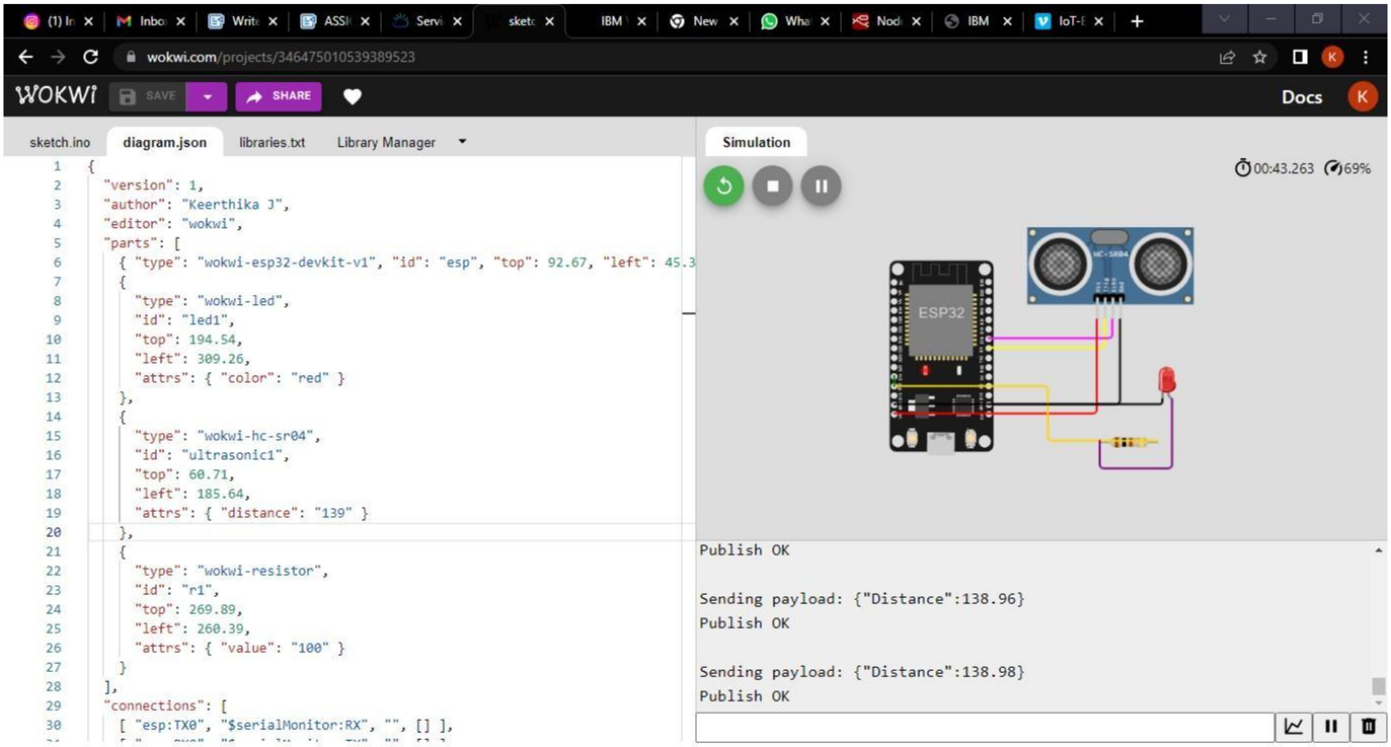
ASSIGNMENT 4

Write code and connections in working for ultrasonic sensor. Whenever the distance is less than 100 cms send "alert" to IBM cloud and display in device recent events.

code:

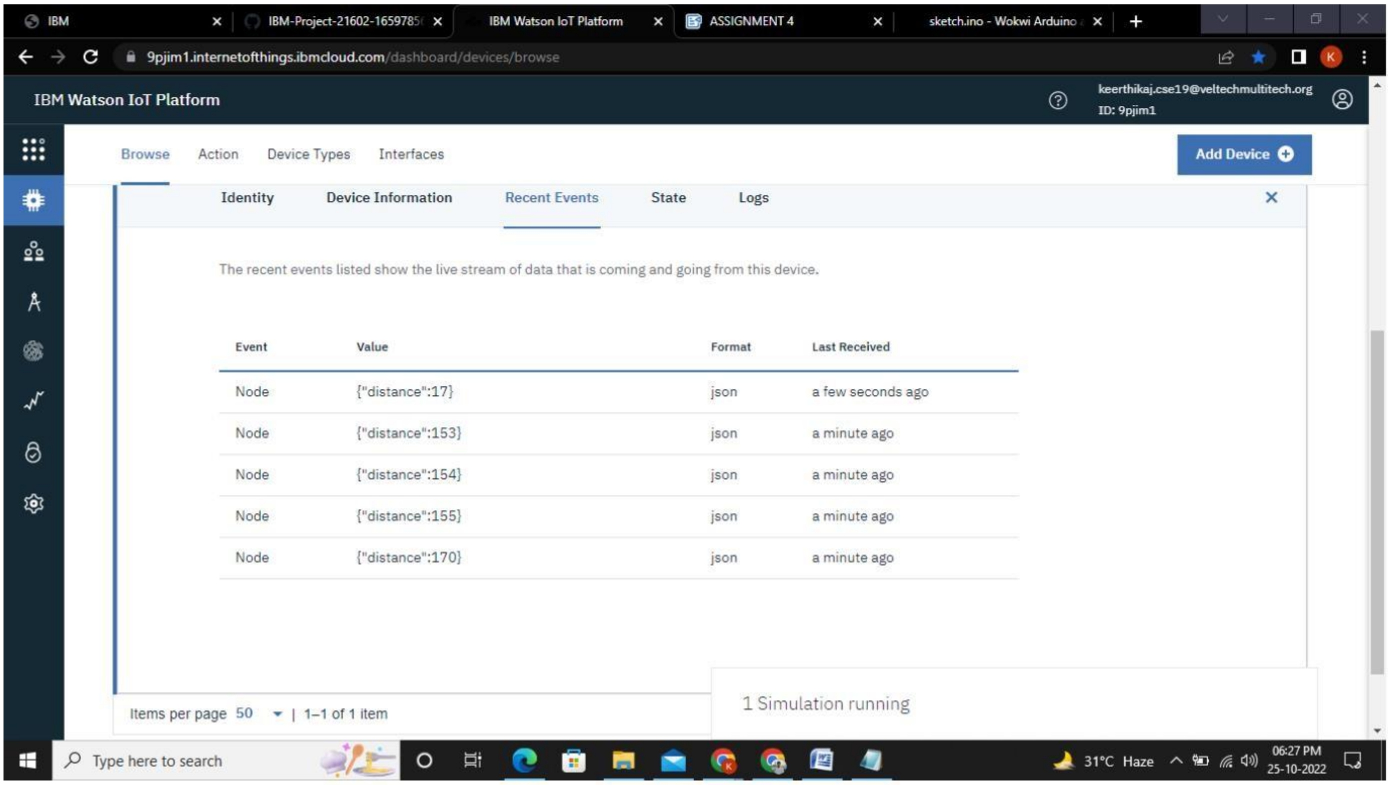
```
#include <WiFi.h>
#include <PubSubClient.h>
WiFiClient wifClient;
String data3;
#defne ORG "x0fxss"
#defne DEVICE_TYPE "Noder"
#defne DEVICE_ID "1234"
#defne TOKEN "987654321"
#defne speed 0.034
#defne led 14
char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; char
publishTopic[] = "iot-2/evt/shanmugam_assignment4/fmt/json"; char topic[] =
"iot-2/cmd/home/fmt/String"; char authMethod[] = "use-token-auth";

char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
PubSubClient client(server, 1883, wifClient);



const int trigpin=5;
const int echopin=18;
String command;
String data="";

long duration;
foat dist;



void setup()
{
    Serial.begin(115200);
    pinMode(led, OUTPUT);
    pinMode(trigpin, OUTPUT);
    pinMode(echopin, INPUT);
    wifConnect();
    mqttConnect();
}

void loop() {
    bool isNearby = dist < 100;
    digitalWrite(led, isNearby);

    publishData();
    delay(500);

    if (!client.loop()) {
        mqttConnect();
    }
}

void wifConnect() {
    Serial.print("Connecting to "); Serial.print("Wif");
    WiFi.begin("Wokwi-GUEST", "", 6);
    while (WiFi.status() != WL_CONNECTED)
        { delay(500);
        Serial.print(".");
    }
    Serial.print("WiFi connected, IP address: ");
Serial.println(WiFi.localIP()); }


void mqttConnect() {
    if (!client.connected()) {
        Serial.print("Reconnecting MQTT client to "); Serial.println(server); while
        (!client.connect(clientId, authMethod, token)) {
            Serial.print(".");
            delay(500);
        }
        initManagedDevice();
        Serial.println();
    }
}

void initManagedDevice() {
    if (client.subscribe(topic)) {
        // Serial.println(client.subscribe(topic));
        Serial.println("IBM subscribe to cmd OK");
    } else {
        Serial.println("subscribe to cmd FAILED");
    }
}
void publishData()
{
    digitalWrite(trigpin,LOW);
    digitalWrite(trigpin,HIGH);
    delayMicroseconds(10);
    digitalWrite(trigpin,LOW);
    duration=pulseIn(echopin,HIGH);
    dist=duration*speed/2;
    if(dist<100){
        String payload = "{\"Alert Distance\":";
        payload += dist;
        payload += "}";
        Serial.print("\n");
        Serial.print("Sending payload: ");
        Serial.println(payload);
        if (client.publish(publishTopic, (char*) payload.c_str()))
            { Serial.println("Publish OK");
        }


    }
        if(dist>100){
        String payload = "{\"Distance\":";
        payload += dist;
        payload += "}";

        Serial.print("\n");
        Serial.print("Sending payload: ");
        Serial.println(payload);
          if(client.publish(publishTopic, (char*) payload.c_str())) {
            Serial.println("Publish OK");
        }else {
            Serial.println("Publish FAILED");
        }
    }
    }
```

OUTPUT:- i) When distance greater than 100 cm



```
{
  "version": 1,
  "author": "Keerthika J",
  "editor": "wokwi",
  "parts": [
    { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": 92.67, "left": 45.3
    {
      "type": "wokwi-led",
      "id": "led1",
      "top": 194.54,
      "left": 309.26,
      "attrs": { "color": "red" }
    },
    {
      "type": "wokwi-hc-sr04",
      "id": "ultrasonic1",
      "top": 60.71,
      "left": 185.64,
      "attrs": { "distance": "139" }
    },
    {
      "type": "wokwi-resistor",
      "id": "r1",
      "top": 269.89,
      "left": 260.39,
      "attrs": { "value": "100" }
    }
  ],
  "connections": [
    [ "esp:TX0", "$serialMonitor:RX", "", [] ],
```

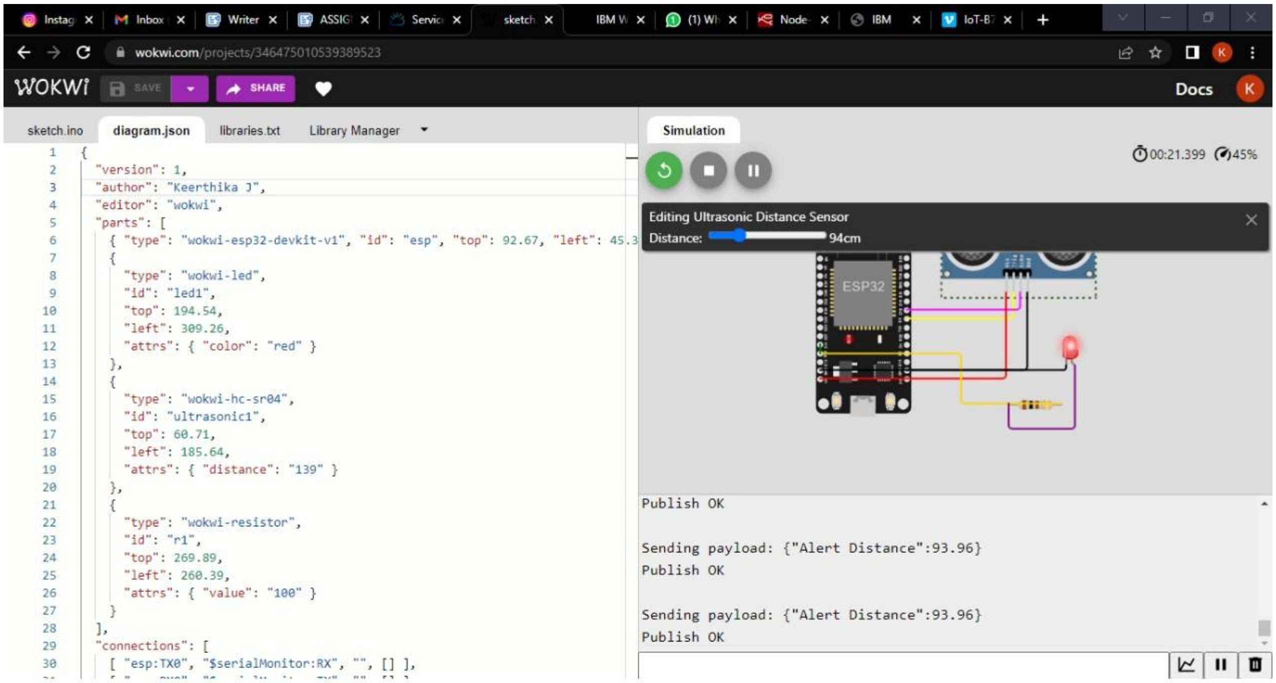
```
Publish OK

Sending payload: {"Distance":138.96}
Publish OK

Sending payload: {"Distance":138.98}
Publish OK
```



The recent events listed show the live stream of data that is coming and going from this device.

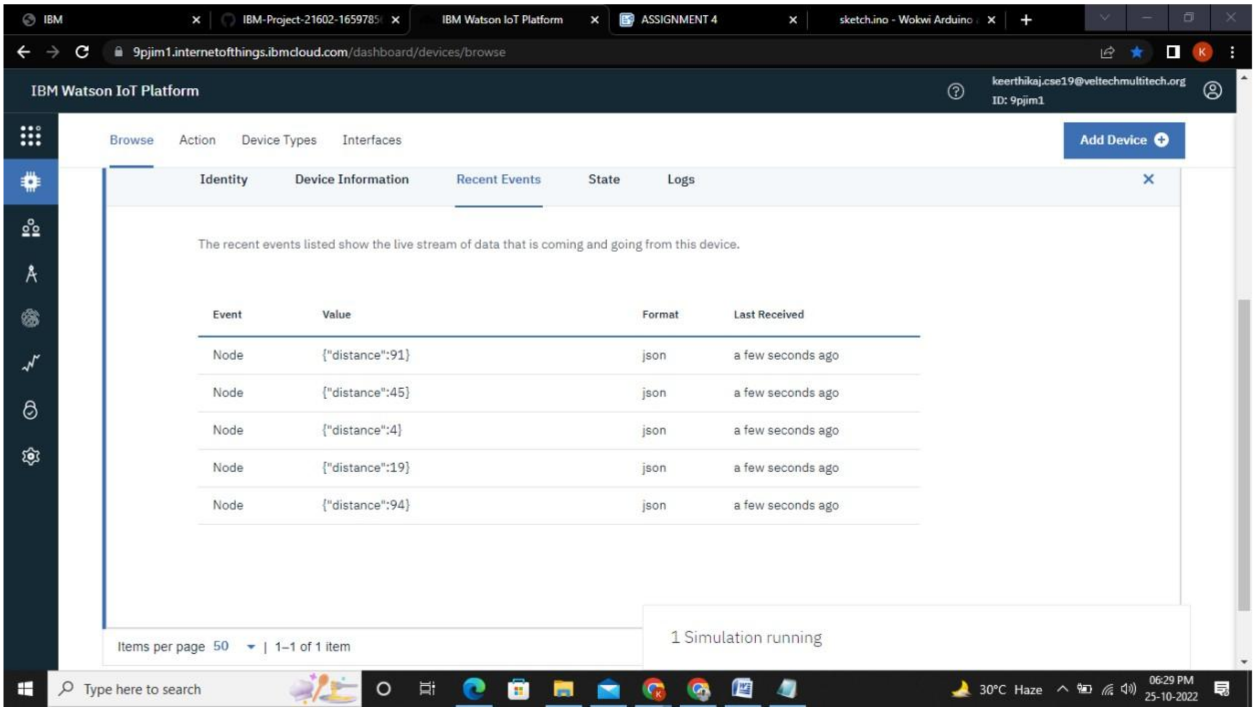
Event	Value	Format	Last Received
Node	{"distance":17}	json	a few seconds ago
Node	{"distance":153}	json	a minute ago
Node	{"distance":154}	json	a minute ago
Node	{"distance":155}	json	a minute ago
Node	{"distance":170}	json	a minute ago

ii) When distance less than 100 cms.



```
{
  "version": 1,
  "author": "Keerthika J",
  "editor": "wokwi",
  "parts": [
    { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": 92.67, "left": 45.3
    {
      "type": "wokwi-led",
      "id": "led1",
      "top": 194.54,
      "left": 309.26,
      "attrs": { "color": "red" }
    },
    {
      "type": "wokwi-hc-sr04",
      "id": "ultrasonic1",
      "top": 60.71,
      "left": 185.64,
      "attrs": { "distance": "139" }
    },
    {
      "type": "wokwi-resistor",
      "id": "r1",
      "top": 269.89,
      "left": 260.39,
      "attrs": { "value": "100" }
    }
  ],
  "connections": [
    [ "esp:TX0", "$serialMonitor:RX", "", [] ],
```

```
Publish OK

Sending payload: {"Alert Distance":93.96}
Publish OK

Sending payload: {"Alert Distance":93.96}
Publish OK
```



The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
Node	{"distance":91}	json	a few seconds ago
Node	{"distance":45}	json	a few seconds ago
Node	{"distance":4}	json	a few seconds ago
Node	{"distance":19}	json	a few seconds ago
Node	{"distance":94}	json	a few seconds ago