

**PROJECT DEVELOPMENT PHASE**  
**DELIVERY OF SPRINT 3**

Date	14 November 2022
Team ID	PNT2022TMID33765
Project Name	Personal Assistance for Seniors who are Self-reliant

**SPRINT III: Development of Python code (Software implementation)**

**Outline of Sprint 3**

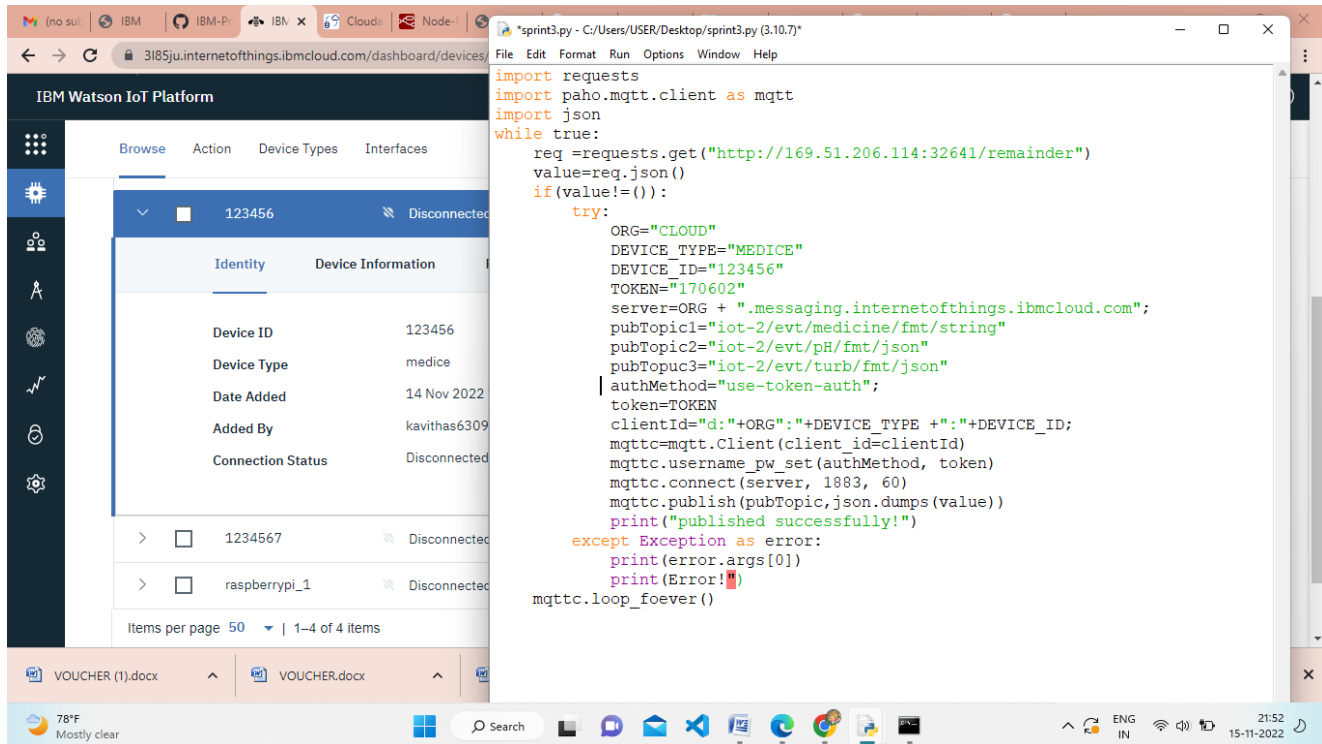
This sprint delivery document contains the following,

- 1)Python code to receive data from node red and send to IoT Watson platform
- 2)Updation of nodes in the node-red platform
- 3)The results of the web UI after deploying.

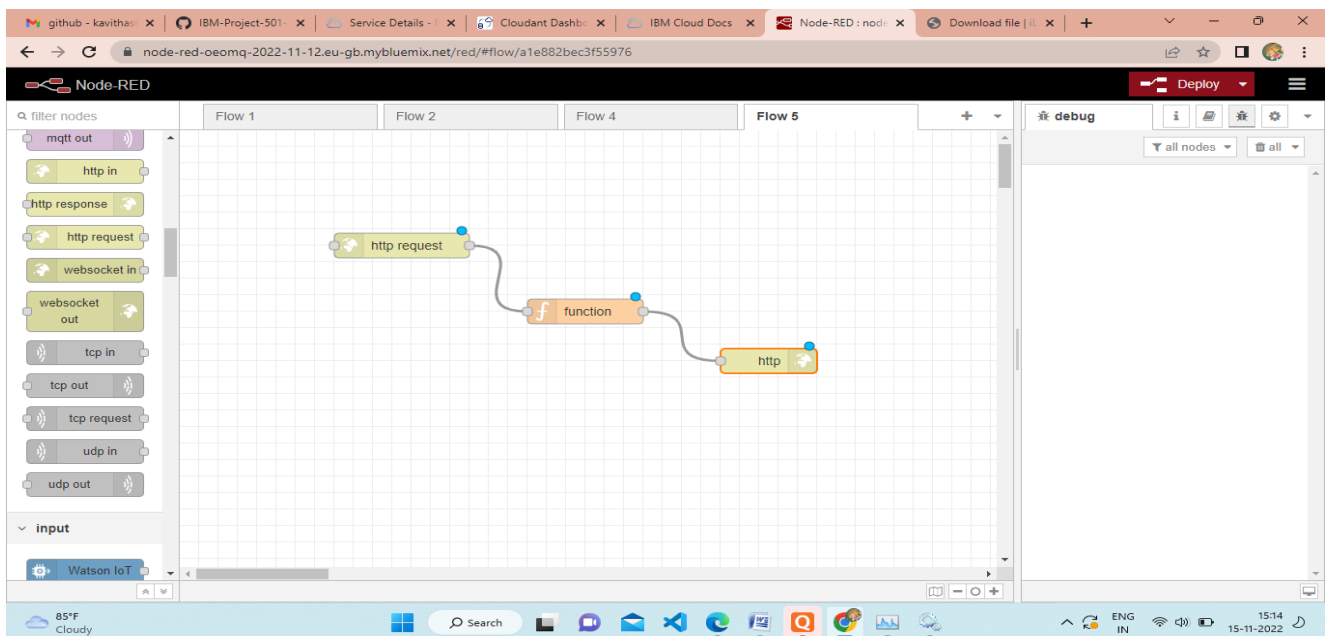
**Python code to receive data from node red and send to IoT Watson platform**

```
import requests
import paho.mqtt.client as mqtt
import json
while true:
    req =requests.get('http://169.51.206.114:32641/remainder')
    value=req.json()
    if(value!=()):
        try:
            ORG="CLOUD"
            DEVICE_TYPE="MEDICE"
            DEVICE_ID="123456"
            TOKEN="170602"
            server=ORG + ".messaging.internetofthings.ibmcloud.com";
            pubTopic1="iot-2/evt/medicine/fmt/string"
            pubTopic2="iot-2/evt/pH/fmt/json"
            pubTopic3="iot-2/evt/turb/fmt/json"
            authMethod="use-token-auth";
            token=TOKEN
            clientId="d:"+ORG+": "+DEVICE_TYPE +": "+DEVICE_ID;
            mqttc=mqtt.Client(client_id=clientId)
            mqttc.username_pw_set(authMethod, token)
            mqttc.connect(server, 1883, 60)
            mqttc.publish(pubTopic,json.dumps(value))
            print("published successfully!")
        except Exception as error:
            print(error.args[0])
```

```
print(Error!"))
mqttc.loop_forever()
```

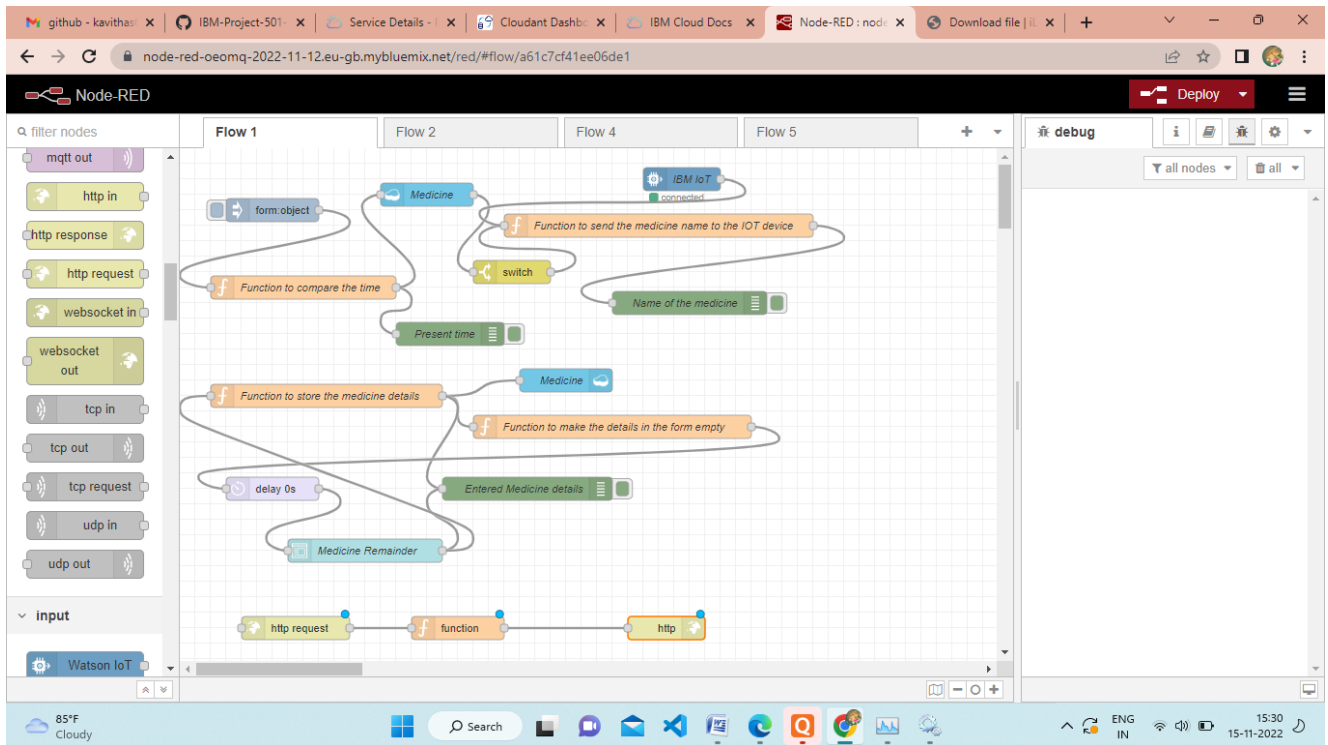


## 1) Updation of nodes in the node-red platform



The following function is returned in order to store get the medicine name in http to facilitate request and response via python code

### The Updated node red flow diagram



Creating the IBM Watson account and add the devices, simulating the devices:

The screenshot displays the IBM Watson IoT Platform dashboard with the following elements:

- Navigation Bar:** Includes tabs for Browse, Action, Device Types, and Interfaces.
- Browse Devices Section:** Contains a table of devices and a "Send" button for events.
- Event Configuration Panel:** Shows details for an event named "event\_1" with a frequency of "20 x Every Minute".
- Event Payload:** A JSON object defining the event data: 

```
{ 0: { 1: "randomNumber": random(0, 100), 2: "medicine": "Amitriptyline", 3: "time": "20:00", 4: } }
```
- Device Table:** Lists devices with columns for Device ID, Status, and Device Type.

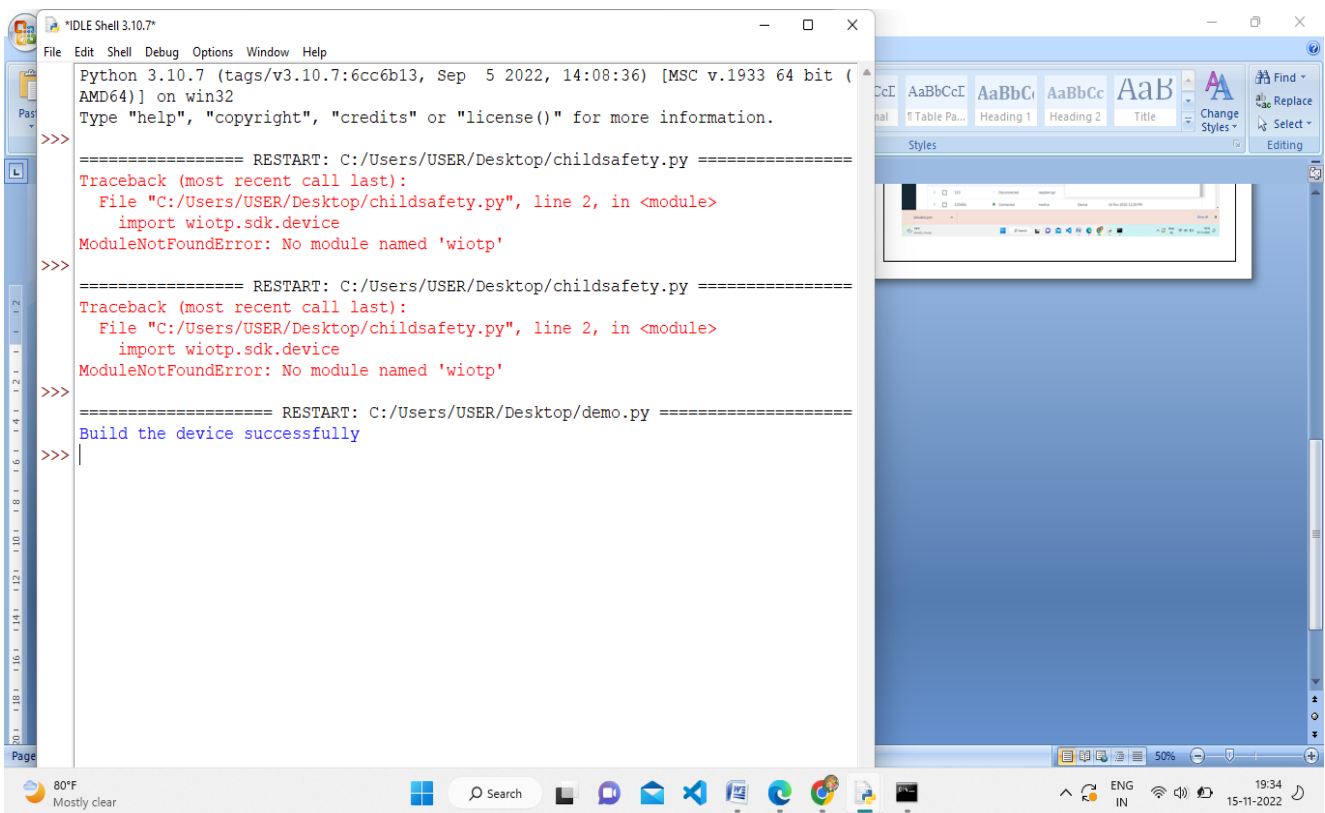
Device ID	Status	Device Type
123	Disconnected	raspberrypi
123456	Connected	medice

## 2) The results of the web UI after deploying.

The above result reveals that the medicine name is sent to the IoT Watson platform using the above developed python code.

The medicine name “Ofloxacin” is scheduled to be took at 23:00. The name of the medicine is displayed in the IoT platform at 23:00

The next step would be implementing TTS service to spell out the medicine name at the appropriate time.



The screenshot displays a Windows desktop environment. On the left, an 'IDLE Shell 3.10.7\*' window is open, showing a Python 3.10.7 prompt. The user has entered commands to restart and run a script named 'chilfsafety.py'. The output shows a 'ModuleNotFoundError: No module named 'wiotp'' error. The user then enters a command to run 'demo.py', which outputs 'Build the device successfully'. On the right, a web browser window is partially visible, showing a blue background. The Windows taskbar at the bottom includes the Start button, a search bar, and various application icons. The system tray shows the date and time as 19:34 on 15-11-2022.

```
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep 5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/USER/Desktop/chilfsafety.py =====
Traceback (most recent call last):
  File "C:/Users/USER/Desktop/chilfsafety.py", line 2, in <module>
    import wiotp.sdk.device
ModuleNotFoundError: No module named 'wiotp'
>>>
===== RESTART: C:/Users/USER/Desktop/chilfsafety.py =====
Traceback (most recent call last):
  File "C:/Users/USER/Desktop/chilfsafety.py", line 2, in <module>
    import wiotp.sdk.device
ModuleNotFoundError: No module named 'wiotp'
>>>
===== RESTART: C:/Users/USER/Desktop/demo.py =====
Build the device successfully
>>> |
```