Sprint 3

Team ID	PNT2022TMID34792
Project Name	Personal Assistance for Senior Who Are S elf-Reliant

Iot device program :

```
#include <WiFi.h>//library for wifi

#include <PubSubClient.h>//library for MQtt

#include <LiquidCrystal_I2C.h>

#include "DHT.h"// Library for dht11

#define DHTPIN 15 // what pin we're connected to

#define DHTTYPE DHT11 // define type of sensor DHT 11

#define Buzzer 2

DHT dht (DHTPIN, DHTTYPE);// creating the instance by passing pin and typr of dht

connected

void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);



//-------credentials of IBM Accounts------


#define ORG "1l6lvg"//IBM ORGANITION ID

#define DEVICE_TYPE "nodeMCU"//Device type mentioned in ibm watson IOT Platform

#define DEVICE_ID "12345"//Device ID mentioned in ibm watson IOT Platform
```

```
#define TOKEN "?nUW@lkY)OglhHt)i6" //Token

String data3="";



//-------- Customise the above values --------

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name char

publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of event perform and format

in which data to be send

char subscribetopic[] = "iot-2/cmd/command/fmt/String";// cmd REPRESENT command

type

AND COMMAND IS TEST OF FORMAT STRING

char authMethod[] = "use-token-auth";// authentication method char

token[] = TOKEN;

char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id LiquidCrystal_I2C

lcd(0x27,16,2);



//---------------------------------------

WiFiClient wifiClient; // creating the instance for wificlient

PubSubClient client(server, 1883, callback ,wifiClient); //calling the predefined client id by
```

```
passing parameter like server id,portand wificredential void setup()// configureing
the ESP32
{
 Serial.begin(115200); dht.begin();
 pinMode(Buzzer,OUTPUT);
 delay(10); Serial.println();
wificonnect();
mqttconnect();
}

void loop()// Recursive Function
{ if
(!client.loop()) {
mqttconnect();
 } }
void PublishData(float temp, float humid) {
mqttconnect();//function call for connecting to ibm

} void mqttconnect() {
if (!client.connected())
{
 Serial.print("Reconnecting client to ");
Serial.println(server); while (!!!client.connect(clientId,
```

```
authMethod, token)) { Serial.print(".");
delay(500);
 }
 initManagedDevice();
 Serial.println();
 } }
void wificonnect() //function defination for wificonnect
{
 Serial.println();
 Serial.print("Connecting to ");

 WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to establish the
connection while (WiFi.status() != WL_CONNECTED) { delay(500);
 Serial.print(".");
 }
 Serial.println("");
 Serial.println("WiFi connected");
 Serial.println("IP address: ");
 Serial.println(WiFi.localIP());
}

void initManagedDevice() { if
(client.subscribe(subscribetopic)) {
```

```
 Serial.println((subscribetopic));
 Serial.println("subscribe to cmd OK");
 } else {
 Serial.println("subscribe to cmd FAIBuzzer");
 }
}

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++) {
//Serial.print((char)payload[i]); data3 +=
(char)payload[i];
 }

 Serial.println("Medicine Name: "+ data3);
if(data3 != "")
 { lcd.init(); lcd.print(data3);
digitalWrite(Buzzer,HIGH);
delay(20000);
digitalWrite(Buzzer,LOW);
 } else
```

```
 {
digitalWrite(Buzzer,LOW);
 }
data3="";
}
```