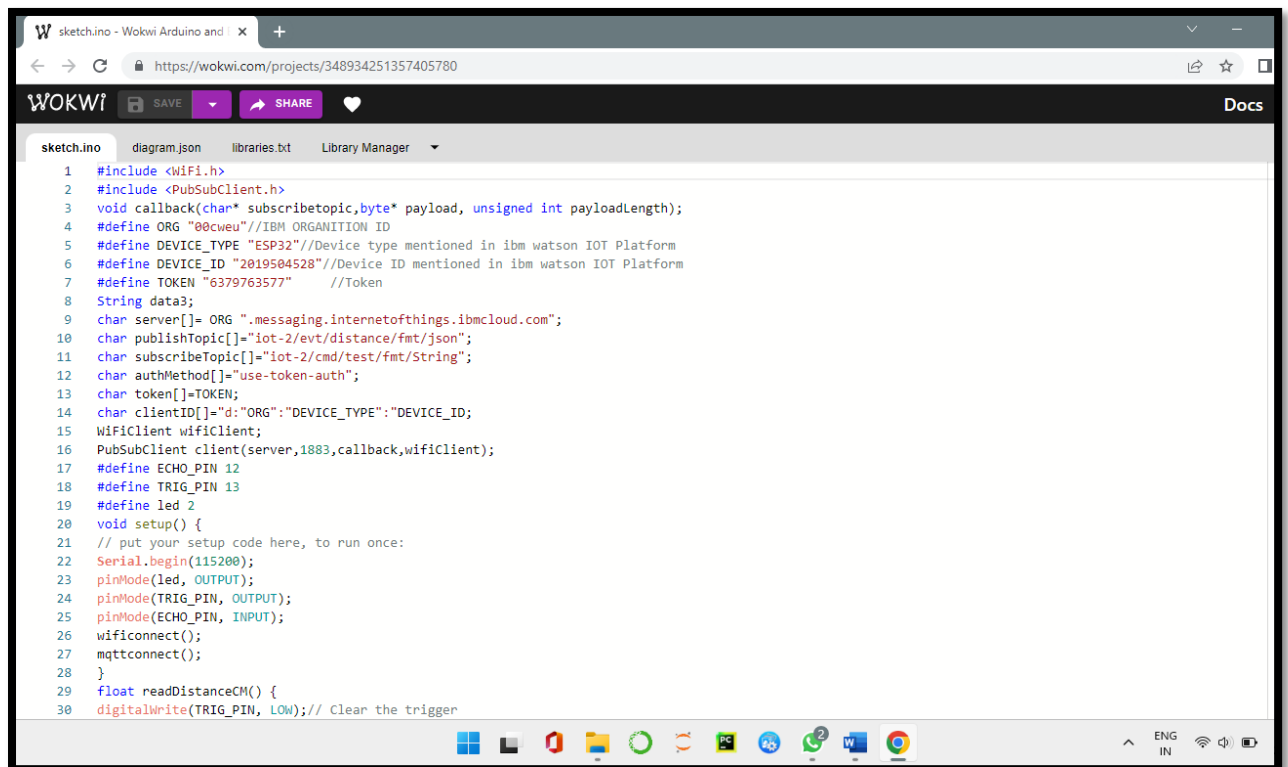# Assignment-4

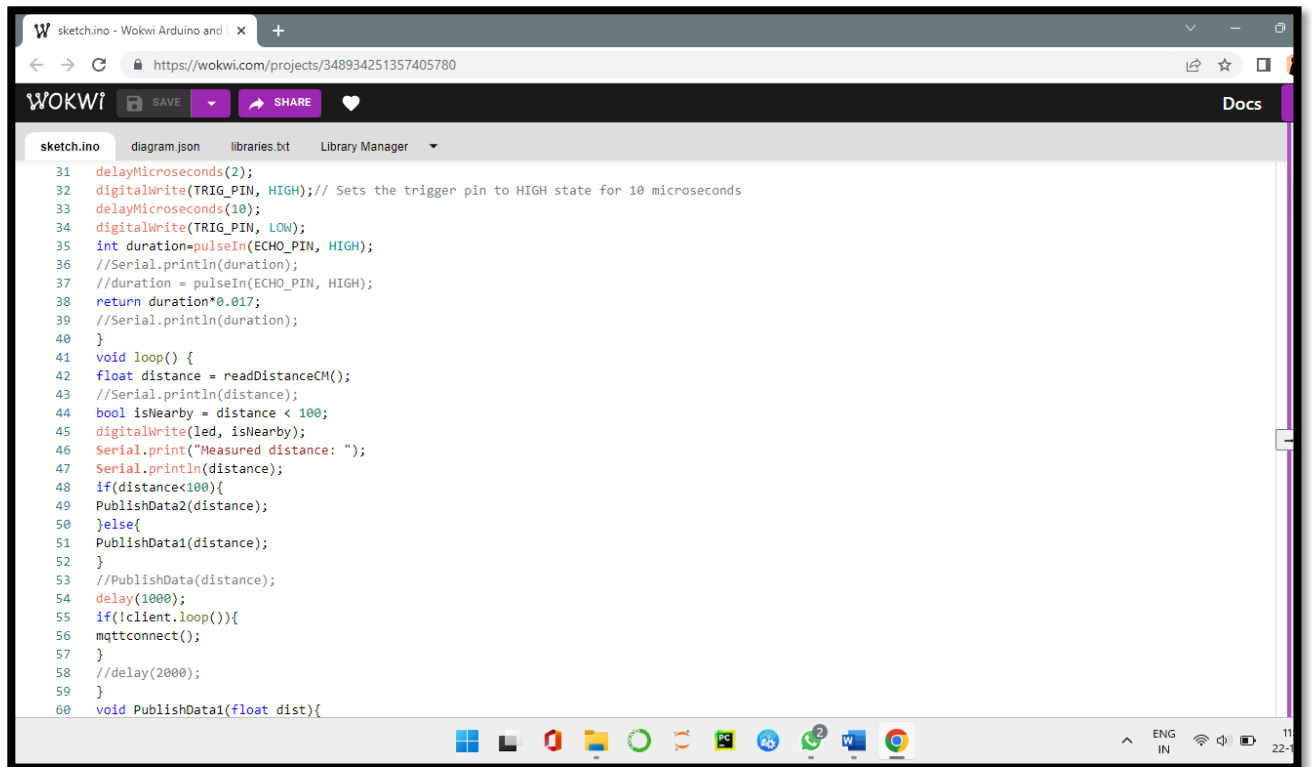| Name	HARRIET PERDITA.P
Roll Number	2019504528

Problem Statement:

Write code and connections in Wokwi for ultrasonic sensor. Whenever distance is less than 100 cm send "alert" to IBM cloud and display in device recent events.
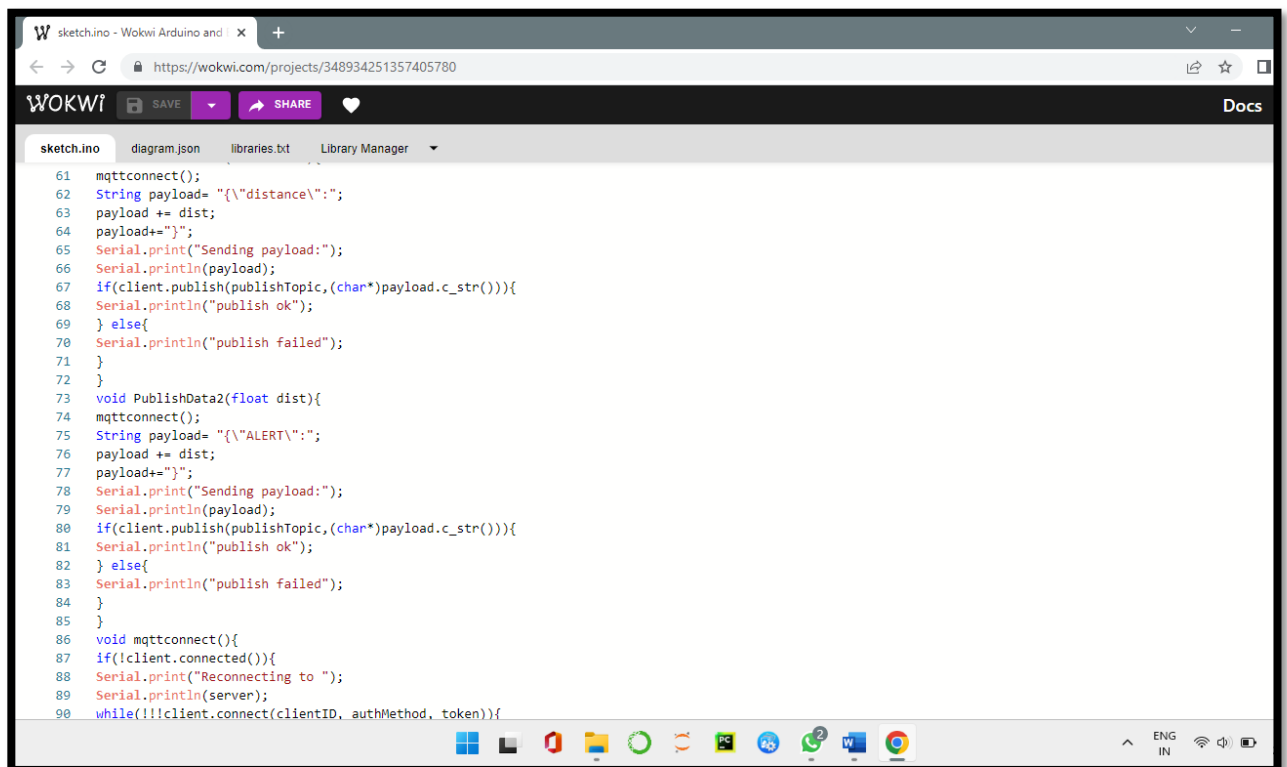
Source Code:


```
1   #include <WiFi.h>
2   #include <PubSubClient.h>
3   void callback(char* subscribetopic,byte* payload, unsigned int payloadLength);
4   #define ORG "00cweu"//IBM ORGANITION ID
5   #define DEVICE_TYPE "ESP32"//Device type mentioned in ibm watson IOT Platform
6   #define DEVICE_ID "2019504528"//Device ID mentioned in ibm watson IOT Platform
7   #define TOKEN "6379763577"     //Token
8   String data3;
9   char server[]= ORG ".messaging.internetofthings.ibmcloud.com";
10  char publishTopic[]="iot-2/evt/distance/fmt/json";
11  char subscribeTopic[]="iot-2/cmd/test/fmt/String";
12  char authMethod[]="use-token-auth";
13  char token[]=TOKEN;
14  char clientID[]="d:"ORG":"DEVICE_TYPE":"DEVICE_ID;
15  WiFiClient wifiClient;
16  PubSubClient client(server,1883,callback,wifiClient);
17  #define ECHO_PIN 12
18  #define TRIG_PIN 13
19  #define led 2
20  void setup() {
21  // put your setup code here, to run once:
22  Serial.begin(115200);
23  pinMode(led, OUTPUT);
24  pinMode(TRIG_PIN, OUTPUT);
25  pinMode(ECHO_PIN, INPUT);
26  wificonnect();
27  mqttconnect();
28  }
29  float readDistanceCM() {
30  digitalWrite(TRIG_PIN, LOW);// Clear the trigger
```

```
31      delayMicroseconds(2);
32      digitalWrite(TRIG_PIN, HIGH);// Sets the trigger pin to HIGH state for 10 microseconds
33      delayMicroseconds(10);
34      digitalWrite(TRIG_PIN, LOW);
35      int duration=pulseIn(ECHO_PIN, HIGH);
36      //Serial.println(duration);
37      //duration = pulseIn(ECHO_PIN, HIGH);
38      return duration*0.017;
39      //Serial.println(duration);
40      }
41      void loop() {
42      float distance = readDistanceCM();
43      //Serial.println(distance);
44      bool isNearby = distance < 100;
45      digitalWrite(led, isNearby);
46      Serial.print("Measured distance: ");
47      Serial.println(distance);
48      if(distance<100){
49      PublishData2(distance);
50      }else{
51      PublishData1(distance);
52      }
53      //PublishData(distance);
54      delay(1000);
55      if(!client.loop()){
56      mqttconnect();
57      }
58      //delay(2000);
59      }
60      void PublishData1(float dist){
```

```
61      mqttconnect();
62      String payload= "{\"distance\":";
63      payload += dist;
64      payload+="}";
65      Serial.print("Sending payload:");
66      Serial.println(payload);
67      if(client.publish(publishTopic,(char*)payload.c_str())){
68      Serial.println("publish ok");
69      } else{
70      Serial.println("publish failed");
71      }
72      }
73      void PublishData2(float dist){
74      mqttconnect();
75      String payload= "{\"ALERT\":";
76      payload += dist;
77      payload+="}";
78      Serial.print("Sending payload:");
79      Serial.println(payload);
80      if(client.publish(publishTopic,(char*)payload.c_str())){
81      Serial.println("publish ok");
82      } else{
83      Serial.println("publish failed");
84      }
85      }
86      void mqttconnect(){
87      if(!client.connected()){
88      Serial.print("Reconnecting to ");
89      Serial.println(server);
90      while(!!!client.connect(clientID, authMethod, token)){
```

```
 91   Serial.print(".");
 92   delay(500);
 93   }
 94   initManagedDevice();
 95   Serial.println();
 96   }
 97   }
 98   void wificonnect(){
 99   Serial.println();
100   Serial.print("Connecting to");
101   WiFi.begin("Wokwi-GUEST","",6);
102   while(WiFi.status()!=WL_CONNECTED){
103   delay(500);
104   Serial.print(".");
105   }
106   Serial.println("");
107   Serial.println("WIFI CONNECTED");
108   Serial.println("IP address:");
109   Serial.println(WiFi.localIP());
110   }
111   void initManagedDevice(){
112   if(client.subscribe(subscribeTopic)){
113   Serial.println((subscribeTopic));
114   Serial.println("subscribe to cmd ok");
115   }else{
116   Serial.println("subscribe to cmd failed");
117   }
118   }
119   void callback(char* subscribeTopic, byte* payload, unsigned int
120   payloadLength){
```

```
121   Serial.print("callback invoked for topic:");
122   Serial.println(subscribeTopic);
123   for(int i=0; i<payloadLength; i++){
124   data3 += (char)payload[i];
125   }
126   Serial.println("data:"+ data3);
127   if(data3=="lighton"){
128   Serial.println(data3);
129   digitalWrite(led,HIGH);
130   }else{
131   Serial.println(data3);
132   digitalWrite(led,LOW);
133   }
134   data3="";
135   }
```

Wokwi Link:

https://wokwi.com/projects/348934251357405780

Output:

Normal Case:

Alert Case:

IBM Cloud Storage: