Student Name	M.Rishnika
Student Roll Number	720319106017

Question-1:

Solution:

Link: **https://wokwi.com/projects/3441982264521529**

```
#include <WiFi.h>
#include <PubSubClient.h>
WiFiClient wifiClient;

#define trigpin     18
#define echopin     5

String data3;

#define ORG "q467tn"//IBM ORGANITION ID
#define DEVICE_TYPE "child"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "123454"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "93457641"

#define speed 0.034
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Data/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, wifiClient);
void publishData();

String command;
String data="";
long duration;
float dist;

void setup()
{
  Serial.begin(115200);
  wifiConnect();
  pinMode(trigpin, OUTPUT);
  pinMode(echopin, INPUT);

 mqttConnect();
```

```cpp
}

void wifiConnect() {
  Serial.print("Connecting to "); Serial.print("Wifi");
  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 loop()
{

    int pulseWidth = 0;
    digitalWrite(trigpin, HIGH);
    delayMicroseconds(100);
    digitalWrite(trigpin, LOW);
    pulseWidth = pulseIn(echopin, HIGH);
    Serial.print("AlertDistance: ");
    Serial.println(pulseWidth/58);

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

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()
{
```

```cpp
digitalWrite(trigpin,LOW);
digitalWrite(trigpin,HIGH);
delayMicroseconds(10);
digitalWrite(trigpin,LOW);
duration=pulseIn(echopin,HIGH);
dist=duration*speed/2;
if(dist<100){
  String payload = "{\"Normal 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>101 && dist<111){
  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("Warning crosses 110cm -- it automaticaly of the loop");

  }else {
    Serial.println("Publish FAILED");
  }

}


}
void callback(char* subscribeTopic, byte* payload, unsigned int payloadLength){
Serial.print("callback invoked for topic:");
Serial.println(subscribeTopic);
for(int i=0; i<payloadLength; i++){
  dist += (char)payload[i];
}
Serial.println("data:"+ data3);
if(data3=="lighton"){
  Serial.println(data3);

}
data3="";
}
```
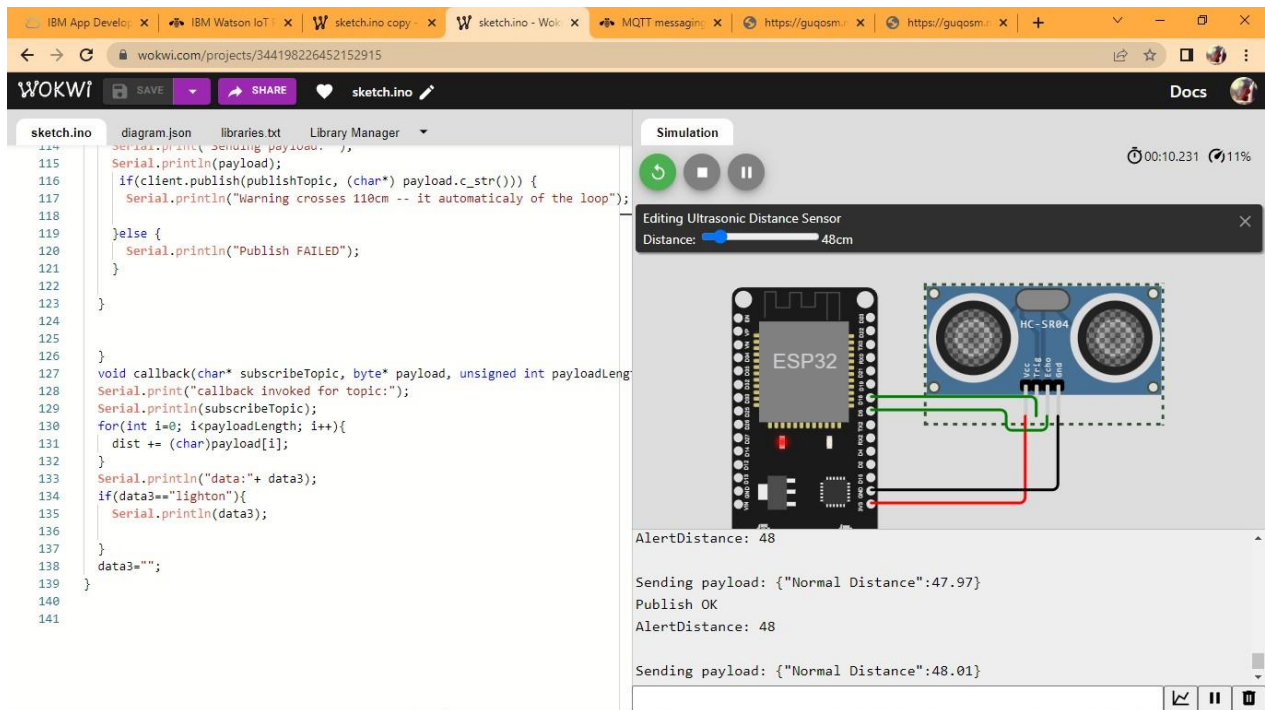
Output:

https://guqosm.internetofthings.ibmcloud.com/dashboard/devices/browse