism built on top of pretrained sentence embeddings and is deployed using Flask and Streamlit. The primary goal was to enable real-time interaction where users could share their thoughts or emotions and receive meaningful replies.

**2. Background and Motivation**

**2.1 The Role of Chatbots in Mental Health**

The adoption of AI chatbots in mental health is no longer a futuristic concept. From providing general support to suicide prevention, bots have been leveraged to offer 24/7 accessibility and anonymity. However, most chatbots fall short in two major areas:

* Lack of empathetic responses
* Generic, non-contextual answers

By utilizing a retrieval-based model trained on actual empathetic dialogues, we aimed to overcome these limitations.

**2.2 Dataset Used**

The project used the **EmpatheticDialogues** dataset created by Facebook AI. This dataset consists of over 25,000 short conversations initiated by a prompt (“situation”) and followed by human responses labeled by emotion. Key characteristics of the dataset include:

* Emotion-labeled dialogue
* Real user-generated conversations
* 32 predefined emotional categories

This dataset provided the ideal base to train a chatbot capable of understanding both context and emotional tone.

**3. Methodology**

**3.1 Data Cleaning and Preprocessing**

The original dataset had multiple columns and occasionally inconsistent entries. We applied the following preprocessing steps:

* Selected relevant columns: Situation, empathetic\_dialogues
* Removed empty or null values
* Eliminated entries with only whitespace
* Merged the conversational pairs into single coherent responses

This cleaning ensured that the final data used for vectorization was consistent and usable.

**3.2 Embedding Generation**

We used the pretrained all-MiniLM-L6-v2 model from **SentenceTransformers** to convert all “Situation” texts into dense vector embeddings. This model is known for a good trade-off between speed and accuracy.

**Steps:**

1. Load cleaned data
2. Encode each “Situation” using the model
3. Save resulting vectors as situation\_embeddings.npy or precomputed\_embeddings.pkl

This allows fast similarity search without repeated inference.

**3.3 Retrieval Logic**

The chatbot matches a user’s input to the most similar “Situation” in the dataset and returns the corresponding empathetic response. The retrieval system works as follows:

* Convert user input into an embedding
* Use cosine similarity to find the closest match
* Return the corresponding Agent response from the dataset

**3.4 Fallback Strategy**

In cases where the similarity score is too low or if the response is missing, we implemented a fallback mechanism using simple keyword rules:

* If the message contains words like *happy, excited*, reply with a congratulatory or joyful tone.
* If it contains *sad, depressed, lost*, reply with comforting words.

This ensured that the chatbot remains responsive and logical even for out-of-distribution queries.

**4. System Architecture**

The architecture consists of two main components:

**4.1 Flask API**

The Flask server exposes two endpoints:

* /load\_model: Initializes and loads embeddings
* /predict: Accepts POST requests with user input and returns a suitable response

**4.2 Streamlit UI**

The frontend is built using Streamlit. It mimics a real chat interface where users can type a message and view the chatbot’s reply. Features include:

* Conversational flow display
* Real-time response
* Easy-to-use UI for non-technical users

**4.3 Folder Structure**

mental-health-chatbot-api/

├── api\_app.py

├── streamlit\_app.py

├── retrieval.py

├── data.csv

├── data\_cleaned.csv

├── generate\_embeddings.py

├── precomputed\_embeddings.pkl

├── requirements.txt

└── venv/

**5. Results and Evaluation**

**5.1 Functionality**

The chatbot performs well across common mental health-related prompts. It is capable of understanding a range of emotions and matching appropriate replies from the dataset.

**Example interactions:**

* User: “I feel sad today”

Bot: “I’m sorry you’re feeling that way. You’re not alone — would you like to talk about it?”

* User: “My dog gave birth”

Bot: “That sounds like a special moment! Pets bring us so much joy.”

* User: “I’m scared for tomorrow”

Bot: “It’s okay to feel nervous. Is there anything I can help you with?”

**5.2 Strengths**

* Real, empathetic replies based on dataset
* Fast and responsive (precomputed embeddings)
* Handles unknown questions with fallback logic
* Easy to run with clear instructions

**5.3 Limitations**

* Responses are limited to dataset coverage
* No multi-turn conversation tracking
* No personalization (responses are generic for all users)

**6. Challenges Faced**

* **Data Formatting**: The dataset had formatting issues that required cleaning before processing.
* **Deployment Size**: Some files (like embeddings) were over 90MB, causing GitHub LFS warnings.
* **Model Speed**: Real-time embedding generation was slow, which was resolved using precomputed files.
* **Flask-Streamlit Communication**: Needed proper error handling when the server was not running.

**7. Conclusion**

This project successfully demonstrates how a lightweight, retrieval-based chatbot can be used to support mental health conversation needs. By using real empathetic dialogues and sentence embeddings, the chatbot offers a comforting, human-like interaction that can be extended in future work.

**8. Future Work**

* Integrate an emotion classifier to understand tone more accurately
* Add memory/context support for multi-turn conversations
* Deploy the app publicly on Hugging Face or Render
* Improve UI for mobile compatibility
* Support multiple languages

**9. References**

* Rashkin, H., Smith, E. M., Li, M., & Boureau, Y. L. (2019). Towards Empathetic Open-domain Conversation Models. Facebook AI.
* SentenceTransformers Documentation: https://www.sbert.net/
* Flask Documentation: https://flask.palletsprojects.com/
* Streamlit Documentation: https://docs.streamlit.io/

**Submitted by:**

Nazokat Gayimova & Farkhodjon Nematov

Master’s in Software Engineering

July 2025