Mental Health Chatbot using DialoGPT with Emotion and Keyword Detection

By Sriram B

INTRODUCTION:

Mental health has become one of the most crucial concerns in today's fast-paced and digitally connected world. Many people experience anxiety, stress, depression, or feelings of isolation but may hesitate to reach out for help.To address this, conversational agents can offer non-judgmental, instant, and anonymous support.This project introduces a Mental Health Chatbot that engages users in meaningful conversation, detects their emotions, and offers empathetic responses based on keywords or generative dialogue.The goal is not to replace therapy but to act as a first-level support tool that encourages emotional expression and directs users to helplines when needed.

ABSTRACT:

This project presents a mental health chatbot that provides emotional support through a conversational interface.It utilizes DialoGPT for generating responses, keyword-based suggestions for mental health conditions, and an emotion classifier to understand the user's emotional state.The chatbot features a Streamlit frontend and a Flask backend, offering a clean and interactive UI for natural human-like interactions.

TECNOLOGIES USED:

- Python – core programming language
- Flask – lightweight API backend
- Streamlit – modern web UI for chat interface
- Hugging Face Transformers – DialoGPT and Emotion Classifier

- Pandas & Altair – data handling and visualization
- CSV/JSON – for keyword lookup and chat history

STEPS INVOLVED IN THE PROJECT:

The project was developed through the following systematic steps:

1. Problem Understanding and Goal Setting

- Identified the need for an empathetic chatbot to assist users with emotional distress.
- Decided to combine generative AI with emotion detection and keyword-based help.

2. Dataset Preparation

Collected two CSV files:

- keyword_responses.csv – maps mental health keywords to supportive messages and helplines.
- offensive_words.csv – contains sensitive or concerning terms used for monitoring intent.

3. Model Selection and Integration

DialoGPT: Chosen as the main conversational model for generating responses when no keyword matches.

Emotion Classifier: Integrated distilbert-base-uncased-emotion from Hugging Face to detect emotional tone in the user's message.

4. Backend Development (Flask)

Built an API with endpoints:

/chat: Handles incoming user messages and returns bot responses and emotion scores.

/reset: Clears session history.

Implemented logic to:

- First check for keyword match.
- Else generate reply using DialoGPT.
- Run emotion detection on user input.

5. Frontend Development (Streamlit)

Designed an interactive chat UI using Streamlit with:

- Chat bubbles for user and bot messages
- Typing animation
- Input box and quick-reply chips (suggestions)

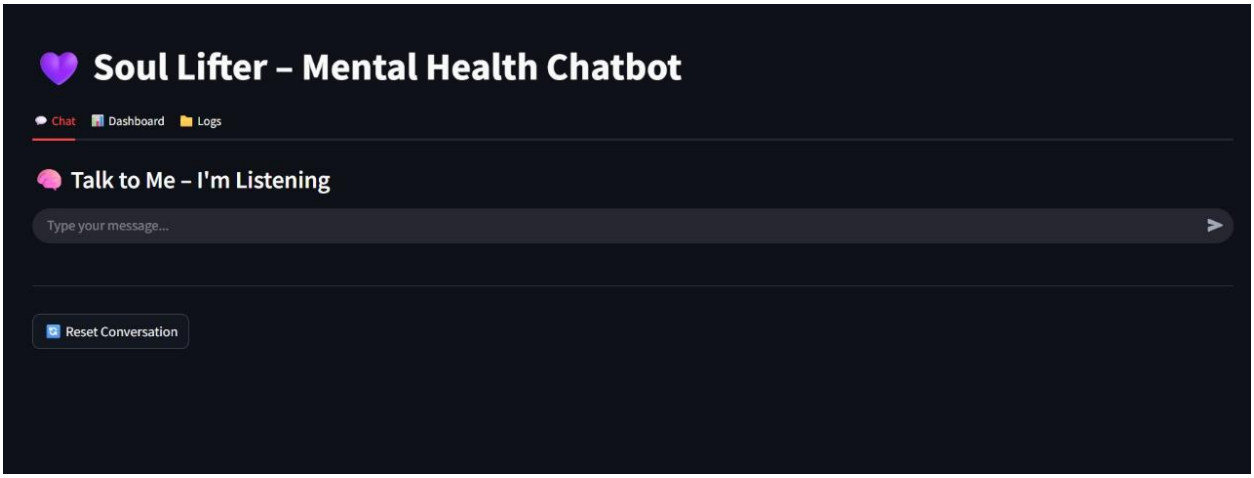
6. Testing and Debugging

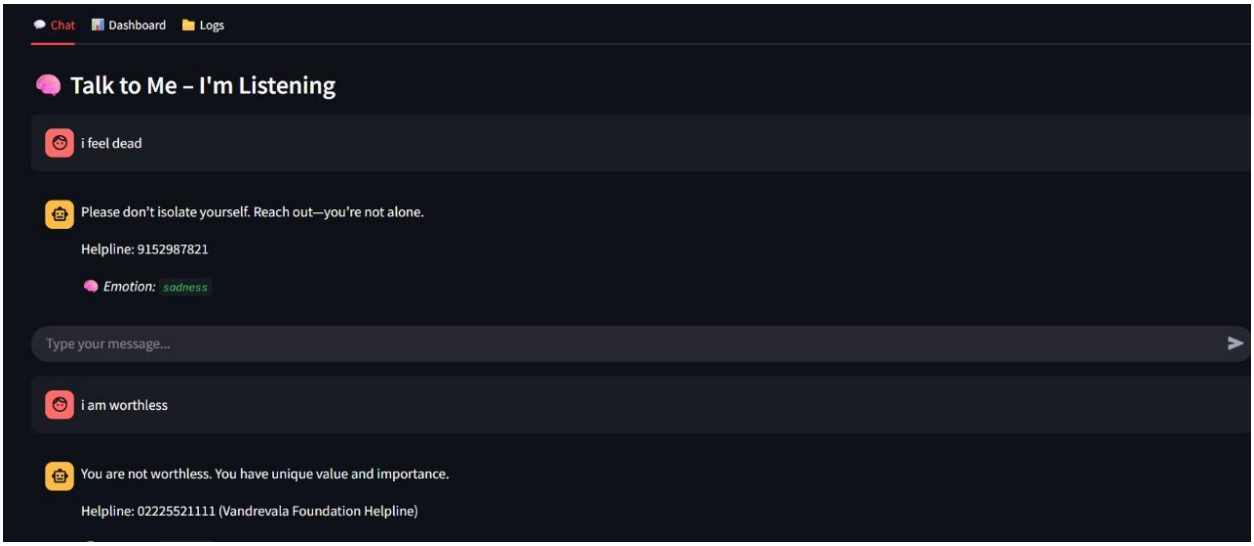
- Verified full chat loop: user input → backend processing → bot reply → UI update.
- Tested emotion classifier across different moods.
- Validated keyword accuracy and fallback model responses.

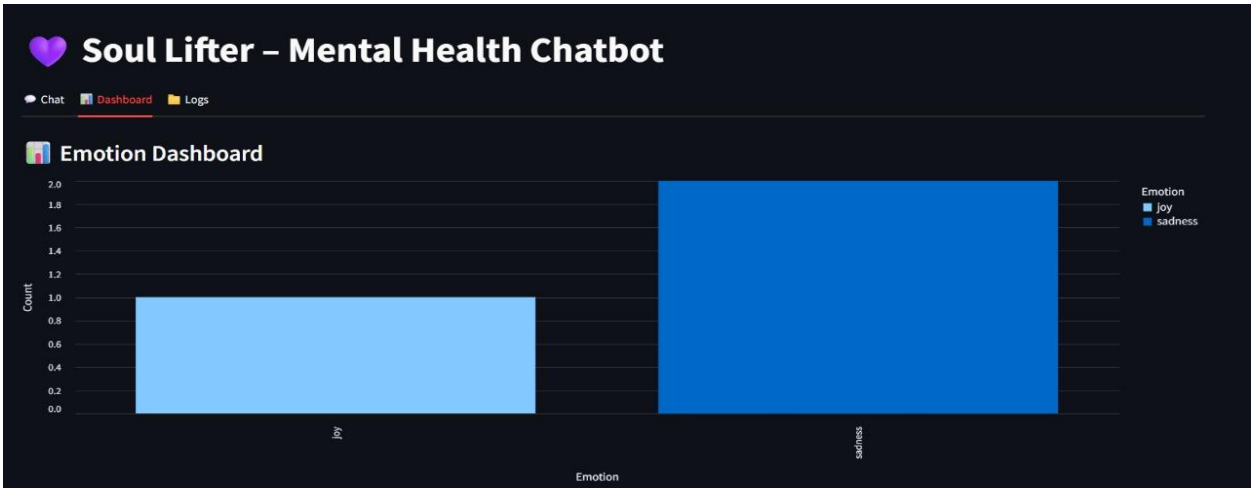
7. Deployment and Output Logging

- Ensured Streamlit runs on localhost:8501 and Flask on localhost:5000.
- Enabled JSON logging of full conversations.
- Provided the ability to save, reset, or analyze chats through the dashboard.

SCREENSHOTS:







CONCLUSION:

The chatbot acts as a helpful and empathetic companion for users experiencing emotional distress or needing someone to talk to. While it is not a replacement for professional mental health care, it can serve as a supportive first step in opening up. The use of DialoGPT and emotion classification enhances its ability to respond naturally and sensitively.

Future enhancements could include:

- Adding multilingual support
- Enabling voice-based input/output
- Connecting users to real-time counselors or live support