```
//ESP8266
#include <ESP8266WiFi.h>
#include <dht.h> // inlcude the library of DHT11
#define dht_apin 5  // define the analog pin that we need to connect
dht DHT; // giving out dht sensor a name
#include <ArduinoJson.h>

 ////////////////////////////////
String apiKey = "D8VOXM6F1Q8AQUE6";     //  Enter your Write API key from ThingSpeak
const char *ssid =  "ahmed osama";     // replace with your wifi ssid and wpa2 key
const char *pass =  "Ahmed123456789";
const char* server = "api.thingspeak.com";
 ////////////////////////////////



unsigned long previousMillis = 0;
unsigned long currentMillis;
const unsigned long period = 10000;
WiFiClient client;

void setup() {
 Serial.begin(9600);

  // put your setup code here, to run once:

      while (!Serial) continue;
       delay(10);
       Serial.println("Connecting to ");
       Serial.println(ssid);
       WiFi.begin(ssid, pass);
      while (WiFi.status() != WL_CONNECTED)
```

```
   {
         delay(500);
         Serial.print(".");
   }
    Serial.println("");
    Serial.println("WiFi connected");
  Serial.println("Program started");
}

void loop()
{
  float  T;float H;
  StaticJsonDocument<1000> doc;
DHT.read11(dht_apin); //function to read the values from the pin A0
 H=DHT.humidity;
 T=DHT.temperature;
 float h = analogRead(A0);
float G =h/1023*100;
doc["H"]=H;
 doc["T"]=T;
 doc["G"]=G;
  // put your main code here, to run repeatedly:
serializeJsonPretty(doc, Serial);

///////////////////////////////////////////////////
///////////////////////////////////////////////////
///////////////////////////////////////////////////


if (client.connect(server,80))   //   "184.106.153.149" or api.thingspeak.com
{
 String postStr = apiKey;
 postStr +="&field1=";
 postStr += String(T);
 postStr +="&field2=";
 postStr += String(H);
 postStr += "&field3=";
 postStr += String(G);
 postStr += "\r\n\r\n";

 client.print("POST /update HTTP/1.1\n");
 client.print("Host: api.thingspeak.com\n");
 client.print("Connection: close\n");
 client.print("X-THINGSPEAKAPIKEY: "+apiKey+"\n");
 client.print("Content-Type: application/x-www-form-urlencoded\n");
client.print("Content-Length: ");
client.print(postStr.length());
 client.print("\n\n");
client.print(postStr);

}
client.stop();

  // thingspeak needs minimum 15 sec delay between updates



}
```

```
//ARDUINO
#include <Wire.h>
#include <Keypad.h>
#include <LiquidCrystal_I2C.h>
```

```
#include <SoftwareSerial.h>

#include <ArduinoJson.h>// SCL to   A5 & SDA to A4

LiquidCrystal_I2C lcd(0x27,16,2);

#define green_led A3

#define red_led A2

#define buzzer 12

#define fan 13

#define lamb 10

const byte ROWS = 4; //four rows

const byte COLS = 4; //four columns

//define the cymbols on the buttons of the keypads

char Keys[ROWS][COLS] = {

  {'1', '2', '3', 'A'},

  {'4', '5', '6', 'B'},

  {'7', '8', '9', 'C'},

  {'*', '0', '#', 'D'}

};

byte rowPins[ROWS] = {2,3,4,5};

byte colPins[COLS] = {6,7,8,9};

//initialize an instance of class NewKeypad

Keypad customKeypad = Keypad( makeKeymap(Keys), rowPins, colPins, ROWS, COLS);

boolean presentValue = false;

boolean final;

String num1;

int num;

char op='D';

float H;

float T;

float G;

 //////////////////////////////

/////////////////////////////////////////////////////////////////
```

```
/////////////////////////////////////////////////////////////////
void setup(){
/////////////////////////////////////////////////////////////////
Serial.begin(9600);
  lcd.init();
  lcd.backlight();
 pinMode(red_led,OUTPUT);
pinMode(green_led,OUTPUT);
pinMode(buzzer,OUTPUT);
pinMode(fan,OUTPUT);
pinMode(lamb,OUTPUT);

    /////////////////////////////
    lcd.setCursor(0, 0);
    lcd.print(F("  WELCOME USER"));
    delay(3000);
    lcd.clear();
    /////////////////////////////
   lcd.setCursor(0,0);
   lcd.print("SET Temp:");
}
void loop(){
            digitalWrite(red_led,LOW);
            digitalWrite(green_led,LOW);
            digitalWrite(buzzer,LOW);
            digitalWrite(fan,LOW);
             digitalWrite(10,LOW);

   char key = customKeypad.getKey();
//////////////////////////////////////////////////////////////////////////////
```

```
  if ( key != NO_KEY &&
(key=='1'key=='2'||key=='3'||key=='4'||key=='5'||key=='6'||key=='7'||key=='8'||key=='9'|
|key=='0'))

 {

   if (presentValue != true)

   {

     num1 = num1 + key;

     int numLength = num1.length();

     lcd.setCursor(15 - numLength, 0); //to adjust one whitespace for operator

     lcd.print(num1);


   }

   }else if (key != NO_KEY && key == 'C')

       {


      lcd.clear();

       presentValue = false;

       final = false;

       num1 = "";

       op = ' ';

        lcd.setCursor(0,0);

      lcd.print("SET T:");

     } else if ( final = true&&  key != NO_KEY && key == 'D')

   {

      lcd.clear();

       num= num1.toInt();

////////////////////////////////////////////////////////////////////////////////////


while(true)

{

  StaticJsonDocument<1000> doc;

deserializeJson(doc, Serial);
```

```
H = doc["H"];
T = doc["T"];
G = doc["G"];
            lcd.setCursor(0,0);
         lcd.print("T=");
         lcd.setCursor(2,0);
         lcd.print(T);
          lcd.setCursor(6,0);
         lcd.print("C");

        lcd.setCursor(7,0);
         lcd.print("&");
         lcd.setCursor(8,0);
         lcd.print("H=");
          lcd.setCursor(10,0);
         lcd.print(H);
          lcd.setCursor(15,0);
         lcd.print("%");
           lcd.setCursor(0,1);
         lcd.print("G=");
          lcd.setCursor(2,1);
         lcd.print(G);
          lcd.setCursor(6,1);
         lcd.print("%");
         lcd.setCursor(7,1);
         lcd.print("");
               if (T >=num ){

          digitalWrite(red_led,HIGH);
           digitalWrite(green_led,LOW);
         digitalWrite(buzzer,HIGH);
```

```
           digitalWrite(fan,HIGH);
          digitalWrite(lamb,LOW);


          lcd.setCursor(9,1);
          lcd.print("DANGER   ");
           }else if (T<=num)
           {


           digitalWrite(red_led,LOW);
           digitalWrite(green_led,HIGH);
           digitalWrite(buzzer,LOW);
           digitalWrite(fan,LOW);
           digitalWrite(lamb,HIGH);


         lcd.setCursor(9,1);
          lcd.print("SAFE    ");


             }


 }
   }
}
```