**AIR QUALITY MONITORING USING IOT**

**PHASE-4 WEB DEVELOPMENT**

In this phase, we have developed a BLYNK PAGE of our Wowki simulation, which would show the PCO2 and humidity values.

**CODE:**

#define BLYNK\_TEMPLATE\_ID "TMPLgwKssgggsnFXp"

#define BLYNK\_DEVICE\_NAME "Air Quality Monitoring"

#define BLYNK\_AUTH\_TOKEN "k03gT6nJosdsfsffesrJV\_S5SXEAdgdsdghhgPZvXEwSKDfj"

#define BLYNK\_PRINT **Serial**

#include <WiFi.h>

#include <BlynkSimpleEsp8266.h>

#include <DHT.h>

//#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd(0x27, 16, 2);

  byte degree\_symbol[8] =

              {

                0b00111,

                0b00101,

                0b00111,

                0b00000,

                0b00000,

                0b00000,

                0b00000,

                0b00000

              };

char auth[] = BLYNK\_AUTH\_TOKEN;

char ssid[] = "";  // type your wifi name

char pass[] = "";  // type your wifi password

BlynkTimer timer;

int gas = A0;

int sensorThreshold = 100;

#define DHTPIN 2 //Connect Out pin to D2 in NODE MCU

#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);

void sendSensor()

{

  float h = dht.readHumidity();

  float t = dht.readTemperature(); // or dht.readTemperature(true) for Fahrenheit

     if (isnan(h) || isnan(t)) {

**Serial**.println("Failed to read from DHT sensor!");

    return;

  }

   int analogSensor = analogRead(gas);

  Blynk.virtualWrite(V2, analogSensor);

**Serial**.print("Gas Value: ");

**Serial**.println(analogSensor);

  // You can send any value at any time.

  // Please don't send more that 10 values per second.

    Blynk.virtualWrite(V0, t);

    Blynk.virtualWrite(V1, h);

**Serial**.print("Temperature : ");

**Serial**.print(t);

**Serial**.print("    Humidity : ");

**Serial**.println(h);

}

void setup()

{

**Serial**.begin(115200);

 //pinMode(gas, INPUT);

  Blynk.begin(auth, ssid, pass);

  dht.begin();

  timer.setInterval(30000L, sendSensor);

 //Wire.begin();

   lcd.begin();

//  lcd.backlight();

 // lcd.clear();

  lcd.setCursor(3,0);

  lcd.print("Air Quality");

  lcd.setCursor(3,1);

  lcd.print("Monitoring");

  delay(2000);

  lcd.clear();

  }

void loop()

{

  Blynk.run();

  timer.run();

 float h = dht.readHumidity();

  float t = dht.readTemperature(); // or dht.readTemperature(true) for Fahrenheit

    int gasValue = analogRead(gas);

  lcd.setCursor(0,0);

  lcd.print("Temperature ");

  lcd.setCursor(0,1);

  lcd.print(t);

  lcd.setCursor(6,1);

  lcd.write(1);

  lcd.createChar(1, degree\_symbol);

  lcd.setCursor(7,1);

  lcd.print("C");

  delay(4000);

  lcd.clear();

  lcd.setCursor(0, 0);

  lcd.print("Humidity ");

  lcd.print(h);

  lcd.print("%");

  delay(4000);

  lcd.clear();

  //lcd.setCursor(0,0);

 // lcd.print(gasValue);

 // lcd.clear();

  if(gasValue<600)

  {

    lcd.setCursor(0,0);

    lcd.print("Gas Value: ");

    lcd.print(gasValue);

    lcd.setCursor(0, 1);

    lcd.print("Fresh Air");

**Serial**.println("Fresh Air");

    delay(4000);

    lcd.clear();

  }

  else if(gasValue>600)

  {

    lcd.setCursor(0,0);

    lcd.print(gasValue);

    lcd.setCursor(0, 1);

    lcd.print("Bad Air");

**Serial**.println("Bad Air");

    delay(4000);

    lcd.clear();

  }

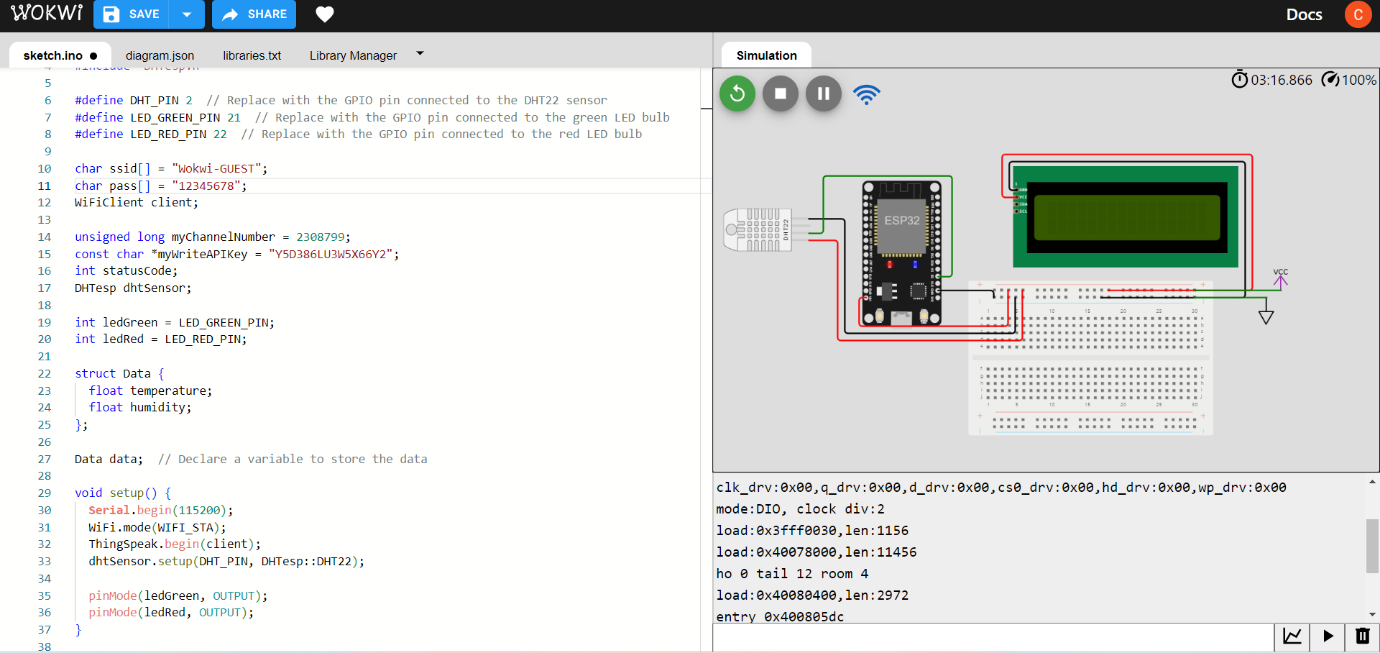
   if(gasValue > 600){

    //Blynk.email("abcdefghi@gmail.com", "Alert", "Bad Air!");

    Blynk.logEvent("pollution\_alert","Bad Air");

  }

 }



OUTPUT: 