Graphics and multimedia

Capture Video/Audio from Webcam or Microphone and Display on Multimedia Interface

# AIM

To develop a basic chat-like application that allows:  Sending and receiving text, images, and audio

 Using local flle selection and a simulated messaging UI

# Program

import tkinter as tk

from tkinter import filedialog, scrolledtext from PIL import Image, ImageTk

from playsound import playsound import threading

# Function to send text message def send\_text():

msg = text\_entry.get() if msg:

chat\_box.insert(tk.END, f"You: {msg}\n") text\_entry.delete(0, tk.END)

EXPERIMENT : 9

ROLL NO : 231701012

NAME : Divya Dharrshan T

# Function to send image def send\_image():

path = filedialog.askopenfilename(filetypes=[("Image Files", "\*.jpg \*.png \*.jpeg")]) if path:

chat\_box.insert(tk.END, f"You sent an image: {path}\n") img = Image.open(path)

img.thumbnail((100, 100))

img\_tk = ImageTk.PhotoImage(img)

image\_label = tk.Label(chat\_frame, image=img\_tk) image\_label.image = img\_tk

image\_label.pack()

chat\_box.yview(tk.END)

# Function to send audio def send\_audio():

path = filedialog.askopenfilename(filetypes=[("Audio Files", "\*.mp3 \*.wav")]) if path:

chat\_box.insert(tk.END, f"You sent an audio message: {path}\n") threading.Thread(target=lambda: playsound(path)).start()

# GUI Setup window = tk.Tk()

window.title("Multimedia Messaging App") window.geometry("400x500")

chat\_frame = tk.Frame(window)

chat\_box = scrolledtext.ScrolledText(chat\_frame, wrap=tk.WORD, width=50, height=20) chat\_box.pack()

chat\_frame.pack(pady=10)

text\_entry = tk.Entry(window, width=30) text\_entry.pack(side=tk.LEFT, padx=5)

btn\_text = tk.Button(window, text="Send Text", command=send\_text) btn\_image = tk.Button(window, text="Send Image", command=send\_image) btn\_audio = tk.Button(window, text="Send Audio", command=send\_audio)

btn\_text.pack(side=tk.LEFT) btn\_image.pack(side=tk.LEFT) btn\_audio.pack(side=tk.LEFT)

window.mainloop()

OUTPUT



# Result

A multimedia messaging interface was successfully created with options to send text, image, and audio content.

The application successfully simulated chat messaging behavior in a local environment.