from flask import Flask, request, jsonify

from paho.mqtt import client as mqtt\_client

import sqlite3

app = Flask(\_\_name\_\_)

# MQTT Configuration

BROKER = 'broker.hivemq.com'

PORT = 1883

TOPIC = "flood/alert"

CLIENT\_ID = "backend\_flood\_detection"

# SQLite Database Configuration

db\_file = "flood\_data.db"

def init\_db():

conn = sqlite3.connect(db\_file)

cursor = conn.cursor()

cursor.execute('''CREATE TABLE IF NOT EXISTS alerts (

id INTEGER PRIMARY KEY AUTOINCREMENT,

message TEXT,

timestamp DATETIME DEFAULT CURRENT\_TIMESTAMP

)''')

conn.commit()

conn.close()

# MQTT Callback Functions

def on\_connect(client, userdata, flags, rc):

if rc == 0:

print("Connected to MQTT Broker!")

client.subscribe(TOPIC)

else:

print(f"Failed to connect, return code {rc}")

def on\_message(client, userdata, msg):

message = msg.payload.decode()

print(f"Received `{message}` from `{msg.topic}` topic")

# Save the alert in the database

conn = sqlite3.connect(db\_file)

cursor = conn.cursor()

cursor.execute("INSERT INTO alerts (message) VALUES (?)", (message,))

conn.commit()

conn.close()

# MQTT Client Setup

mqtt\_client = mqtt\_client.Client(CLIENT\_ID)

mqtt\_client.on\_connect = on\_connect

mqtt\_client.on\_message = on\_message

mqtt\_client.connect(BROKER, PORT)

# Start MQTT loop in a separate thread

mqtt\_client.loop\_start()

# API Endpoints

@app.route("/alerts", methods=["GET"])

def get\_alerts():

conn = sqlite3.connect(db\_file)

cursor = conn.cursor()

cursor.execute("SELECT \* FROM alerts ORDER BY timestamp DESC")

alerts = cursor.fetchall()

conn.close()

return jsonify([{

"id": alert[0],

"message": alert[1],

"timestamp": alert[2]

} for alert in alerts])

@app.route("/alerts", methods=["POST"])

def add\_test\_alert():

data = request.json

message = data.get("message", "Test alert")

conn = sqlite3.connect(db\_file)

cursor = conn.cursor()

cursor.execute("INSERT INTO alerts (message) VALUES (?)", (message,))

conn.commit()

conn.close()

return jsonify({"status": "Alert added", "message": message}), 201

if \_\_name\_\_ == "\_\_main\_\_":

init\_db()

app.run(host="0.0.0.0", port=5000)