PROJECT DEVELOPMENT PHASE

SPRINT 3

Hardware Implementation

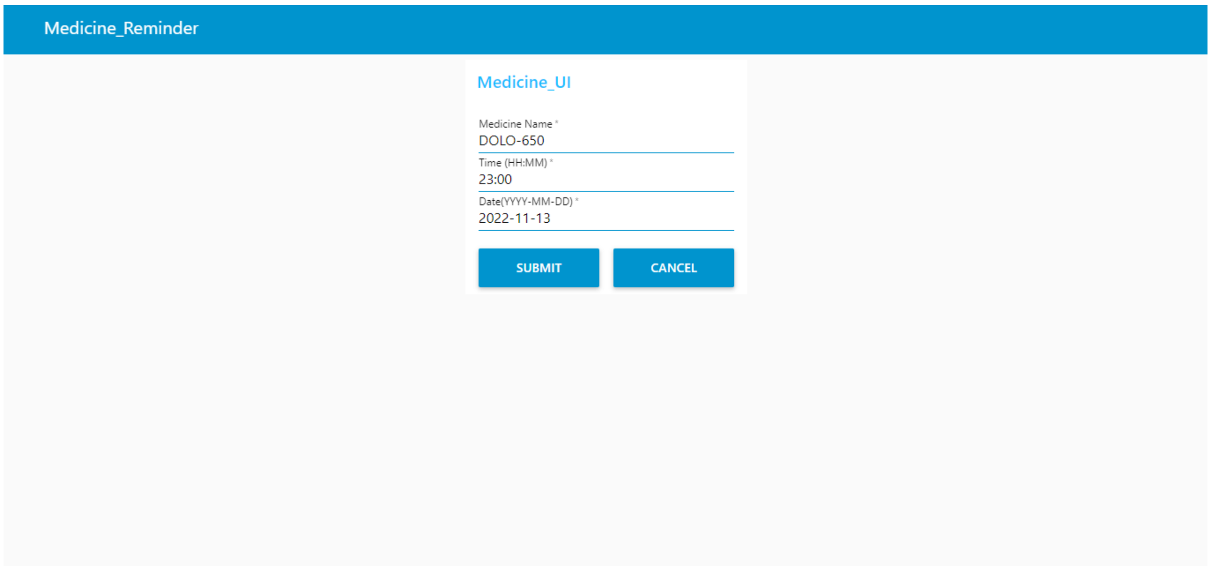
Date	12 November 2022
Team ID	PNT2022TMID18885
Project Name	Project - Personal Assistance for Seniors Who Are Self-Reliant

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	**Registration:** Creation of IBM serviceslike NodeRED, CloudantDB, TTS Service and design of IoT system	USN-1	As a user,I should login into my IBM Cloud account.	2	High	Anne Angelina J, Kawin M
Sprint-2	**Web UI:** Creating web UI using node-red and connect it toIBM Cloudant db	USN-2	As a user,I should be able to feed the medicine name and intake time in the web UI	2	High	Akshayasri S, Bhavani R K
Sprint-3	**Hardware implementation:** Developing Python code to retrieve data from cloudant db to send that data to IoT device at the appropriate time	USN-3	As a user, I should be able to send the medicine name to the IoT device at the scheduled time	2	High	Akshayasri S, Kawin M
Sprint-4	**Software implementation:** Converting the data received from cloud asvoice using IBM Text toSpeech service	USN-4	As a user, I must be able hear the medicine name which is to be taken at the appropriatetime	2	High	Anne Angelina J, Bhavani R K

Objective:

- Developing code to retrieve data from cloudant db to send that data to IoT device at the appropriate time.

I. Scheduling medicine name and intake time:

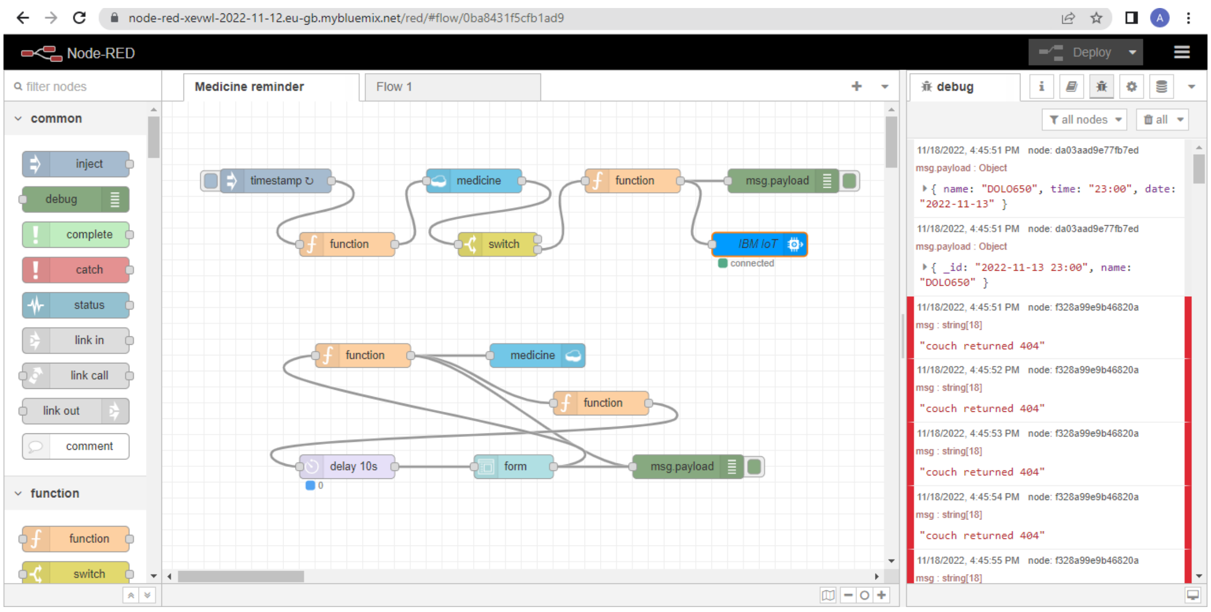

Medicine_Reminder

Medicine_UI

Medicine Name *	DOLO-650
Time (HH:MM) *	23:00
Date(YYYY-MM-DD) *	2022-11-13

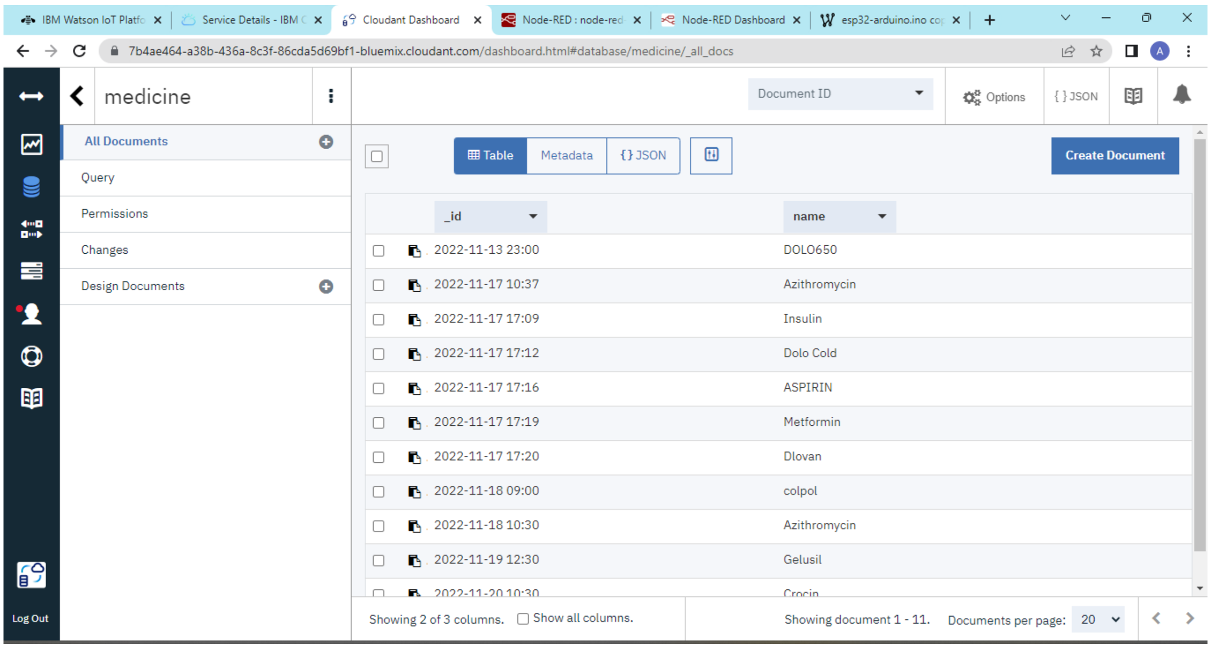
SUBMIT CANCEL

II. Medicine details displayed in Node-Red debug window:



```
11/18/2022, 4:45:51 PM   node: da03aad9e77fb7ed
msg.payload : Object
▸{ name: "DOLO650", time: "23:00", date:
"2022-11-13" }

11/18/2022, 4:45:51 PM   node: da03aad9e77fb7ed
msg.payload : Object
▸{ _id: "2022-11-13 23:00", name:
"DOLO650" }

11/18/2022, 4:45:51 PM   node: f328a99e9b46820a
msg : string[18]
"couch returned 404"

11/18/2022, 4:45:52 PM   node: f328a99e9b46820a
msg : string[18]
"couch returned 404"

11/18/2022, 4:45:53 PM   node: f328a99e9b46820a
msg : string[18]
"couch returned 404"

11/18/2022, 4:45:54 PM   node: f328a99e9b46820a
msg : string[18]
"couch returned 404"

11/18/2022, 4:45:55 PM   node: f328a99e9b46820a
msg : string[18]
```

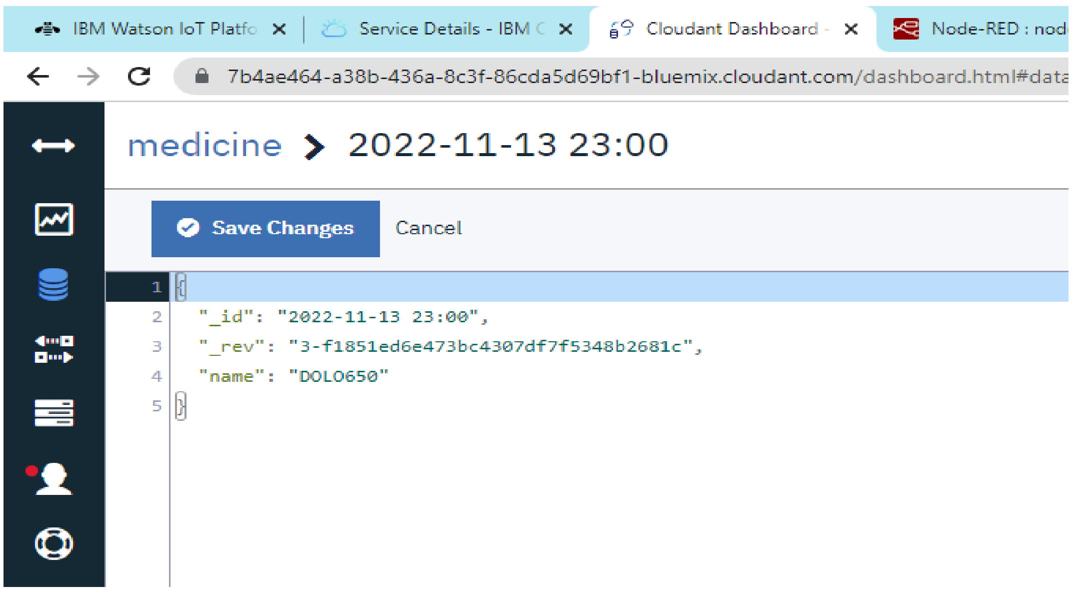
III. Medicine details pushed and displayed in IBM Cloudant db:



	_id	name
	2022-11-13 23:00	DOLO650
	2022-11-17 10:37	Azithromycin
	2022-11-17 17:09	Insulin
	2022-11-17 17:12	Dolo Cold
	2022-11-17 17:16	ASPIRIN
	2022-11-17 17:19	Metformin
	2022-11-17 17:20	Dlovan
	2022-11-18 09:00	colpol
	2022-11-18 10:30	Azithromycin
	2022-11-19 12:30	Gelusil
	2022-11-20 10:30	Crocin

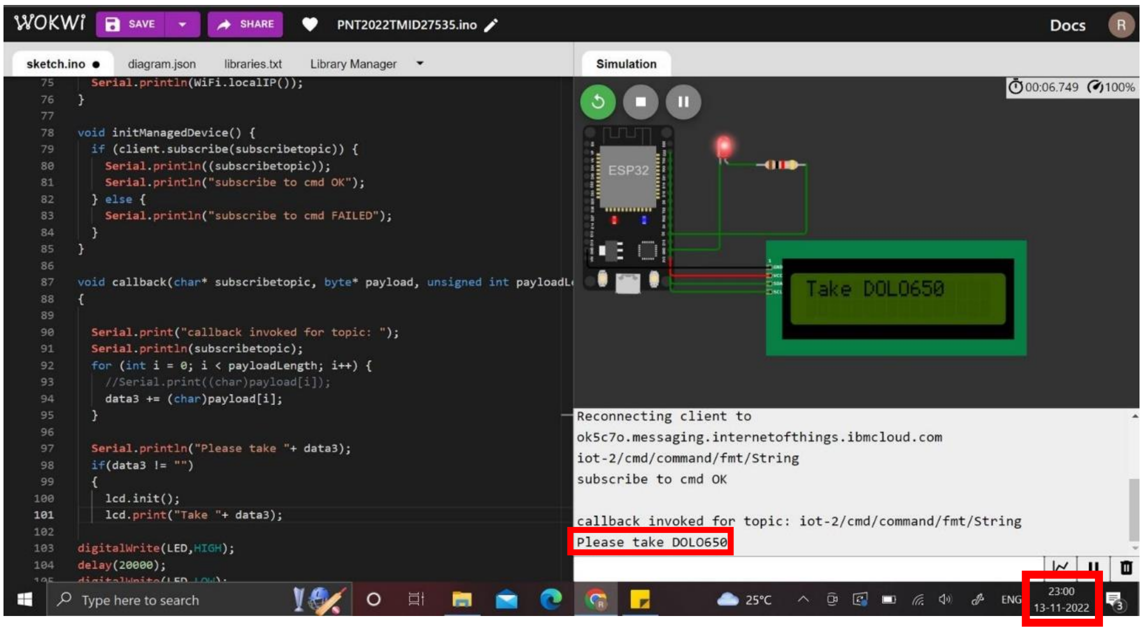
Showing 2 of 3 columns. Show all columns. Showing document 1 - 11. Documents per page: 20

Scheduled DOLO650 medicine to be taken at 23:00 (11:00 PM)



```
{
  "_id": "2022-11-13 23:00",
  "_rev": "3-f1851ed6e473bc4307df7f5348b2681c",
  "name": "DOLO650"
}
```

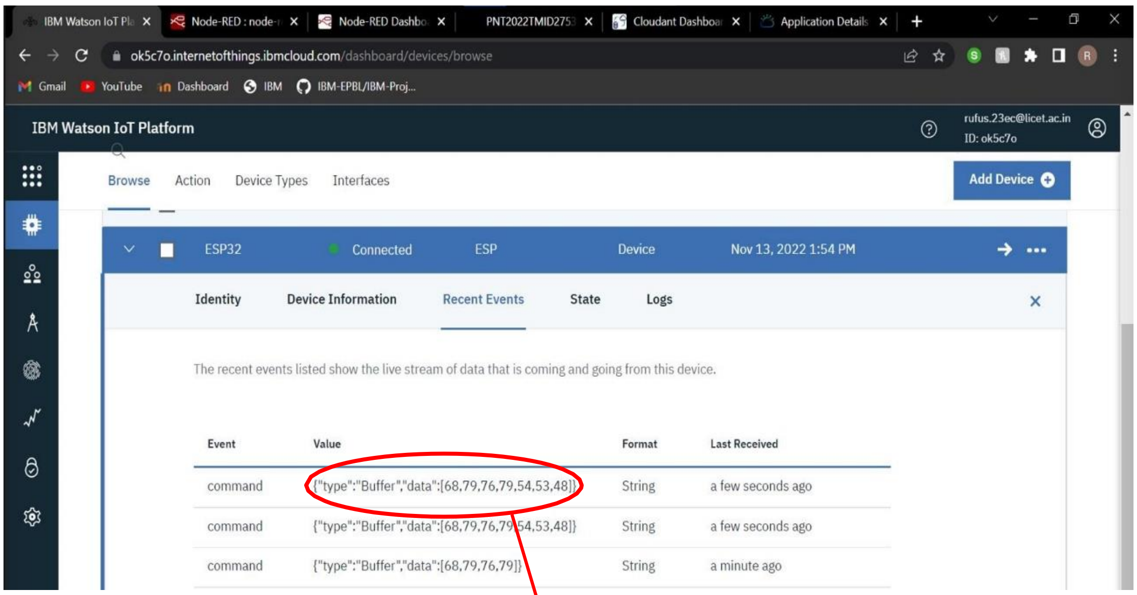
IV. Medicine name sent to ESP32 on the scheduled time 23:00:



```
  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 payloadL
{

  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("Please take "+ data3);
  if(data3 != "")
  {
    lcd.init();
    lcd.print("Take "+ data3);

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

```
Reconnecting client to
ok5c7o.messaging.internetofthings.ibmcloud.com
iot-2/cmd/command/fmt/String
subscribe to cmd OK

callback invoked for topic: iot-2/cmd/command/fmt/String
Please take DOLO650
```

➢ Medicine name sent to ESP32 on the scheduled time 23:00:

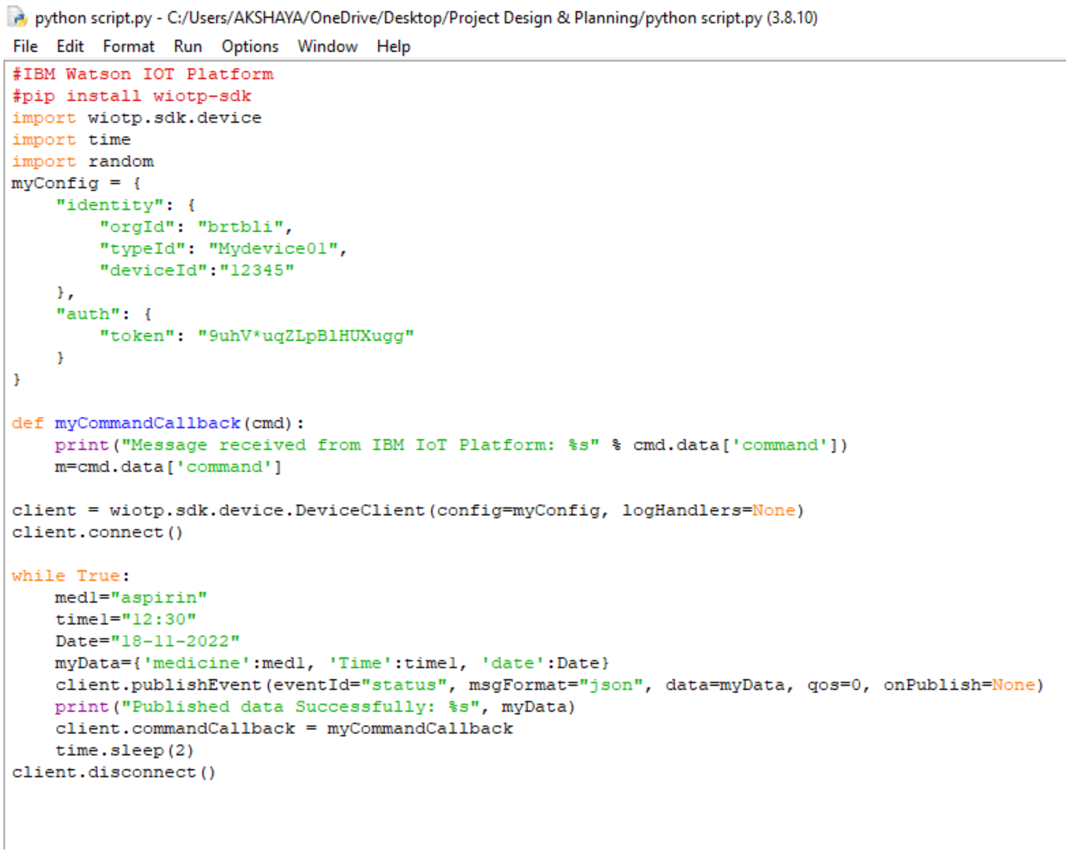


The recent events listed show the live stream of data that is coming and going from this device.

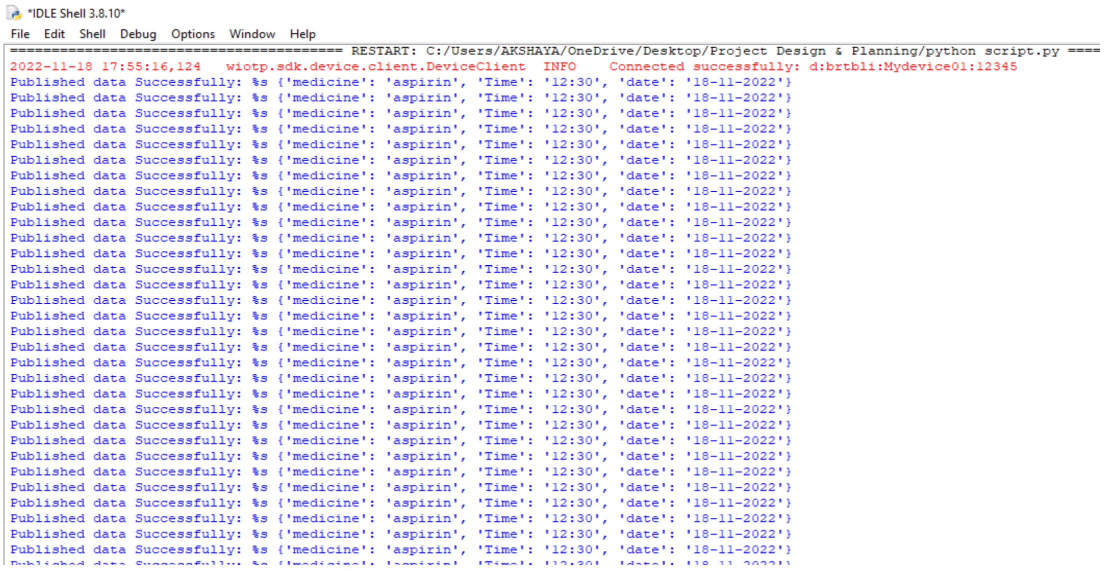
Event	Value	Format	Last Received
command	{"type":"Buffer","data":[68,79,76,79,54,53,48]}	String	a few seconds ago
command	{"type":"Buffer","data":[68,79,76,79,54,53,48]}	String	a few seconds ago
command	{"type":"Buffer","data":[68,79,76,79]}	String	a minute ago

Medicine name (DOLO650) sent to IBM Watson IoT as ASCII values

➢ Python Script

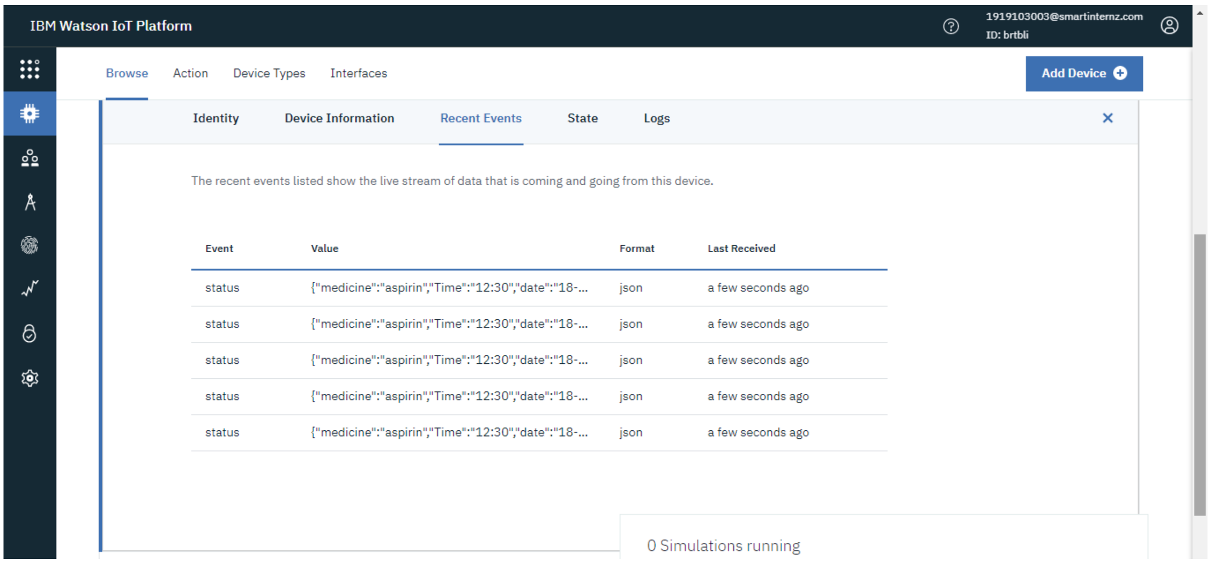

python script.py - C:/Users/AKSHAYA/OneDrive/Desktop/Project Design & Planning/python script.py (3.8.10)
File Edit Format Run Options Window Help

```python
#IBM Watson IOT Platform
#pip install wiotp-sdk
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity": {
        "orgId": "brtbli",
        "typeId": "Mydevice01",
        "deviceId":"12345"
    },
    "auth": {
        "token": "9uhV*uqZLpBlHUXugg"
    }
}

def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
    m=cmd.data['command']

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

while True:
    med1="aspirin"
    time1="12:30"
    Date="18-11-2022"
    myData={'medicine':med1, 'Time':time1, 'date':Date}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
    print("Published data Successfully: %s", myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)
client.disconnect()
```



```
======================================== RESTART: C:/Users/AKSHAYA/OneDrive/Desktop/Project Design & Planning/python script.py ====
2022-11-18 17:55:16,124   wiotp.sdk.device.client.DeviceClient  INFO    Connected successfully: d:brtbli:Mydevice01:12345
Published data Successfully: %s {'medicine': 'aspirin', 'Time': '12:30', 'date': '18-11-2022'}
Published data Successfully: %s {'medicine': 'aspirin', 'Time': '12:30', 'date': '18-11-2022'}
Published data Successfully: %s {'medicine': 'aspirin', 'Time': '12:30', 'date': '18-11-2022'}
Published data Successfully: %s {'medicine': 'aspirin', 'Time': '12:30', 'date': '18-11-2022'}
Published data Successfully: %s {'medicine': 'aspirin', 'Time': '12:30', 'date': '18-11-2022'}
Published data Successfully: %s {'medicine': 'aspirin', 'Time': '12:30', 'date': '18-11-2022'}
Published data Successfully: %s {'medicine': 'aspirin', 'Time': '12:30', 'date': '18-11-2022'}
Published data Successfully: %s {'medicine': 'aspirin', 'Time': '12:30', 'date': '18-11-2022'}
Published data Successfully: %s {'medicine': 'aspirin', 'Time': '12:30', 'date': '18-11-2022'}
Published data Successfully: %s {'medicine': 'aspirin', 'Time': '12:30', 'date': '18-11-2022'}
Published data Successfully: %s {'medicine': 'aspirin', 'Time': '12:30', 'date': '18-11-2022'}
Published data Successfully: %s {'medicine': 'aspirin', 'Time': '12:30', 'date': '18-11-2022'}
Published data Successfully: %s {'medicine': 'aspirin', 'Time': '12:30', 'date': '18-11-2022'}
Published data Successfully: %s {'medicine': 'aspirin', 'Time': '12:30', 'date': '18-11-2022'}
Published data Successfully: %s {'medicine': 'aspirin', 'Time': '12:30', 'date': '18-11-2022'}
Published data Successfully: %s {'medicine': 'aspirin', 'Time': '12:30', 'date': '18-11-2022'}
Published data Successfully: %s {'medicine': 'aspirin', 'Time': '12:30', 'date': '18-11-2022'}
Published data Successfully: %s {'medicine': 'aspirin', 'Time': '12:30', 'date': '18-11-2022'}
Published data Successfully: %s {'medicine': 'aspirin', 'Time': '12:30', 'date': '18-11-2022'}
Published data Successfully: %s {'medicine': 'aspirin', 'Time': '12:30', 'date': '18-11-2022'}
Published data Successfully: %s {'medicine': 'aspirin', 'Time': '12:30', 'date': '18-11-2022'}
Published data Successfully: %s {'medicine': 'aspirin', 'Time': '12:30', 'date': '18-11-2022'}
Published data Successfully: %s {'medicine': 'aspirin', 'Time': '12:30', 'date': '18-11-2022'}
Published data Successfully: %s {'medicine': 'aspirin', 'Time': '12:30', 'date': '18-11-2022'}
Published data Successfully: %s {'medicine': 'aspirin', 'Time': '12:30', 'date': '18-11-2022'}
Published data Successfully: %s {'medicine': 'aspirin', 'Time': '12:30', 'date': '18-11-2022'}
Published data Successfully: %s {'medicine': 'aspirin', 'Time': '12:30', 'date': '18-11-2022'}
Published data Successfully: %s {'medicine': 'aspirin', 'Time': '12:30', 'date': '18-11-2022'}
Published data Successfully: %s {'medicine': 'aspirin', 'Time': '12:30', 'date': '18-11-2022'}
Published data Successfully: %s {'medicine': 'aspirin', 'Time': '12:30', 'date': '18-11-2022'}
Published data Successfully: %s {'medicine': 'aspirin', 'Time': '12:30', 'date': '18-11-2022'}
```

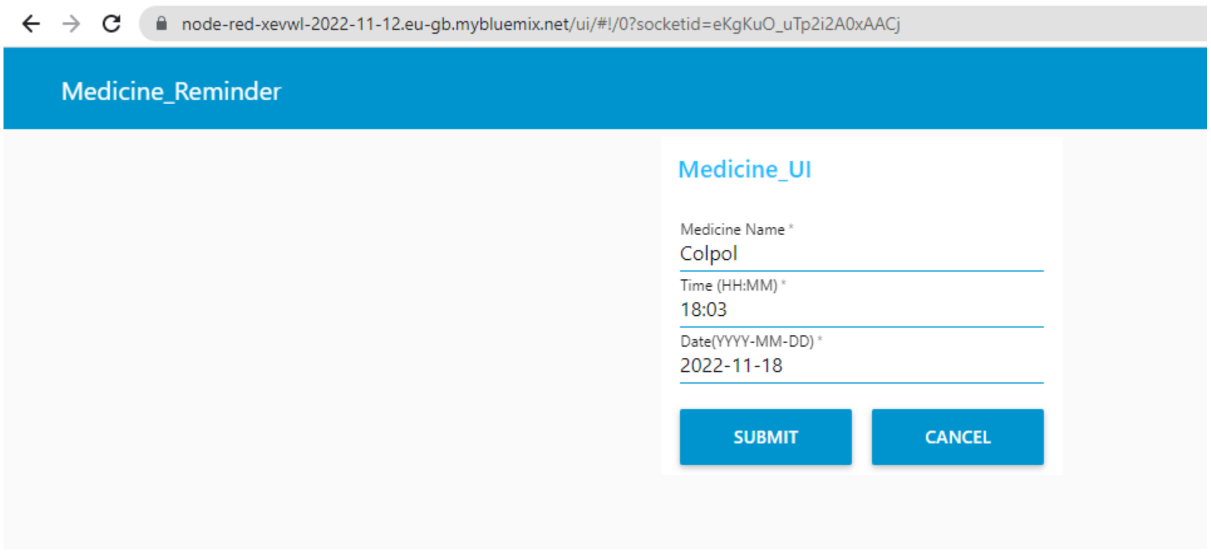
➢ Medicine data uploaded to IBM Watson platform using python script



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

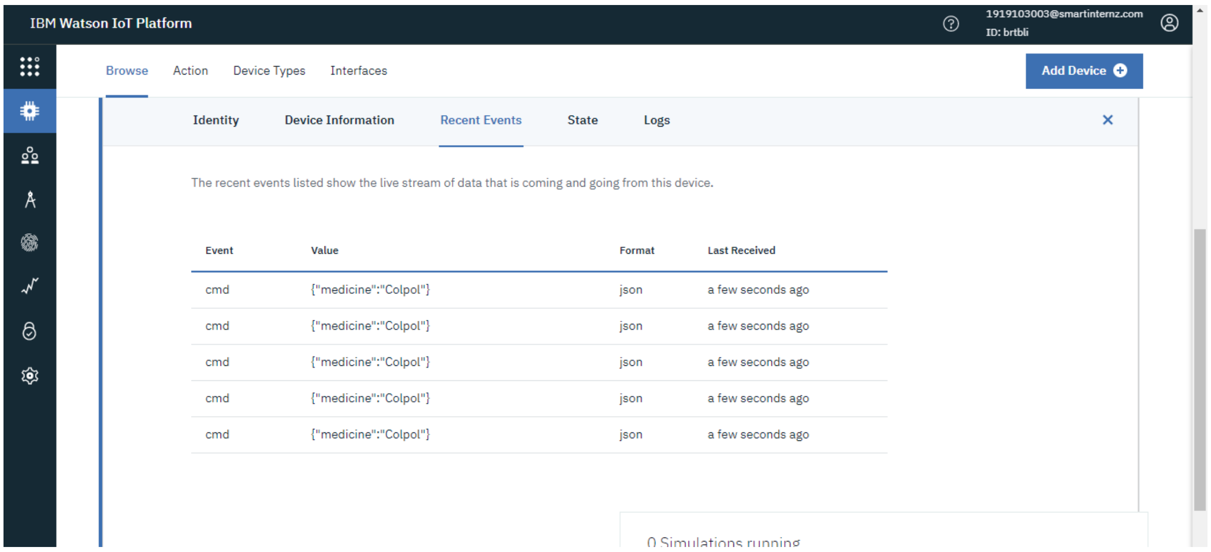
Event	Value	Format	Last Received
status	{"medicine":"aspirin","Time":"12:30","date":"18-...	json	a few seconds ago
status	{"medicine":"aspirin","Time":"12:30","date":"18-...	json	a few seconds ago
status	{"medicine":"aspirin","Time":"12:30","date":"18-...	json	a few seconds ago
status	{"medicine":"aspirin","Time":"12:30","date":"18-...	json	a few seconds ago
status	{"medicine":"aspirin","Time":"12:30","date":"18-...	json	a few seconds ago

➢ Medicine name sent to IBM Watson platform through web application created using Node-RED platform


node-red-xevwl-2022-11-12.eu-gb.mybluemix.net/ui/#!/0?socketid=eKgKuO_uTp2i2A0xAACj
Medicine_Reminder
Medicine_UI
SUBMIT
CANCEL

Medicine Name *
Colpol
Time (HH:MM) *
18:03
Date(YYYY-MM-DD) *
2022-11-18

➢ IBM Watson platform



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

Event	Value	Format	Last Received
cmd	{"medicine":"Colpol"}	json	a few seconds ago
cmd	{"medicine":"Colpol"}	json	a few seconds ago
cmd	{"medicine":"Colpol"}	json	a few seconds ago
cmd	{"medicine":"Colpol"}	json	a few seconds ago
cmd	{"medicine":"Colpol"}	json	a few seconds ago

0 Simulations running

Code for Simulation:

```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt
#include <LiquidCrystal_I2C.h>
#define LED 2
void callback(char* subscribetopic, byte* payload, unsigned int
payloadLength);


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

#define ORG "brtbli"//IBM ORGANITION ID
#define DEVICE_TYPE "Mydevice01"//Device type mentioned in ibm watson IOT
Platform
#define DEVICE_ID "12345"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "9uhV*uqZLpBlHUXugg"     //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/medicine/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);
  pinMode(LED,OUTPUT);
  delay(10);
  Serial.println();
  wificonnect();
  mqttconnect();
}

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

/*....................................retrieving to
Cloud...............................*/


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("Please take "+ data3);
  if(data3 != "")
  {
    lcd.init();
    lcd.print("Take"+ data3);

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

  }

  else
  {
digitalWrite(LED,LOW);

  }
data3="";
 }
```

Python Script to receive data from node-red by using IBM Watson IoT platform:

```
#IBM Watson IOT Platform
#pip install wiotp-sdk
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity": {
        "orgId": "brtbli",
        "typeId": "Mydevice01",
        "deviceId":"12345"
    },
    "auth": {
        "token": "9uhV*uqZLpBlHUXugg"
    }
}

def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
    m=cmd.data['command']

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

while True:
    med1="aspirin"
    time1="12:30"
    Date="18-11-2022"
    myData={'medicine':med1, 'Time':time1, 'date':Date}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
onPublish=None)
```

```
    print("Published data Successfully: %s", myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)
client.disconnect()
```