FINAL DELIVERY

Project Title:	Personal Assistant for Senior Citizen
Team ID:	PNT2022TMID50622

Code to subscribe topic in IBM watson:

```
#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 LED 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 "116lvg"//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
```

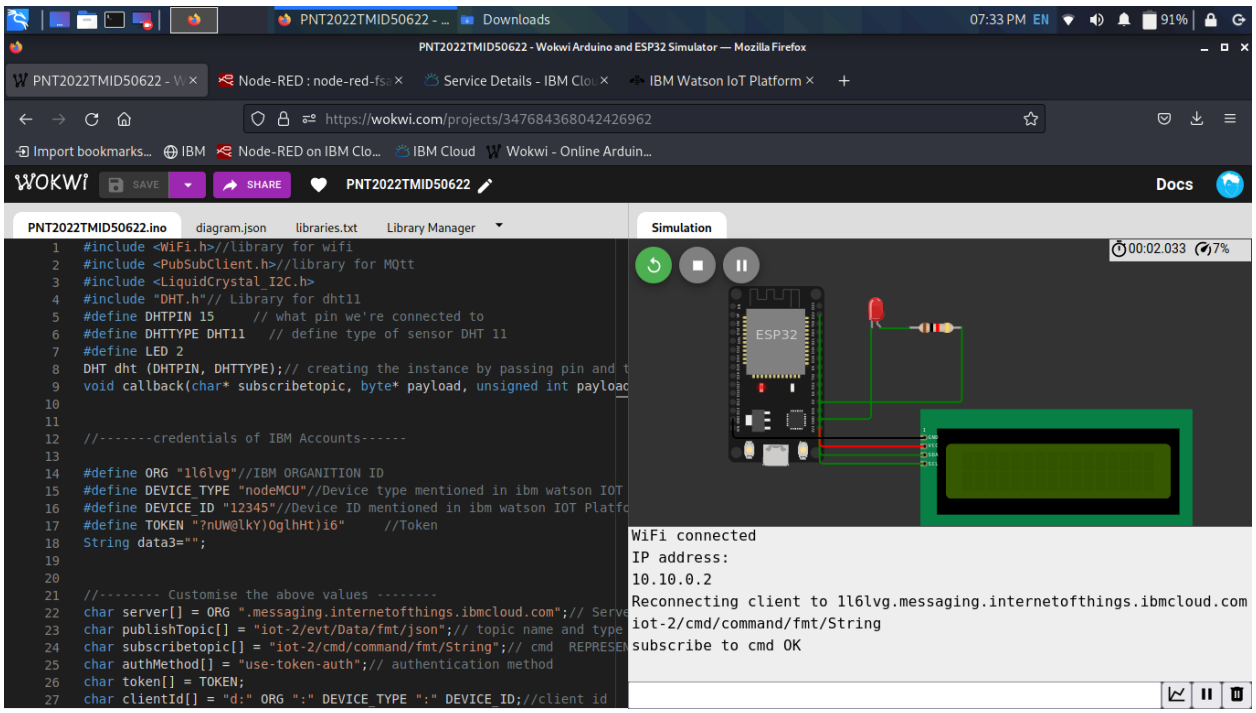
```
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id
LiquidCrystal_I2C lcd(0x27,16,2);

//-------------------------------------
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(LED,OUTPUT);
 delay(10);
 Serial.println();
 wificonnect();
 mqttconnect();
}

void loop()// Recursive Function
{
 if (!client.loop()) {
   mqttconnect();
 }
}

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 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++) {
   //Serial.print((char)payload[i]);
   data3 += (char)payload[i];
 }
  Serial.println("Medicine Name: "+ data3);
 if(data3 != "")
 {
   lcd.init();
   lcd.print(data3);
```

```
digitalWrite(LED,HIGH);
delay(20000);
digitalWrite(LED,LOW);
 }

 else
 {
digitalWrite(LED,LOW);

 }
data3="";

 }
```

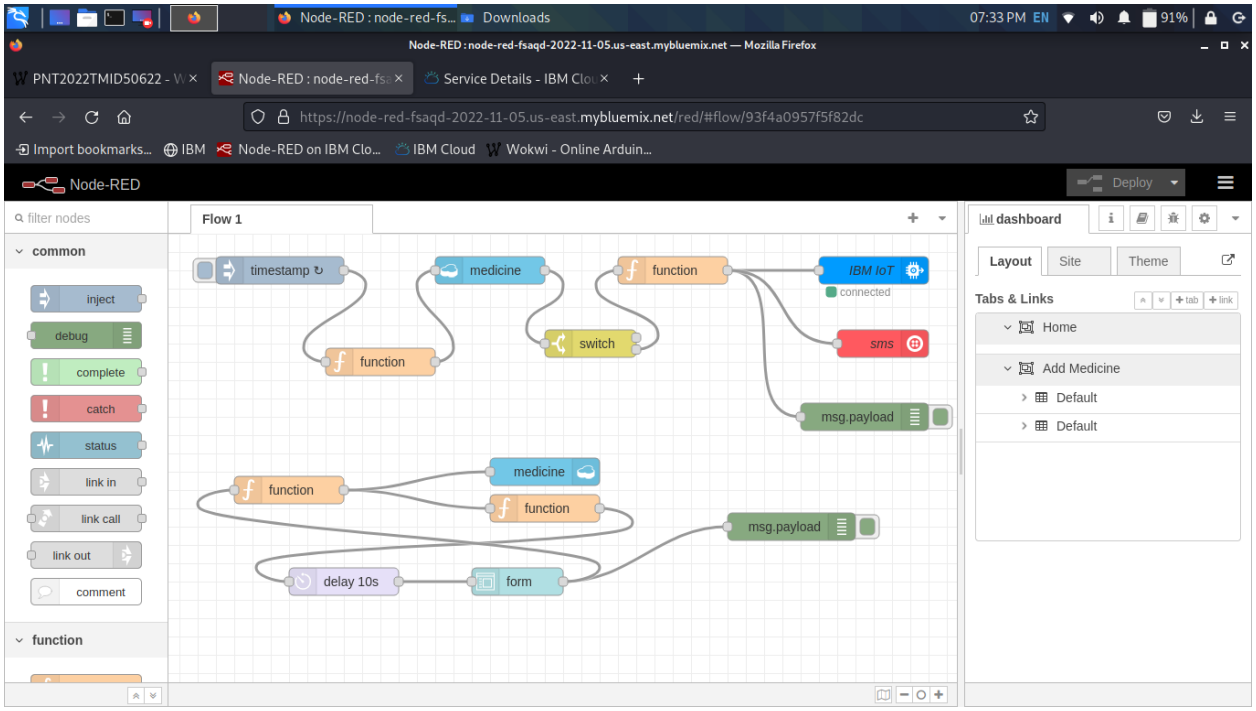
Simulation on Wokwi before medicine added :

Wokwi url: https://wokwi.com/projects/347684368042426962

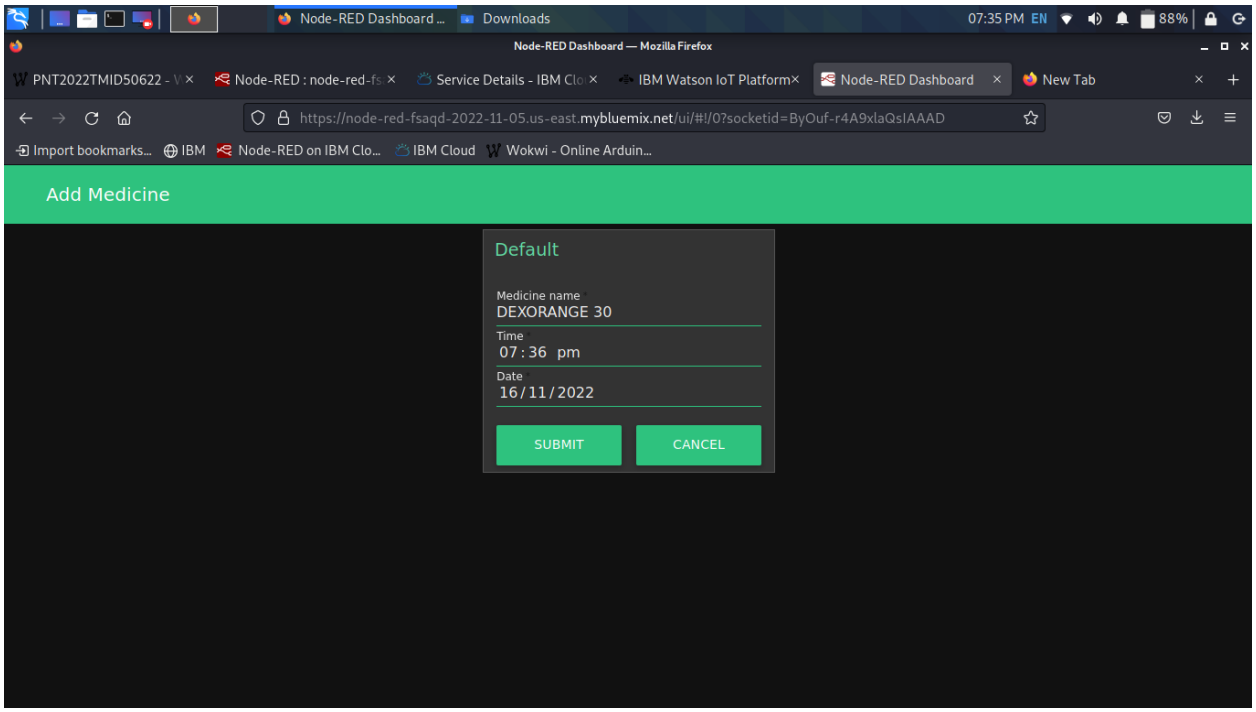


Node-red configuration:

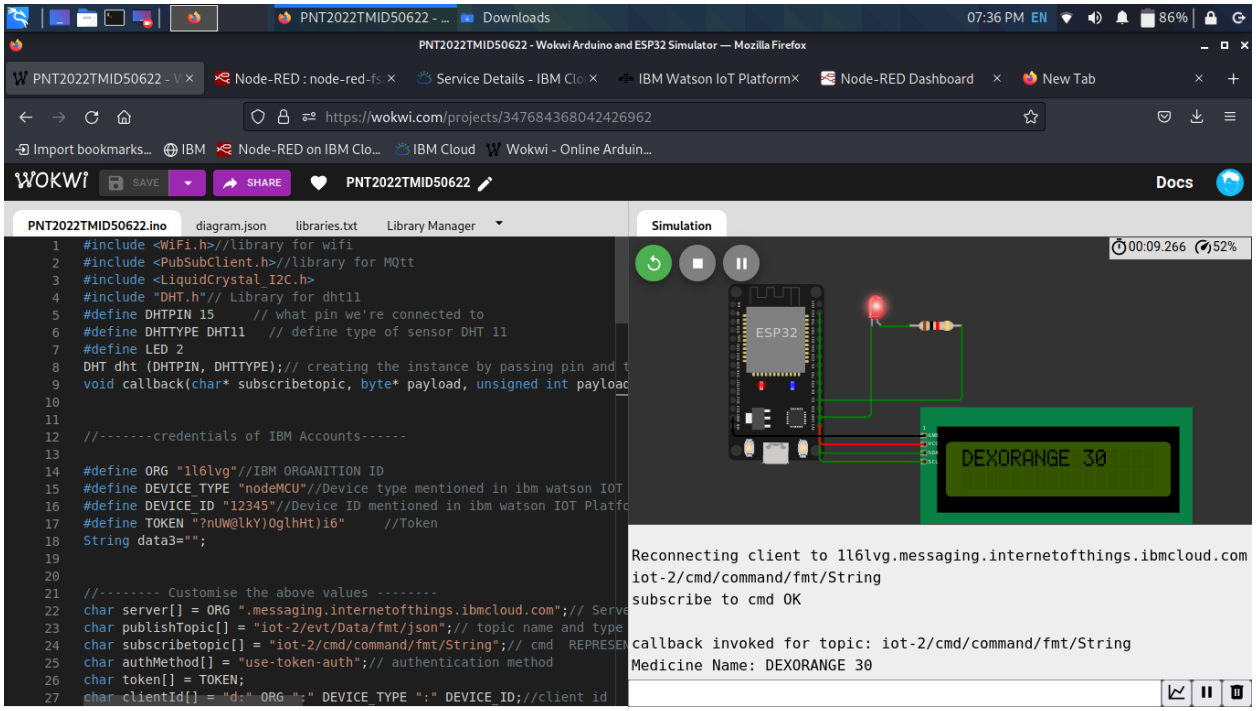
Node-red url: https://node-red-fsaqd-2022-11-05.us-east.mybluemix.net/red/



Added Medicine details in node-red form:



Simulation on wokwi after medicine added:



Medicine details added to the IBM cloud: