#include <Keypad.h>  // Include the Keypad library

#include <SoftwareSerial.h>

// Define constants for buzzer, touch sensor, pulse sensor, microphone, temperature sensor, and pressure sensor

const int buzzerPin = 11;  // Pin connected to the buzzer

const int touchSensorPin = 12;  // Pin connected to the touch sensor

const int pulseSensorPin = A0;  // Analog pin for the pulse sensor

const int microphonePin = A1;  // Pin for the microphone

const int tempSensorPin = A2;  // Analog pin for the temperature sensor

const int pressureSensorPin = A4;  // Analog pin for the pressure sensor

const int pulseThreshold = 300;  // Threshold for pulse

const int soundThreshold = 200;  // Threshold for sound

const float tempThreshold = 30.0;  // Temperature threshold in degrees Celsius

const int pressureThreshold = 1000;  // Pressure sensor threshold value

SoftwareSerial mySerial(13, 14); // RX = 3, TX = 2 (ensure these pins are compatible with your Arduino model for SoftwareSerial)

const byte ROWS = 4;  // Four rows on the keypad

const byte COLS = 4;  // Four columns on the keypad

// Define the key layout

char keys[ROWS][COLS] = {

    {'1', '2', '3', 'A'},

    {'4', '5', '6', 'B'},

    {'7', '8', '9', 'C'},

    {'\*', '0', '#', 'D'}

};

// Define the row and column pinouts

byte rowPins[ROWS] = {9, 8, 7, 6};  // Connect to the row pinouts

byte colPins[COLS] = {5, 4, 3, 2};  // Connect to the column pinouts

// Create the Keypad object

Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);

// Variable to track if the buzzer is on or off

bool buzzerState = false;

void setup() {

    pinMode(buzzerPin, OUTPUT);  // Set the buzzer pin as output

    pinMode(touchSensorPin, INPUT);  // Set the touch sensor pin as input

    pinMode(pulseSensorPin, INPUT);  // Set the pulse sensor pin as input

    pinMode(microphonePin, INPUT);  // Set the microphone pin as input

    pinMode(tempSensorPin, INPUT);  // Set the temperature sensor pin as input

    pinMode(pressureSensorPin, INPUT);  // Set the pressure sensor pin as input

    digitalWrite(buzzerPin, LOW);  // Initialize the buzzer to be off

    mySerial.begin(9600);

    Serial.begin(9600);  // Start serial communication for debugging

}

void loop() {

    char key = keypad.getKey();  // Get the key from the keypad

    int touchState = digitalRead(touchSensorPin);  // Read the touch sensor state

    int pulseValue = analogRead(pulseSensorPin);  // Read the pulse sensor value

    int soundValue = analogRead(microphonePin);  // Read the microphone sensor value

    int tempAnalogValue = analogRead(tempSensorPin);  // Read the temperature sensor value

    int pressureValue = analogRead(pressureSensorPin);  // Read the pressure sensor value

    // Convert analog temperature reading to degrees Celsius (assuming LM35)

    float temperature = (tempAnalogValue \* 5.0 / 1023.0) \* 100.0;

    // Print pulse, sound, temperature, and pressure sensor values to the serial monitor

    Serial.print("Pulse Sensor Value: ");  // Debug message

    Serial.println(pulseValue);

    Serial.print("Sound Sensor Value: ");  // Debug message

    Serial.println(soundValue);

    Serial.print("Temperature: ");  // Debug message

    Serial.println(temperature);

    Serial.print("Pressure Sensor Value: ");  // Debug message

    Serial.println(pressureValue);

    Serial.print("Meassage Sent!");

    // Case 1: If touch sensor is activated, pulse is above threshold, sound is above threshold, and temperature is high

    if (touchState == HIGH && pulseValue > pulseThreshold && soundValue > soundThreshold && temperature > tempThreshold && !buzzerState) {

        digitalWrite(buzzerPin, HIGH);  // Turn on the buzzer

        buzzerState = true;  // Update the buzzer state

        Serial.println("Buzzer turned on! Suspicious sound, high pulse, touch detected, and high temperature.");  // Debug message

    }

    // Case 2: If pressure sensor value is above 300, turn on the buzzer

    if (pressureValue > pressureThreshold && !buzzerState) {

        digitalWrite(buzzerPin, HIGH);  // Turn on the buzzer

        buzzerState = true;  // Update the buzzer state

        Serial.println("Buzzer turned on due to high pressure.");  // Debug message

        Serial.print("Messacge sent!");

    }

    // If the buzzer is on and any key other than '1' is pressed, turn it off

    if (buzzerState && key != NO\_KEY && key != '1') {

        digitalWrite(buzzerPin, LOW);  // Turn off the buzzer

        buzzerState = false;  // Update the buzzer state

        Serial.print("Buzzer turned off. Key Pressed: ");  // Debug message

        Serial.println(key);

    }

    mySerial.println("AT+CMGF=1");         // Set SMS to text mode

  delay(500);                           // Wait for the module to respond

  mySerial.println("AT+CMGS=\"+918668017538\"\r");  // Replace number with your target number

  delay(500);                           // Wait for the '>' prompt from the module

  mySerial.println("Alert message");     // Message content

  delay(100);                            // Short pause to ensure the complete message is in the buffer

  mySerial.write(26);                    // ASCII code for Ctrl+Z to send the message

  delay(500);                           // Wait for the message to be sent

  Serial.println("Message sent!");       // Debug output to hardware serial

    delay(100);  // Short delay to avoid rapid toggling and stabilize readings

}