Mood-Based Mental Health App

# 1. Project Title: Mood-Based Mental Health App

# 2. Team Members:

Aastha Adhikari – 21BCS5947

Diksha Khullar – 21BCS5962

# 3. Domain:

Human-Computer Interaction (HCI)

# 4. Tools & Technologies Used:

Python, Streamlit, HuggingFace Transformers, VS Code

# 5. Project Aim:

Create an AI-driven system that personalizes user interfaces based on individual preferences, behavior, or context.

# 6. Overview:

This mental health application is a demonstration of intelligent, emotion-aware interaction. The user expresses how they feel in natural language, and the system automatically detects their mood using a pre-trained **DistilRoBERTa** emotion classification model. Based on the detected mood, the app dynamically updates its interface—changing background colors, tones, and even suggesting activities tailored to the user's emotional state.

It is built using **Streamlit** for live, web-based interactions, making the app fast, clean, and simple to use. The model runs in real-time, offering a seamless user experience and emotional support.

# 7. Key Features:

* **Live Emotion Detection:** Real-time analysis of the user's input using NLP techniques
* **Mood-Based UI Adaptation:** The interface visually shifts to reflect the user's emotional state
* **Activity Recommendations:** Smart suggestions to help improve or stabilize the user's mood
* **Transformer-Based AI Model:** Advanced emotion recognition with high accuracy
* **Streamlit Integration:** Real-time frontend updates without reloads

# 8. Model Used:

We used a transformer model called: **DistilRoBERTa-based Emotion Classifier Model ID:** j-hartmann/emotion-english-distilroberta-base

It is a distilled version of the RoBERTa model fine-tuned specifically on English emotion datasets. This model recognizes emotions such as:

* Joy
* Sadness
* Anger
* Fear
* Surprise
* Disgust
* Neutral

# 9. Working / Explanation:

This app was designed to simulate an emotionally intelligent interface. It begins by asking the user to type in how they are feeling—any phrase or sentence. This message is sent to a pre-trained sentiment analysis pipeline that predicts the most likely emotion.

For example, if the user writes:  
"I'm feeling so stressed and confused",  
the model might classify it under fear or sadness depending on the wording.

Based on this detected emotion:

The UI theme color changes (e.g., blue for sadness, yellow for joy).

A message is displayed offering encouragement or empathy.

A list of mood-specific activities is suggested, such as breathing exercises, journaling, or music links.

This system is reactive and empathetic—two qualities central to Human-Computer Interaction principles.

# 

# 11. Code:

# app.py

import streamlit as st

st.set\_page\_config(layout="wide", page\_title="Mood Adaptive UI")

from transformers import pipeline

# Load sentiment/emotion classifier

@st.cache\_resource

def load\_classifier():

    return pipeline("text-classification", model="j-hartmann/emotion-english-distilroberta-base", return\_all\_scores=False)

classifier = load\_classifier()

# Mood to theme settings

mood\_themes = {

    "joy": {"bg": "#FFF9C4", "font\_color": "#333", "message": "You're glowing! Keep shining 🌞"},

    "sadness": {"bg": "#BBDEFB", "font\_color": "#000", "message": "It's okay to feel down. Try journaling or take a deep breath 💙"},

    "anger": {"bg": "#FFCDD2", "font\_color": "#000", "message": "Let’s cool off with a relaxing activity 🧘‍♂️"},

    "fear": {"bg": "#D1C4E9", "font\_color": "#000", "message": "You're safe here. Maybe some calming music? 🎵"},

    "surprise": {"bg": "#F0F4C3", "font\_color": "#000", "message": "Whoa! Let’s turn this surprise into excitement! 🎉"},

    "neutral": {"bg": "#ECEFF1", "font\_color": "#000", "message": "Steady and balanced – a great place to be 🤍"},

}

st.title("🧠 Mood-Based Mental Health App")

st.markdown("Enter your current mood in words or describe your feelings:")

user\_input = st.text\_area("How are you feeling today?", placeholder="Write anything...")

if user\_input:

        # Get all scores

    raw\_scores = classifier(user\_input, return\_all\_scores=True)[0]

    scores\_dict = {item['label'].lower(): item['score'] for item in raw\_scores}

    # Custom override logic

        # Check for override based on keywords

    lowered\_input = user\_input.lower()

    if any(phrase in lowered\_input for phrase in ["not feeling good", "not okay", "feeling bad", "can't", "unable", "hopeless", "sad", "tired", "late", "down"]):

        mood = "sadness"

    elif scores\_dict.get("sadness", 0) > 0.5:

        mood = "sadness"

    elif scores\_dict.get("anger", 0) > 0.5:

        mood = "anger"

    elif scores\_dict.get("fear", 0) > 0.5:

        mood = "fear"

    else:

        mood = max(scores\_dict, key=scores\_dict.get)

    st.write("🔍 Raw scores:", scores\_dict)  # You can remove this after testing

    # Get theme details

    theme = mood\_themes.get(mood, mood\_themes["neutral"])

    # Apply background color and font dynamically

    st.markdown(

        f"""

        <style>

        .stApp {{

            background-color: {theme['bg']};

            color: {theme['font\_color']};

        }}

        .content-box {{

            background: white;

            padding: 1.5rem;

            border-radius: 12px;

            box-shadow: 0 4px 10px rgba(0,0,0,0.1);

        }}

        </style>

        """,

        unsafe\_allow\_html=True

    )

    st.markdown(f"<div class='content-box'><h3>🧘 Mood Detected: {mood.title()}</h3><p>{theme['message']}</p></div>", unsafe\_allow\_html=True)

    # Activity Suggestion

    st.markdown("### 🌟 Suggested Activity:")

    if mood in ["sadness", "anger", "fear"]:

        st.write("- Try a breathing exercise [Box Breathing (4-4-4-4)](https://youtu.be/FJJazKtH\_9I)")

        st.write("- Journal your thoughts or listen to [calming music](https://www.youtube.com/watch?v=2OEL4P1Rz04)")

    elif mood == "joy":

        st.write("- Share your happiness with someone 😊")

    elif mood == "surprise":

        st.write("- Reflect: What amazed you today?")

    else:

        st.write("- Continue with your daily focus tasks 📋")

else:

    st.info("Write something above to detect mood and personalize your experience.")

# 12. Output Example:

**Sample Input:**

"I am feeling very sad"

**Model Output:**

Sadness: 98.99%

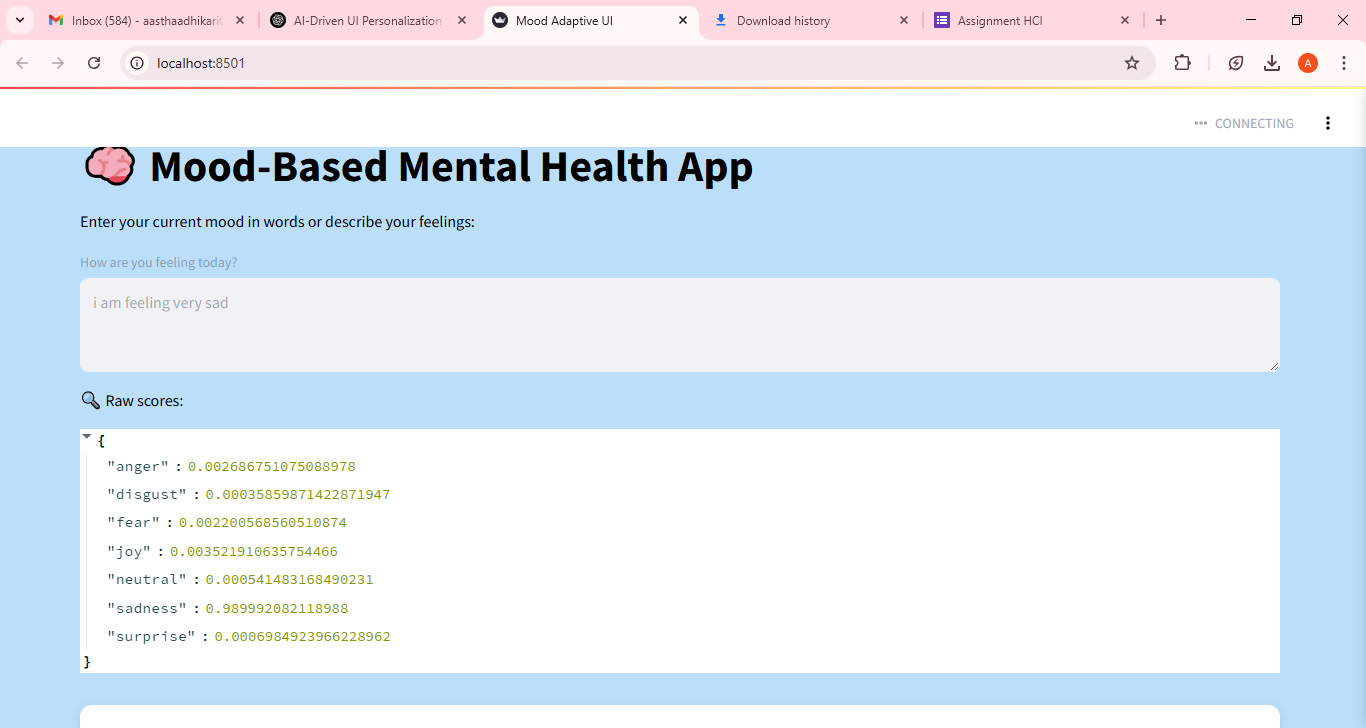
**UI Behavior:**

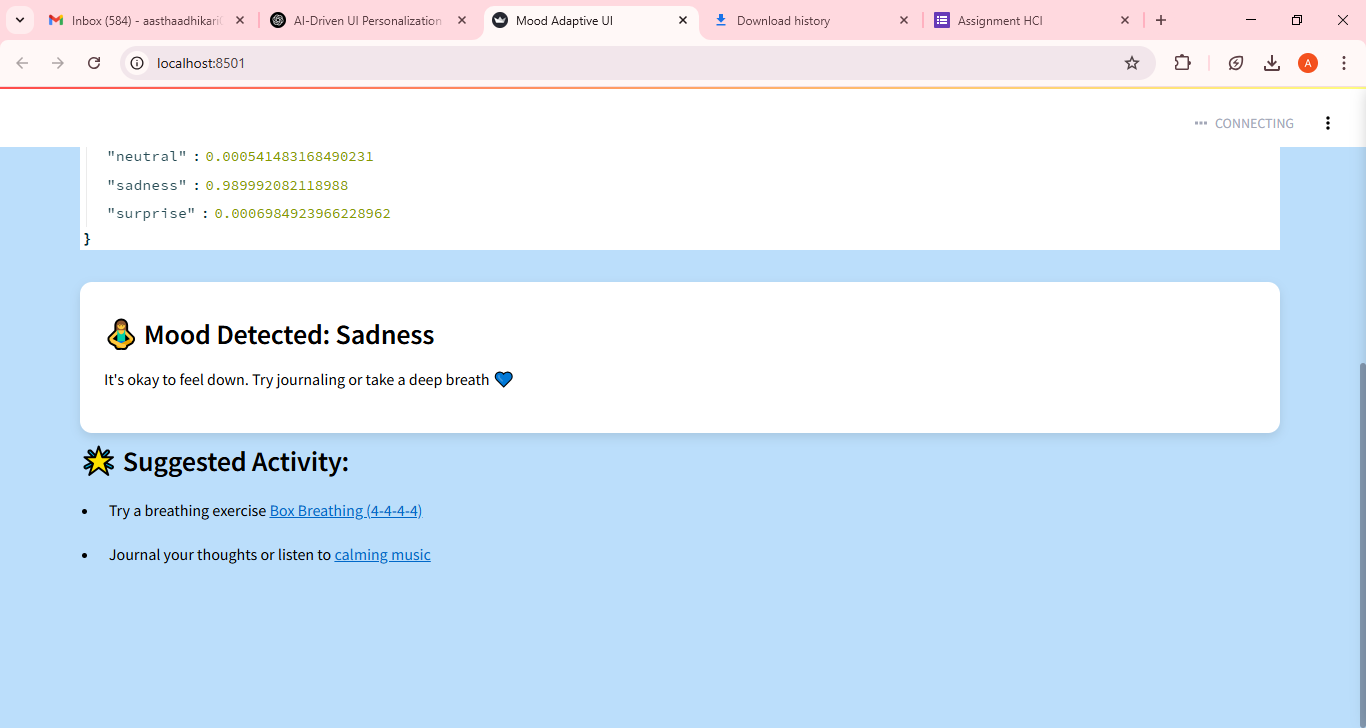
* Background turned **blue**
* Message: *“It’s okay to feel down. Try journaling or take a deep breath 💙”*
* Suggested activities:
  + Box breathing link
  + Calming music playlist
  + Journaling suggestion

**Screenshot:**

* To run the code in the terminal use this command: streamlit run app.py

# 





# 12. Limitations:

* Misclassification on subtle or negative emotional cues:

For vague or sarcastic inputs, the model may misidentify emotions.

Input: "I am not feeling good"  
Output: Joy (Incorrect – Limitation of the model)

* Limited to predefined emotion labels
* No Multi-Modal Inputs:

The app only uses text. Emotions expressed through voice tone or facial expressions are ignored.

* Context Sensitivity:

The model lacks deep context awareness, which may lead to incorrect suggestions.

* Internet Dependency:

Uses an online HuggingFace model unless downloaded locally*.*

# 13. Future Scope:

* **Multimodal Emotion Detection:** Integrating voice and facial analysis along with text.
* **Custom Fine-Tuning:** Training the model on more personalized or diverse emotional data.
* **Mental Health Tracking:** Let users track emotions over days to identify trends and offer reports.
* **Adaptive Recommendations:** More complex behavior-based suggestions based on mood history.

# 14. Conclusion:

This project demonstrates how Human-Computer Interaction can be made more empathetic through AI. By dynamically adapting to the user’s emotional state, we make technology feel more human, supportive, and considerate. It’s a small but meaningful step toward emotionally aware systems that improve mental well-being.

# 15. References or Dataset Links:

HuggingFace model: https://huggingface.co/j-hartmann/emotion-english-distilroberta-base

Streamlit: <https://streamlit.io/>