Assignment -4
Python Programming

Student Name	Anitha C
Student Roll Number	720319106002

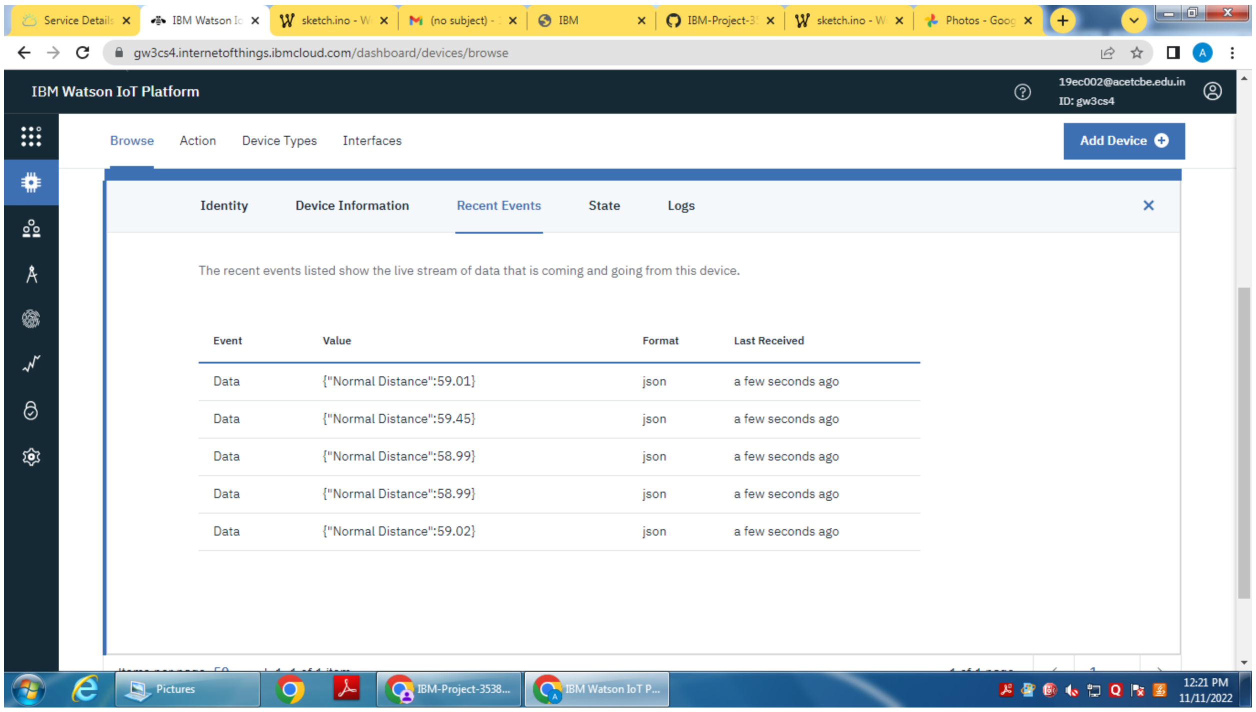
Question-1:

Solution:

Link : https://wokwi.com/projects/348011432680358484

```
#include<WiFi.h>
#include<PubSubClient.h>
WiFiClientwifiClient;

#definetrigpin    18
#defineechopin    5

String data3;

#define ORG "gw3cs4"//IBM ORGANITION ID
#define DEVICE_TYPE "smart"//Device type mentioned in ibmwatson IOT Platform
#define DEVICE_ID "123547"//Device ID mentioned in ibmwatson IOT Platform
#define TOKEN "12345678"

#define speed 0.034
charserver[] = ORG ".messaging.internetofthings.ibmcloud.com";
charpublishTopic[] = "iot-2/evt/Data/fmt/json";
chartopic[] = "iot-2/cmd/home/fmt/String";
charauthMethod[] = "use-token-auth";
chartoken[] = TOKEN;
charclientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
PubSubClientclient(server, 1883, wifiClient);
voidpublishData();

String command;
String data="";
long duration;
```

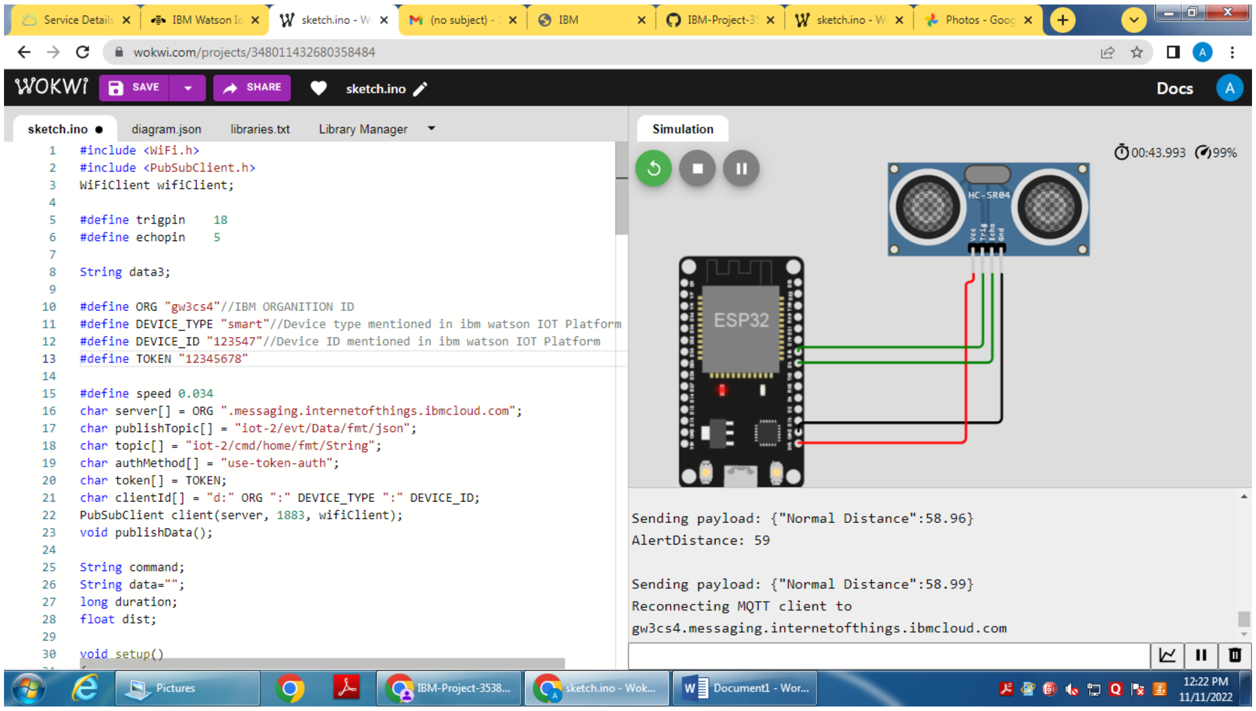
```
floatdist;

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

 mqttConnect();
}

voidwifiConnect() {
  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());
}
voidloop()
{

    intpulseWidth = 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();
  }
}

voidmqttConnect() {
  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();
  }
}

voidinitManagedDevice() {
  if (client.subscribe(topic)) {
    // Serial.println(client.subscribe(topic));
    Serial.println("IBM subscribe to cmd OK");
  } else {
    Serial.println("subscribe to cmd FAILED");
  }
}
voidpublishData()
{
  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");
    }

  }


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

  }
  data3="";
}
```



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

Event	Value	Format	Last Received
Data	{"Normal Distance":59.01}	json	a few seconds ago
Data	{"Normal Distance":59.45}	json	a few seconds ago
Data	{"Normal Distance":58.99}	json	a few seconds ago
Data	{"Normal Distance":58.99}	json	a few seconds ago
Data	{"Normal Distance":59.02}	json	a few seconds ago

Output : https://gw3cs4.internetofthings.ibmcloud.com/dashboard/devices/browse



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

#define trigpin    18
#define echopin    5

String data3;

#define ORG "gw3cs4"//IBM ORGANITION ID
#define DEVICE_TYPE "smart"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "123547"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "12345678"

#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()
```

```
Sending payload: {"Normal Distance":58.96}
AlertDistance: 59

Sending payload: {"Normal Distance":58.99}
Reconnecting MQTT client to
gw3cs4.messaging.internetofthings.ibmcloud.com
```