PROGRAM

from transformers import pipeline

from textblob import TextBlob

emotion\_classifier = pipeline("text-classification", model="j-

hartmann/emotion-english-distilroberta-base", top\_k=None)

sentiment\_analyzer = pipeline("sentiment-analysis")

def detect\_frustration(text):

sentiment = sentiment\_analyzer(text)[0]

emotion\_results = emotion\_classifier(text)[0]

blob = TextBlob(text)

polarity = blob.sentiment.polarity

implicit\_frustration = any(word in text.lower() for word in ["hope",

"expected", "better", "disappointed"])

emotions = {e['label']: e['score'] for e in emotion\_results}

primary\_emotion = max(emotions, key=emotions.get)

frustration\_detected = (sentiment['label'] == 'NEGATIVE' or

implicit\_frustration or primary\_emotion in ['anger', 'disgust',

'sadness'])

return frustration\_detected, primary\_emotion

def generate\_response(user\_input):

frustration, emotion = detect\_frustration(user\_input)

if frustration:

return f"I'm really sorry to hear that. It sounds like you're feeling

{emotion}. Let me do my best to help fix this."

else:

return "Thanks for reaching out! How can I assist you further?"

if \_name\_ == "\_main\_":

print("Chatbot: Hi! How can I help you today?")

while True:

user\_input = input("You: ")

if user\_input.lower() in ["exit", "quit"]:

print("Chatbot: Thank you for chatting with us!")

break

response = generate\_response(user\_input)

print("Chatbot:", response)

**OUTPUT**

Chatbot: Hi! How can I help you today?

You: I was hoping for something better.

Chatbot: I'm really sorry to hear that. It sounds like you're feeling sadness. Let me do my best to help fix this.

You: Thanks, you were helpful.

Chatbot: Thanks for reaching out! How can I assist you further?

You: I'm disappointed in your service.

Chatbot: I'm really sorry to hear that. It sounds like you're feeling anger. Let me do my best to help fix this.

You: quit

Chatbot: Thank you for chatting with us!