```
#include <SoftwareSerial.h>
#include <DHT.h> // Include DHT sensor library

#define PIN_TX 11

#define PIN_RX 10
#define PHONE_NUMBER "+916009213177" // Replace with your phone number

#define FLAME_SENSOR_PIN 2       // Pin connected to the flame sensor
#define DHT_PIN A1              // Pin connected to the DHT sensor
#define DHT_TYPE DHT11  // DHT11 sensor type
#define GAS_PIN A0
#define BUZZER_PIN 4

SoftwareSerial sim808Serial(PIN_RX, PIN_TX);
DHT dht(DHT_PIN, DHT_TYPE);   // Initialize DHT sensor

void setup() {
  Serial.begin(9600);
  sim808Serial.begin(9600);
  pinMode(FLAME_SENSOR_PIN, INPUT);
  pinMode(GAS_PIN, INPUT);
  pinMode(BUZZER_PIN, OUTPUT);
  dht.begin();

  Serial.println("Initializing GPS...");
  sim808Serial.println("AT+CGNSPWR=1"); // Turn on GPS
  delay(1000);
}

 bool isFireDetected = false; // Fire detection logic

void loop() {
  bool needtosent=true;
  int flameSensorValue = digitalRead(FLAME_SENSOR_PIN);// Read flame sensor
   int gasLevel = analogRead(GAS_PIN);
  float temperature = dht.readTemperature();           // Read temperature
  float humidity = dht.readHumidity();                 // Read humidity

  if( isFireDetected==true){
    needtosent=false;
  }
  if( isFireDetected==false){
    needtosent=true;
  }
  // Fire condition thresholds (adjust as needed)
  if (flameSensorValue == LOW || gasLevel > 300 ||  temperature > 50 || humidity < 30) {
    isFireDetected = true; // Fire detected
    digitalWrite(BUZZER_PIN, HIGH);
    delay(2000);
    digitalWrite(BUZZER_PIN, LOW);
  }

  if (needtosent && isFireDetected) {
    Serial.println("Fire condition detected. Sending GPS data...");

    // Fetch GPS data if fire is detected
    Serial.println("Checking GPS Fix Status...");
    sim808Serial.println("AT+CGPSSTATUS?"); // Check GPS fix status
    delay(2000);
    String gpsStatus = getModuleResponse();

    if (gpsStatus.indexOf("Location 3D Fix") != -1) {
      Serial.println("Fetching GPS Data...");
      sim808Serial.println("AT+CGNSINF"); // Get GPS data
      delay(2000);
      String gpsData = getModuleResponse();

      // Parse GPS data
      float latitude = parseLatitude(gpsData);
      float longitude = parseLongitude(gpsData);
      String dateTimeIST = parseDateTimeIST(gpsData);

      if (latitude != 0 && longitude != 0) {
        // Generate Google Maps link
        String gmapLink = "https://www.google.com/maps?q=" + String(latitude, 6) + "," +
String(longitude, 6);

        // Display on Serial Monitor
        Serial.println("Latitude: " + String(latitude, 6));
        Serial.println("Longitude: " + String(longitude, 6));
        Serial.println("Google Maps Link: " + gmapLink);

        // Prepare SMS content with fire detection details
        String smsContent = "Fire Detected! Date/Time (IST): " + dateTimeIST + "\n" +
                            "Temperature: " + String(temperature) + "C\n" +
                            "Humidity: " + String(humidity) + "%\n" +
                            "Latitude: " + String(latitude, 6) + "\n" +
                            "Longitude: " + String(longitude, 6) + "\n" +
                            "Map: " + gmapLink;

        // Send SMS
        sendSMS(smsContent);
      }
    } else {
      Serial.println("Waiting for GPS fix...");
    }
  } else {
    Serial.println("No fire detected.");
    isFireDetected = false;
  }
  delay(2000); // Retry every 2 sec
}

String getModuleResponse() {
  String response = "";
  while (sim808Serial.available()) {
    char c = sim808Serial.read();
    response += c;
  }
  Serial.println(response); // Display raw response
  return response;
}

float parseLatitude(String gpsData) {
  int startIdx = nthIndexOf(gpsData, ',', 3) + 1; // Find the 3rd comma
  int endIdx = nthIndexOf(gpsData, ',', 4);       // Find the 4th comma
  return gpsData.substring(startIdx, endIdx).toFloat();
}

float parseLongitude(String gpsData) {
  int startIdx = nthIndexOf(gpsData, ',', 4) + 1; // Find the 4th comma
  int endIdx = nthIndexOf(gpsData, ',', 5);       // Find the 5th comma
  return gpsData.substring(startIdx, endIdx).toFloat();
}

String parseDateTimeIST(String gpsData) {
  int startIdx = nthIndexOf(gpsData, ',', 2) + 1; // Find the 2nd comma
  int endIdx = nthIndexOf(gpsData, ',', 3);       // Find the 3rd comma
  String dateTimeUTC = gpsData.substring(startIdx, endIdx);

  if (dateTimeUTC.length() < 14) return "Invalid Date/Time";

  int year = dateTimeUTC.substring(0, 4).toInt();
  int month = dateTimeUTC.substring(4, 6).toInt();
  int day = dateTimeUTC.substring(6, 8).toInt();
  int hourUTC = dateTimeUTC.substring(8, 10).toInt();
  int minute = dateTimeUTC.substring(10, 12).toInt();
  int second = dateTimeUTC.substring(12, 14).toInt();

  // Convert UTC to IST (UTC + 5:30)
  int hourIST = hourUTC + 5;
  int minuteIST = minute + 30;

  if (minuteIST >= 60) {
    minuteIST -= 60;
    hourIST++;
  }
  if (hourIST >= 24) {
    hourIST -= 24;
    day++; // Adjust for next day; you may need a proper date library for month-end
  }

  char dateTimeIST[20];
  sprintf(dateTimeIST, "%04d-%02d-%02d %02d:%02d:%02d", year, month, day, hourIST,
minuteIST, second);
  return String(dateTimeIST);
}

// Utility function to find the nth occurrence of a character in a string
int nthIndexOf(String str, char c, int n) {
  int index = -1;
  while (n-- > 0) {
    index = str.indexOf(c, index + 1);
    if (index == -1) break;
  }
  return index;
}

void sendSMS(String message) {
  Serial.println("Sending SMS...");
  sim808Serial.println("AT+CMGF=1"); // Set SMS to Text Mode
  delay(1000);
  sim808Serial.println("AT+CMGS=\"" + String(PHONE_NUMBER) + "\"");
  delay(1000);
  sim808Serial.print(message);
  sim808Serial.write(26); // ASCII code of CTRL+Z to send the SMS
  delay(5000);
  Serial.println("SMS Sent!");
}
```