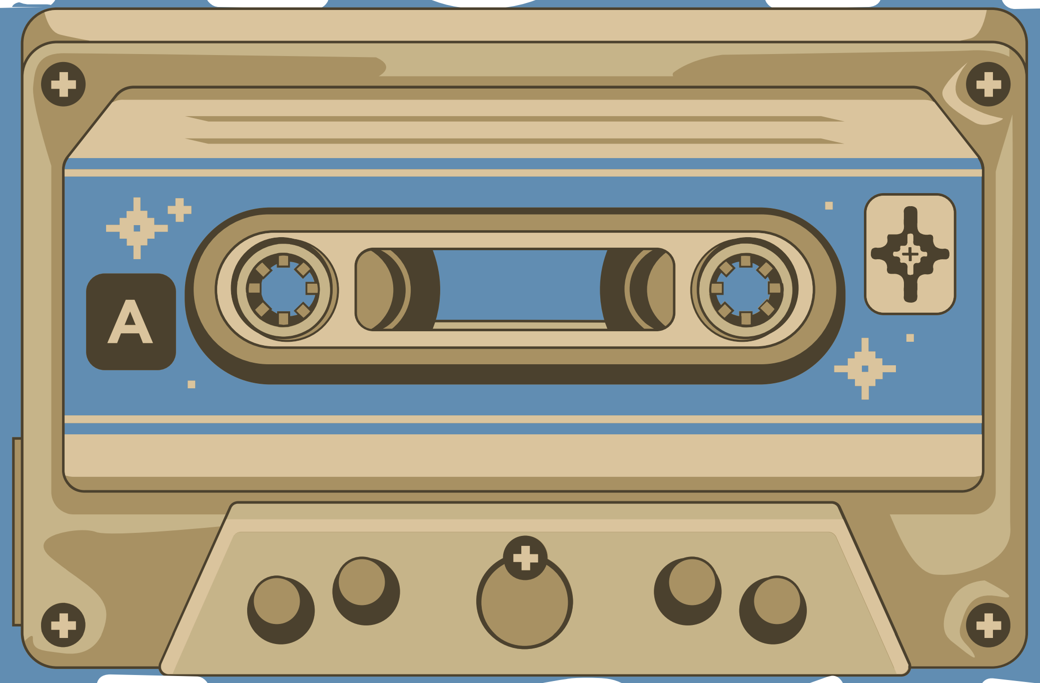
EMOTION

Melody

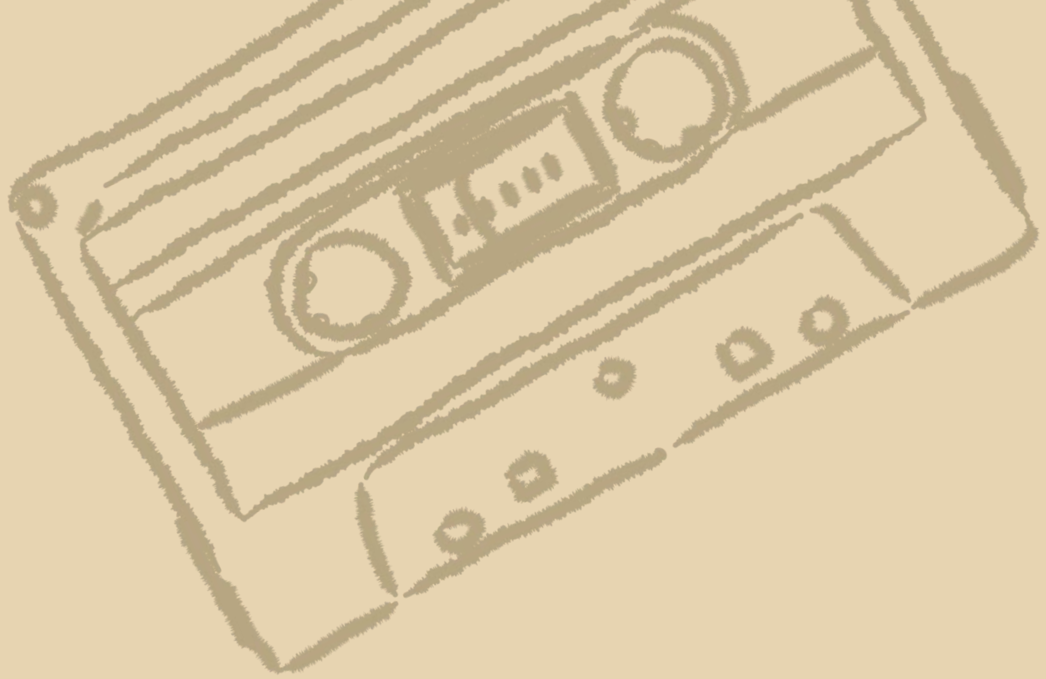


Stays!!

PLAYS

COLLABORATERS

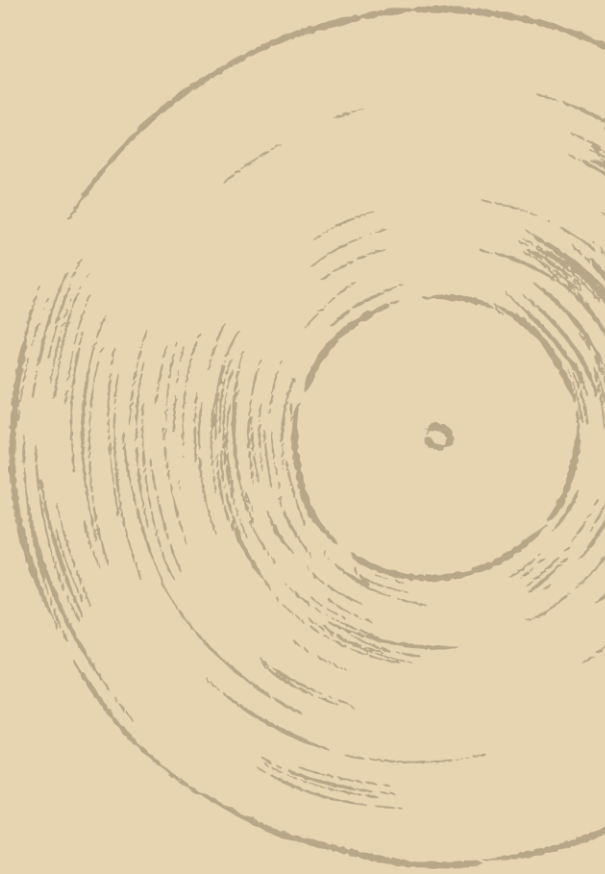
Sowjanya	AP23110011593
Harshitha	AP23110011657
Sravani	AP23110011656
Jagan Mohan Reddy	AP23110011664



INTRODUCTION

”Emotion-Based Music Player” is a Python-based application that seeks to bridge the gap between a user’s emotional state and music by automatically detecting facial expressions and playing music that corresponds to the detected mood.

Let’s start!



LET'S FIGURE IT OUT

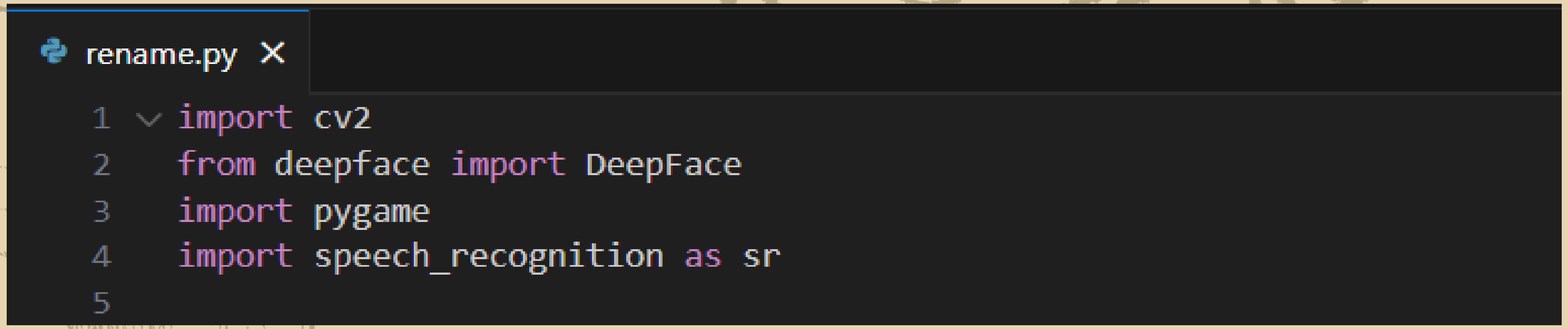
Objectives :

- To detect user emotions through facial expressions using computer vision.
- To play music that matches the user's current mood.
- To track and show mood patterns over time.
- To control the music player using voice commands.
- To provide a smooth and easy-to-use experience in real time.

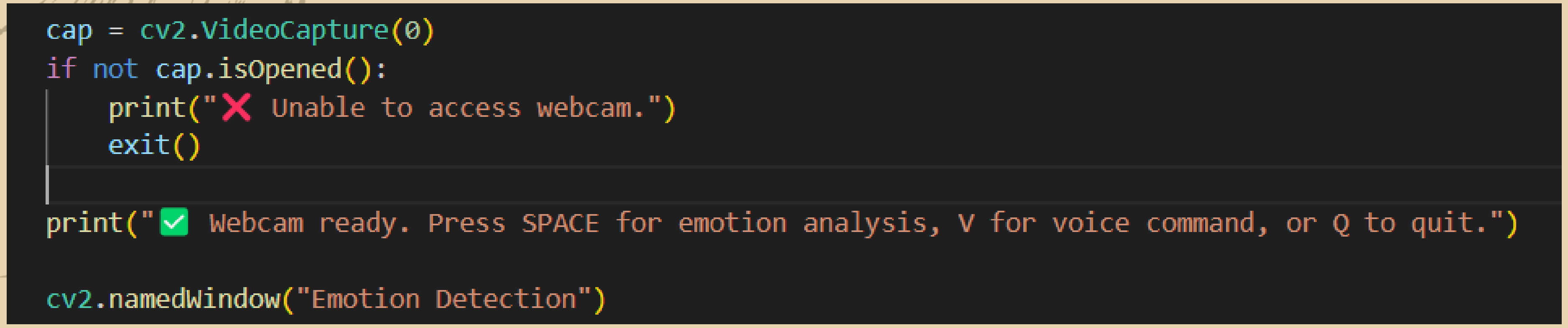
Methodology:

- MethodologyFace Detection (OpenCV):Captures live video and detects the user’s face using OpenCV.
- Emotion Recognition (DeepFace):Analyzes the face to identify the user’s emotion like happy, sad, or angry.
- Music Selection (Pygame):Plays music from a mood-based playlist using Pygame.
- Voice Control (SpeechRecognition):Lets users control the player with voice commands like play, pause, or skip..

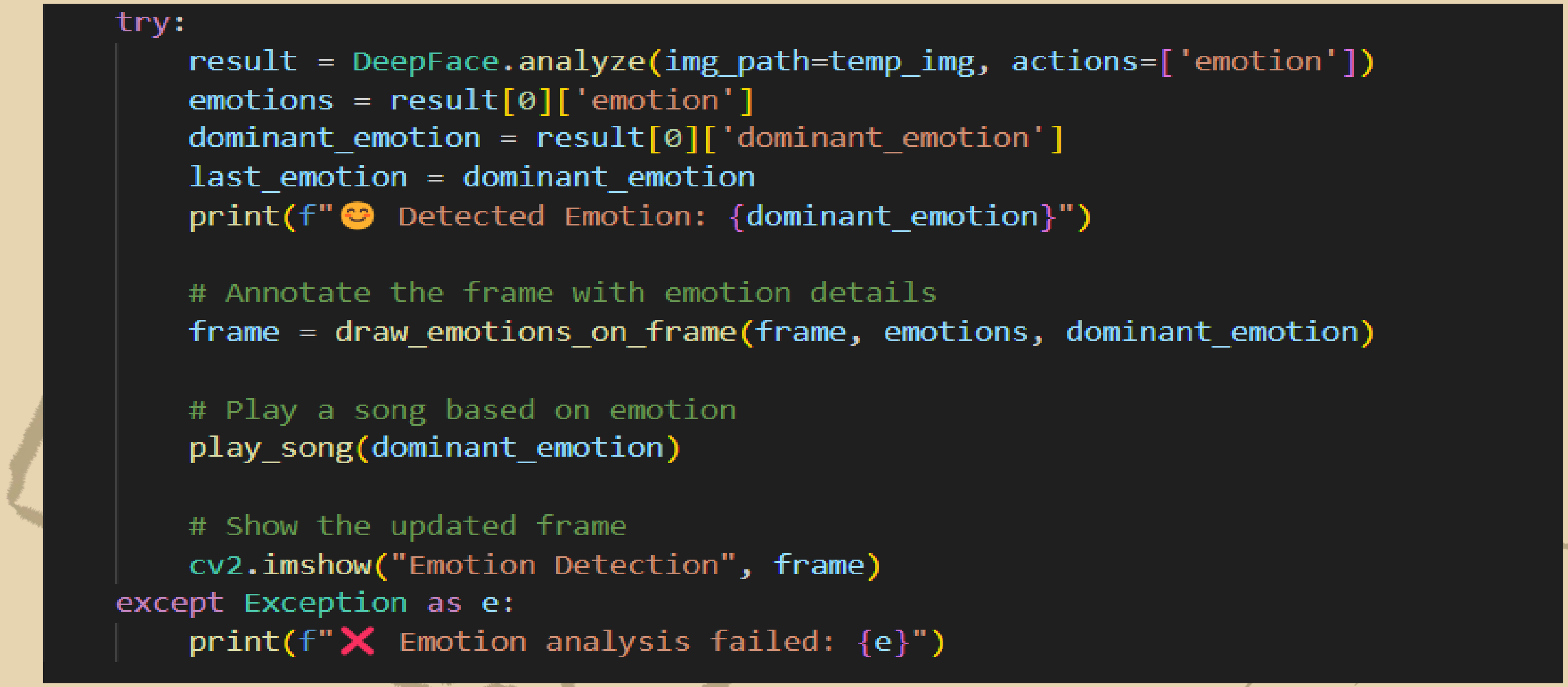
LET'S CHECK OUT THE CODE



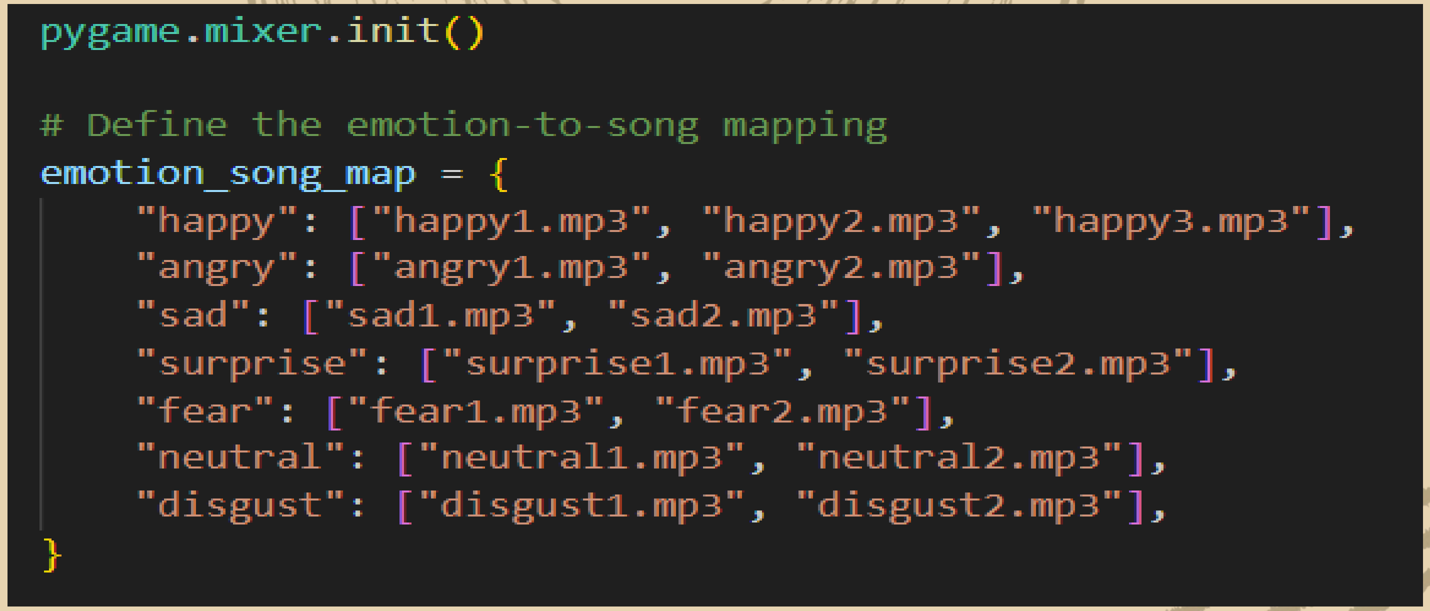
```python
import cv2
from deepface import DeepFace
import pygame
import speech_recognition as sr

```



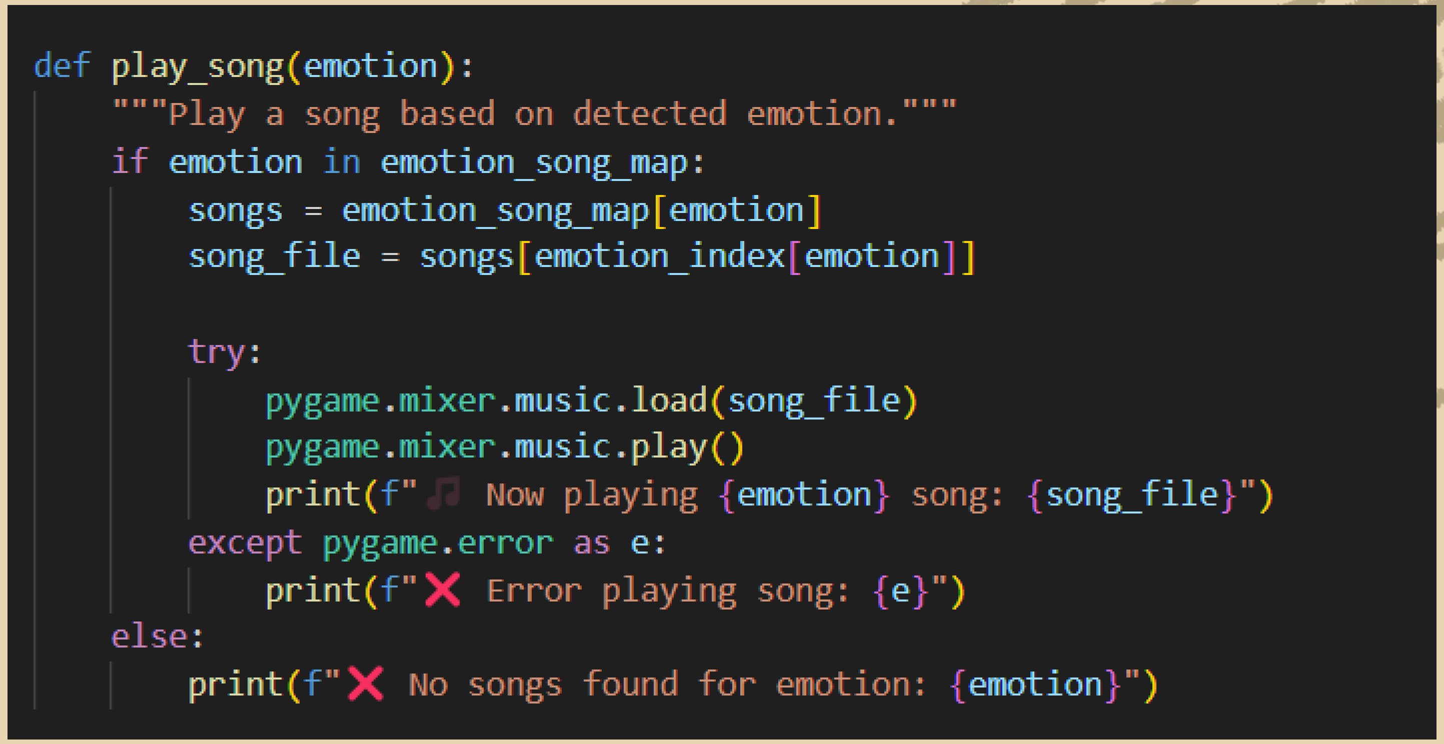
```python
cap = cv2.VideoCapture(0)
if not cap.isOpened():
    print("❌ Unable to access webcam.")
    exit()

print("✅ Webcam ready. Press SPACE for emotion analysis, V for voice command, or Q to quit.")

cv2.namedWindow("Emotion Detection")
```



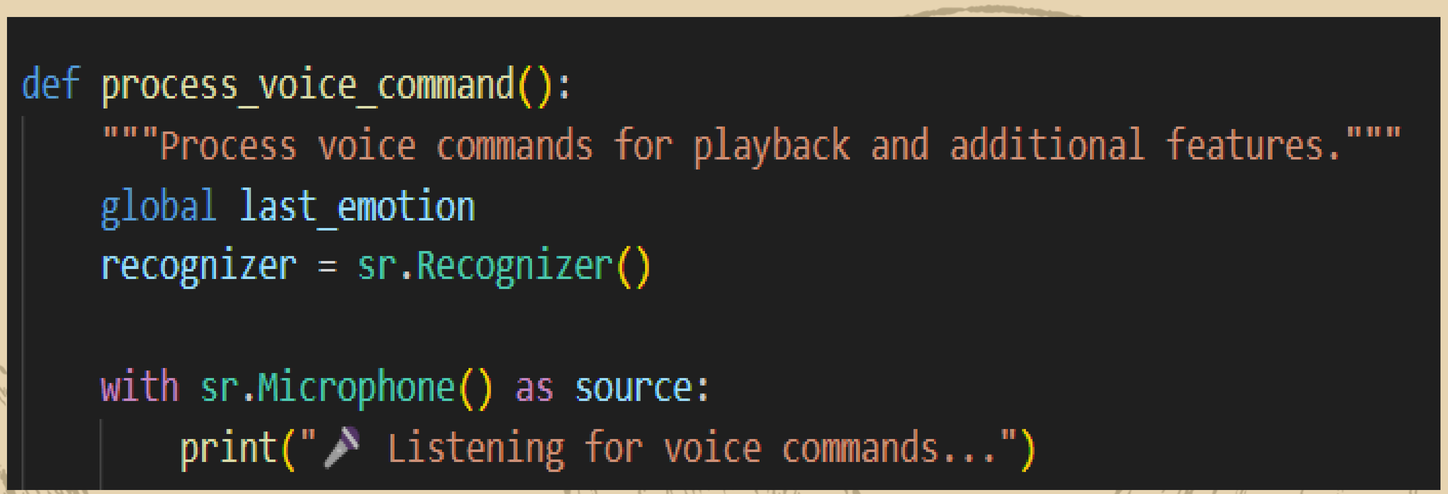
```python
    try:
        result = DeepFace.analyze(img_path=temp_img, actions=['emotion'])
        emotions = result[0]['emotion']
        dominant_emotion = result[0]['dominant_emotion']
        last_emotion = dominant_emotion
        print(f"😊 Detected Emotion: {dominant_emotion}")

        # Annotate the frame with emotion details
        frame = draw_emotions_on_frame(frame, emotions, dominant_emotion)

        # Play a song based on emotion
        play_song(dominant_emotion)

        # Show the updated frame
        cv2.imshow("Emotion Detection", frame)
    except Exception as e:
        print(f"❌ Emotion analysis failed: {e}")
```



```python
pygame.mixer.init()

# Define the emotion-to-song mapping
emotion_song_map = {
    "happy": ["happy1.mp3", "happy2.mp3", "happy3.mp3"],
    "angry": ["angry1.mp3", "angry2.mp3"],
    "sad": ["sad1.mp3", "sad2.mp3"],
    "surprise": ["surprise1.mp3", "surprise2.mp3"],
    "fear": ["fear1.mp3", "fear2.mp3"],
    "neutral": ["neutral1.mp3", "neutral2.mp3"],
    "disgust": ["disgust1.mp3", "disgust2.mp3"],
}
```



```python
def play_song(emotion):
    """Play a song based on detected emotion."""
    if emotion in emotion_song_map:
        songs = emotion_song_map[emotion]
        song_file = songs[emotion_index[emotion]]

        try:
            pygame.mixer.music.load(song_file)
            pygame.mixer.music.play()
            print(f"🎵 Now playing {emotion} song: {song_file}")
        except pygame.error as e:
            print(f"❌ Error playing song: {e}")
    else:
        print(f"❌ No songs found for emotion: {emotion}")
```



```python
def process_voice_command():
    """Process voice commands for playback and additional features."""
    global last_emotion
    recognizer = sr.Recognizer()

    with sr.Microphone() as source:
        print("🎤 Listening for voice commands...")
```

Conclusion:

This project demonstrates how emotion recognition can enhance user experience by delivering personalized music and activity suggestions. It successfully blends computer vision, audio playback, and voice interaction, laying the foundation for emotionally intelligent applications. With future improvements, it holds promise for use in mental wellness, entertainment, and adaptive user interfaces.

Result:

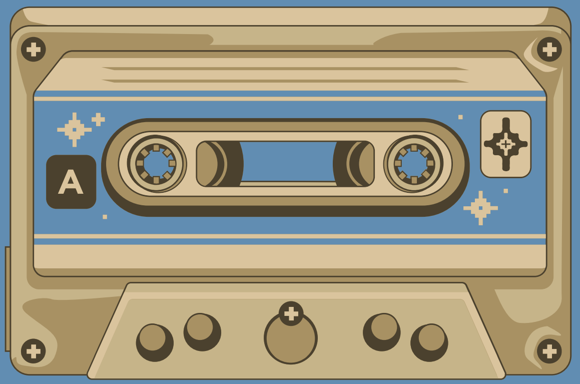
Detected real-time facial emotions using webcam.

Played emotion-matched songs automatically.

Responded to voice commands like play, pause, stop, and next.

Suggested mood-based activities.

Smooth integration of vision, audio, and speech modules.



THANK YOU FOR PLAYING!