import telepot

from picamera import PiCamera

import RPi.GPIO as GPIO

import time

from time import sleep

from telepot.loop import MessageLoop

from subprocess import call

import threading

import smtplib

from email.message import EmailMessage

import os

# Email setup

EMAIL\_ADDRESS = 'agunwaxammy72@gmail.com'

EMAIL\_PASSWORD = 'dedp cttj tcss ldts'

EMAIL\_RECEIVER = 'recipient@example.com'

def send\_email(subject, body, attachment\_path=None):

msg = EmailMessage()

msg['Subject'] = subject

msg['From'] = EMAIL\_ADDRESS

msg['To'] = EMAIL\_RECEIVER

msg.set\_content(body)

if attachment\_path and os.path.exists(attachment\_path):

with open(attachment\_path, 'rb') as f:

file\_data = f.read()

file\_name = os.path.basename(attachment\_path)

# Use correct type based on extension

if file\_name.endswith(".jpg"):

maintype, subtype = 'image', 'jpeg'

else:

maintype, subtype = 'application', 'octet-stream'

msg.add\_attachment(file\_data, maintype=maintype, subtype=subtype, filename=file\_name)

try:

with smtplib.SMTP\_SSL('smtp.gmail.com', 465) as smtp:

smtp.login(EMAIL\_ADDRESS, EMAIL\_PASSWORD)

smtp.send\_message(msg)

print("📨 Email sent successfully.")

except Exception as e:

print("❌ Failed to send email:", e)

# GPIO and Camera setup

IR = 4

BUZZER = 6

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BCM)

GPIO.setup(IR, GPIO.IN)

GPIO.setup(BUZZER, GPIO.OUT)

GPIO.output(BUZZER, GPIO.LOW)

motion = 1

motionNew = 1

alarm\_enabled = False

# Initialize camera,

camera = PiCamera()

bot = telepot.Bot('7584611319:AAHAsG3dEUw5UH8Ad2itNFcKPps0602587A')

def sendNotification():

global camera # Ensure the camera is used correctly within the thread

timestamp = time.strftime("%y%b%d\_%H%M%S")

# Capture snapshot

photo\_file = f"./snapshot\_{timestamp}.jpg"

camera.capture(photo\_file)

print(f"Photo taken: {photo\_file}")

# Send photo on Telegram

bot.sendPhoto(chat\_id, photo=open(photo\_file, 'rb'))

bot.sendMessage(chat\_id, '🚨 An Intruder detected at faculty of engineering security cam')

# Send photo by email

send\_email(

subject="Security Alert: Motion Detected",

body="An Intruder detected at faculty of engineering security cam.",

attachment\_path=photo\_file

)

def monitor\_motion():

global motion, motionNew, chat\_id, alarm\_enabled

while True:

if GPIO.input(IR) == 0 and alarm\_enabled:

print("️Motion detected")

motion = 0

if motionNew != motion:

motionNew = motion

sendNotification()

else:

print("✅ No motion")

motion = 1

if motionNew != motion:

motionNew = motion

sleep(1) # To avoid 100% CPU usage and to give time for GPIO to change

def handle(msg):

global chat\_id, alarm\_enabled

chat\_id = msg['chat']['id']

text = msg['text'].lower()

print(f"📩 Message received: {text} from {chat\_id}")

if text == '/start':

bot.sendMessage(chat\_id, '📷 Security system is active.\nUse /activate or /alarm\_on to activate.')

elif text == '/activate':

alarm\_enabled = True

bot.sendMessage(chat\_id, '🔔 Motion detection enabled.')

elif text == '/deactivate':

alarm\_enabled = False

bot.sendMessage(chat\_id, '🔕 Motion detection disabled.')

elif text == '/alarm\_on':

GPIO.output(BUZZER, GPIO.HIGH)

sleep(0.5) # Delay in seconds

GPIO.output(BUZZER,GPIO.LOW)

sleep(0.5)

#bot.sendMessage(chat\_id, '📢 Buzzer turned ON.')

elif text == '/alarm\_off':

GPIO.output(BUZZER, GPIO.LOW)

bot.sendMessage(chat\_id, '🔇 Buzzer turned OFF.')

elif text == '/status':

buzzer\_state = GPIO.input(BUZZER)

status\_msg = f"🔔 Alarm: {'ON' if alarm\_enabled else 'OFF'}\n📢 Buzzer: {'ON' if buzzer\_state else 'OFF'}"

bot.sendMessage(chat\_id, status\_msg)

else:

bot.sendMessage(chat\_id, '❓ Unknown command.\nAvailable: /start, /activate, /deactivate, /alarm\_on, /alarm\_off, /status')

# Start bot and motion detection thread

MessageLoop(bot, handle).run\_as\_thread()

print('🤖 Bot is listening...')

motion\_thread = threading.Thread(target=monitor\_motion)

motion\_thread.start()

# Keep script running

while True:

time.sleep(10)