App.py

---------------------------------------

import gradio as gr

from gtts import gTTS

import tempfile

import os

import math

# Memory store

gallery\_items = []

emergency\_info = {"number": "", "address": ""}

# Save temporary TTS audio

def text\_to\_speech(text):

if not text:

return None

tts = gTTS(text)

tmp\_path = os.path.join(tempfile.gettempdir(), "tts\_output.mp3")

tts.save(tmp\_path)

return tmp\_path

# Add picture + note

def add\_picture(img, name, note):

if img is None or not name:

return [item["img"] for item in gallery\_items], "⚠️ Please provide both name and image."

gallery\_items.append({"img": img, "name": name, "note": note})

return [item["img"] for item in gallery\_items], "✅ Added successfully!"

# Show text + speak aloud

def show\_picture\_info(evt: gr.SelectData):

idx = evt.index

if idx < 0 or idx >= len(gallery\_items):

return "⚠️ Could not identify the selected photo.", None

item = gallery\_items[idx]

caption = f"{item['name']}: {item['note']}"

audio\_file = text\_to\_speech(caption)

return caption, audio\_file # text + audio path

# Save emergency info

def save\_emergency(number, address):

emergency\_info["number"] = number

emergency\_info["address"] = address

return f"✅ Emergency info saved!\n📞 {number}\n🏠 {address}"

# Haversine distance in meters

def haversine\_m(lat1, lon1, lat2, lon2):

R = 6371000 # meters

phi1, phi2 = math.radians(lat1), math.radians(lat2)

dphi = math.radians(lat2 - lat1)

dlambda = math.radians(lon2 - lon1)

a = math.sin(dphi / 2) \*\* 2 + math.cos(phi1) \* math.cos(phi2) \* math.sin(dlambda / 2) \*\* 2

return 2 \* R \* math.asin(math.sqrt(a))

# Check geofence

def check\_geofence(center\_lat, center\_lon, radius\_m, test\_lat, test\_lon):

try:

d = haversine\_m(center\_lat, center\_lon, test\_lat, test\_lon)

inside = d <= radius\_m

if inside:

return f"✅ Inside safe zone (Distance: {d:.1f} m)"

else:

return f"⚠️ Outside safe zone! (Distance: {d:.1f} m)"

except Exception as e:

return f"Error: {str(e)}"

# Build UI

with gr.Blocks() as demo:

gr.Markdown("## 🧠 Memory Helper App")

with gr.Tab("Family & Friends"):

with gr.Row():

with gr.Column():

img\_in = gr.Image(type="filepath", label="Upload Photo")

name\_in = gr.Textbox(label="Name")

note\_in = gr.Textbox(label="Note / Relation")

add\_btn = gr.Button("Add to Album")

status = gr.Textbox(label="Status", interactive=False)

with gr.Column():

gallery = gr.Gallery(label="Click a photo to hear note", columns=3, height="auto")

selected\_info = gr.Textbox(label="Selected Info", interactive=False)

tts\_output = gr.Audio(label="Speaking Note", autoplay=True)

add\_btn.click(add\_picture, [img\_in, name\_in, note\_in], [gallery, status])

gallery.select(show\_picture\_info, None, [selected\_info, tts\_output])

with gr.Tab("Emergency Info"):

with gr.Row():

with gr.Column():

num\_in = gr.Textbox(label="Emergency Contact Number")

addr\_in = gr.Textbox(label="Home Address")

save\_btn = gr.Button("Save Info")

save\_status = gr.Textbox(label="Saved Info", interactive=False)

save\_btn.click(save\_emergency, [num\_in, addr\_in], save\_status)

with gr.Tab("Geofencing Simulator"):

gr.Markdown("### Define Safe Zone and Test Location")

with gr.Row():

with gr.Column():

center\_lat = gr.Number(label="Safe Zone Center Latitude", value=37.7749)

center\_lon = gr.Number(label="Safe Zone Center Longitude", value=-122.4194)

radius\_m = gr.Slider(label="Radius (meters)", value=500, minimum=50, maximum=5000, step=50)

with gr.Column():

test\_lat = gr.Number(label="Test Latitude", value=37.7750)

test\_lon = gr.Number(label="Test Longitude", value=-122.4180)

check\_btn = gr.Button("Check Location")

geo\_status = gr.Textbox(label="Result", interactive=False)

check\_btn.click(check\_geofence, [center\_lat, center\_lon, radius\_m, test\_lat, test\_lon], geo\_status)

demo.launch()