PROJECT DEVELOPMENT PHASE

DELIVERY OF SPRINT 4

Date	14th November 2022
Team ID	PNT2022TMID35860
Project Name	Project – Personal Assistance for senior citizens who are self-reliant
Team members	K.Gurubaran B.Mejalin Arno J.Vinothagan R.Arunkumar

SPRINT IV: Generating voice commands and Alert system

Outline of Sprint 4

This sprint delivery document contains the following,

1)Developing Python code to implement Text to speech service

2)Alert system via email by Ubidots notification

3)Updation of nodes in the node-red platform

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration. Creation of IBM services like NodeRED, Cloudant DB, TTS Service and design of IoT system	USN-1	As a user, I must be able to login to the IBM platform	2	High	Gurubaran, Arunkumar
Sprint-2	Web UI. Creation of Web UI using NodeRED service	USN-2	As a user, I must be able to update the medicine details in the web UI	2	High	Vinothagan, Mejalin Arno
Sprint-3	Software implementation. Developing Python code to retrieve data from cloudant db to send that data to IoT device	USN-3	As a user, I must be push the details to the IoT device	2	High	Gurubaran, Mejalin Arno
Sprint-4	Final demonstration and user testing. Generating voice commands using IBM Text to Speech service	USN-4	As a user, I must be able hear the medicine name which is to be taken at the appropriate time and check its accuracy	2	High	Vinothagan, Arunkumar

1)Developing Python code to implement Text to speech service

a) Text to Speech piece of code

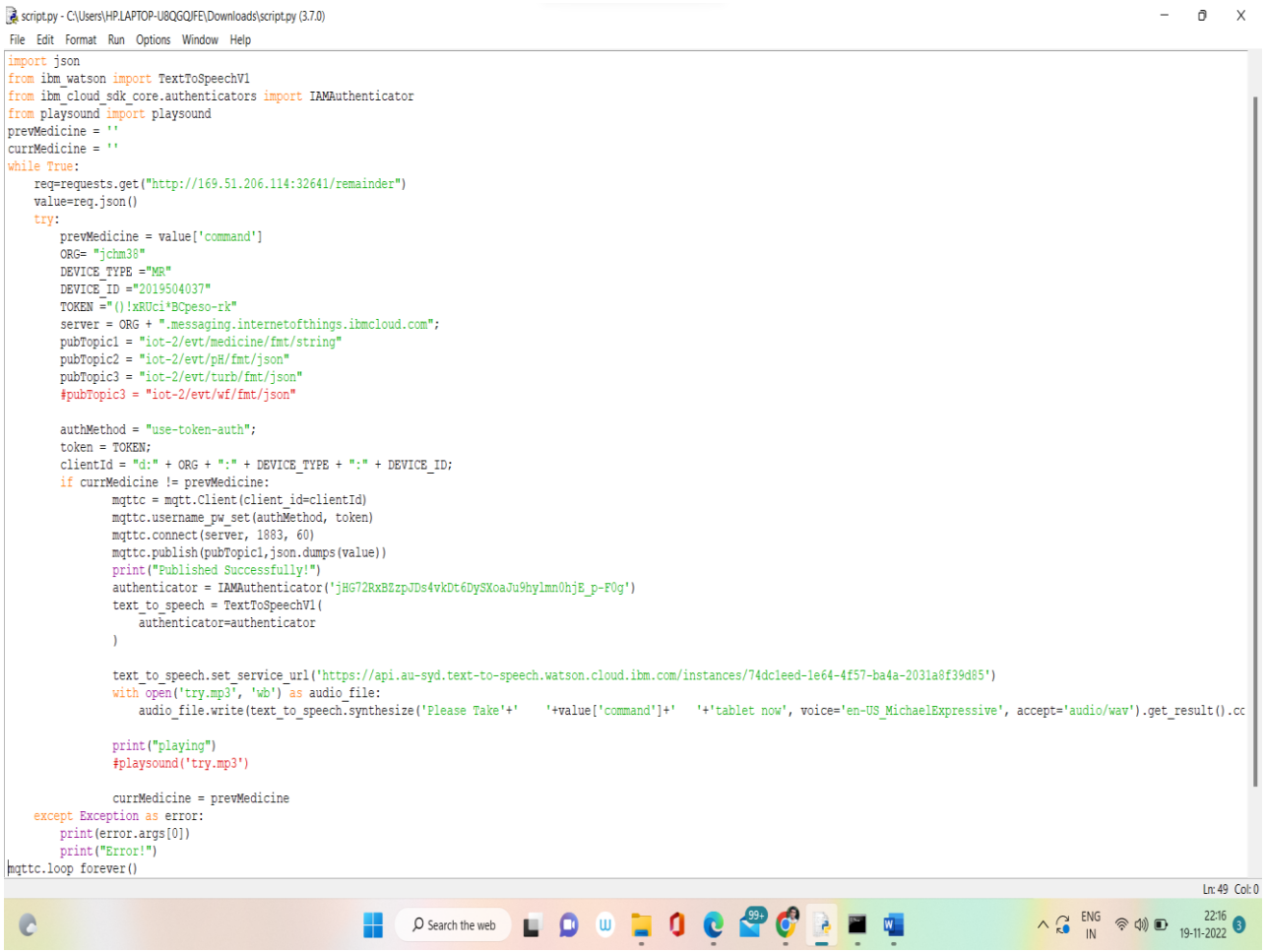
text2speech.py - C:\Users\HP.LAPTOP-U8QGQJFE\Downloads\text2speech.py (3.7.0)

File Edit Format Run Options Window Help

```
from ibm_watson import TextToSpeechV1
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator
from playsound import playsound

authenticator = IAMAuthenticator('jHG72RxBZzpJDs4vkDt6DySXoaJu9hylmn0hjE_p-F0g')
text_to_speech = TextToSpeechV1(
    authenticator=authenticator
)

text_to_speech.set_service_url('https://api.au-syd.text-to-speech.watson.cloud.ibm.com/instances/74dc1eed-1e64-4f57-ba4a-2031a8f39d85')
with open('med.mp3', 'wb') as audio_file:
    audio_file.write(text_to_speech.synthesize('Take Crocin 50 mg Now', voice='en-US_AllisonV3Voice', accept='audio/wav').get_result().content)

print("playing")
playsound('med.mp3')
```

Using ibm_watson library, we are going to implement text to speech service

The final code:



script.py - C:\Users\HP.LAPTOP-U8QGQJFE\Downloads\script.py (3.7.0)

File Edit Format Run Options Window Help

```
import json
from ibm_watson import TextToSpeechV1
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator
from playsound import playsound
prevMedicine = ''
currMedicine = ''
while True:
    req=requests.get("http://169.51.206.114:32641/remainder")
    value=req.json()
    try:
        prevMedicine = value['command']
        ORG= "jchm38"
        DEVICE_TYPE ="MR"
        DEVICE_ID ="2019504037"
        TOKEN ="()!xRUci*BCpeso-rk"
        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"
        #pubTopic3 = "iot-2/evt/wf/fmt/json"

        authMethod = "use-token-auth";
        token = TOKEN;
        clientId = "d:" + ORG + ":" + DEVICE_TYPE + ":" + DEVICE_ID;
        if currMedicine != prevMedicine:
                mqttc = mqtt.Client(client_id=clientId)
                mqttc.username_pw_set(authMethod, token)
                mqttc.connect(server, 1883, 60)
                mqttc.publish(pubTopic1,json.dumps(value))
                print("Published Successfully!")
                authenticator = IAMAuthenticator('jHG72RxBZzpJDs4vkDt6DySXoaJu9hylmn0hjE_p-F0g')
                text_to_speech = TextToSpeechV1(
                    authenticator=authenticator
                )

                text_to_speech.set_service_url('https://api.au-syd.text-to-speech.watson.cloud.ibm.com/instances/74dc1eed-1e64-4f57-ba4a-2031a8f39d85')
                with open('try.mp3', 'wb') as audio_file:
                    audio_file.write(text_to_speech.synthesize('Please Take'+'    '+value['command']+'   '+'tablet now', voice='en-US_MichaelExpressive', accept='audio/wav').get_result().cc

                print("playing")
                #playsound('try.mp3')

                currMedicine = prevMedicine
    except Exception as error:
        print(error.args[0])
        print("Error!")
mqttc.loop_forever()
```

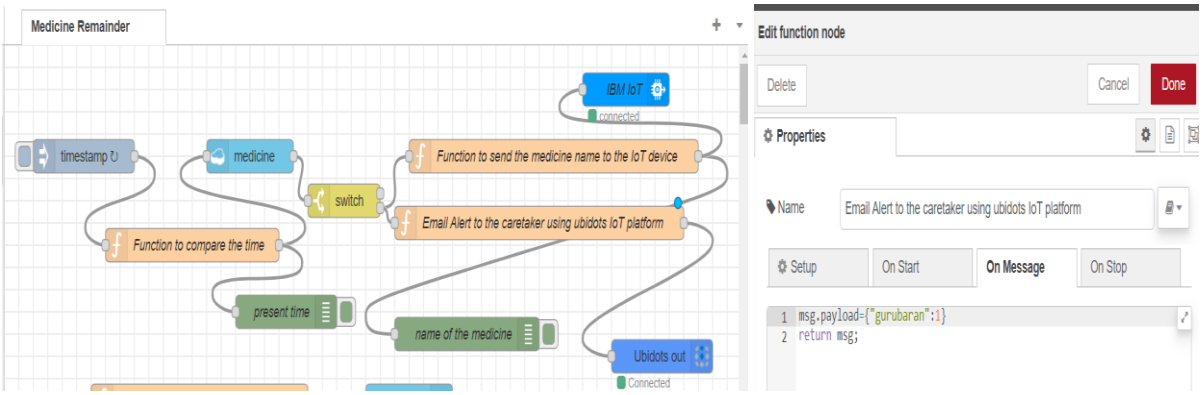
Ln: 49 Col: 0

Output

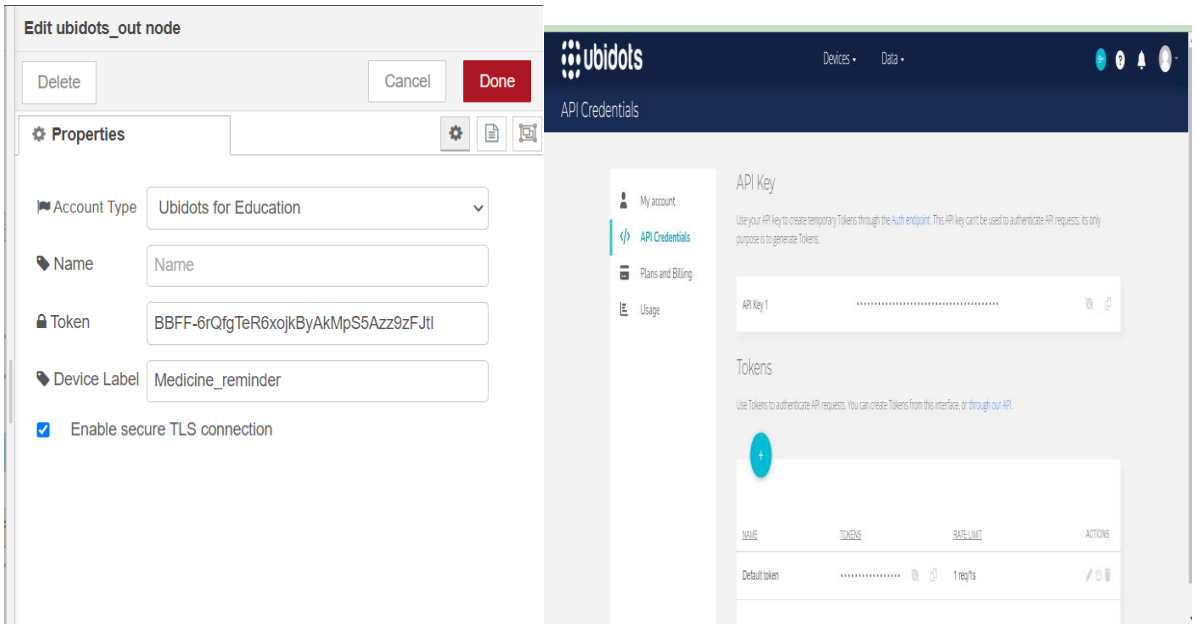
```
*Python 3.7.0 Shell*
File  Edit  Shell  Debug  Options  Window  Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD6
4)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:\Users\HP.LAPTOP-U8QGQJFE\Downloads\script.py ==========
Published Successfully!
playing
Published Successfully!
playing
```

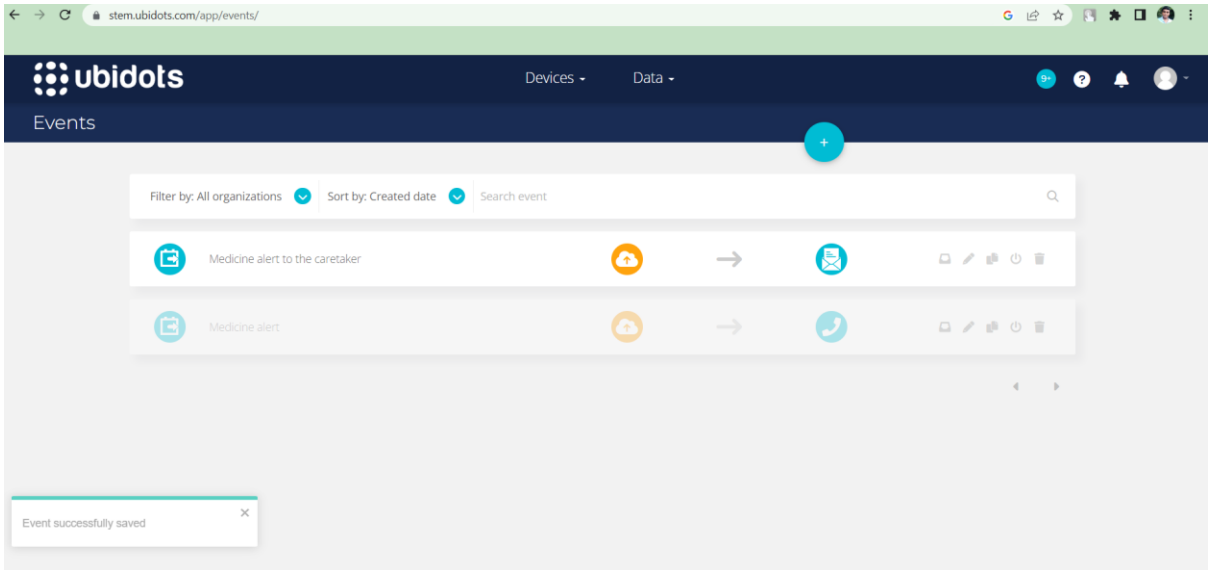
2) Alert system via email by Ubidots notification

By updating the node red flow by adding ubidots out node and creating a function

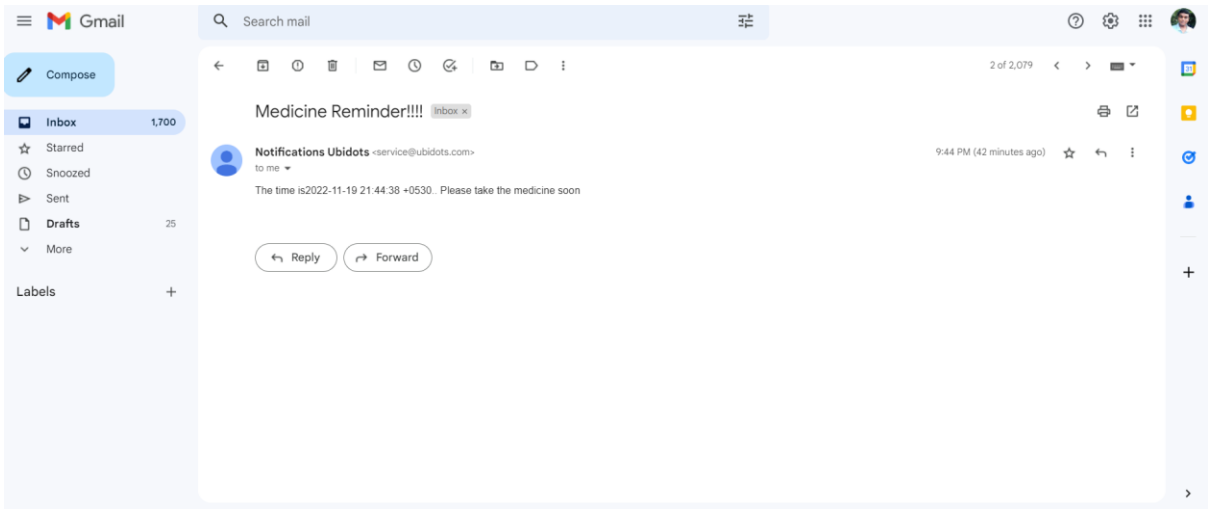


Ubidots node edit and web platform screenshots

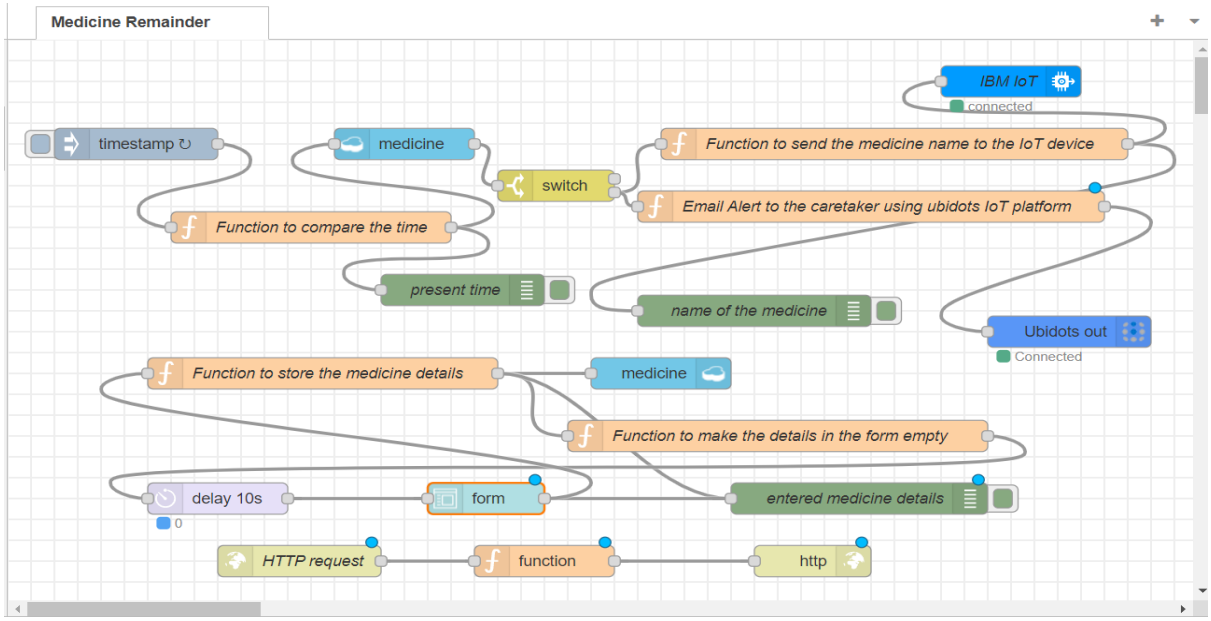




Email sent to the caretaker

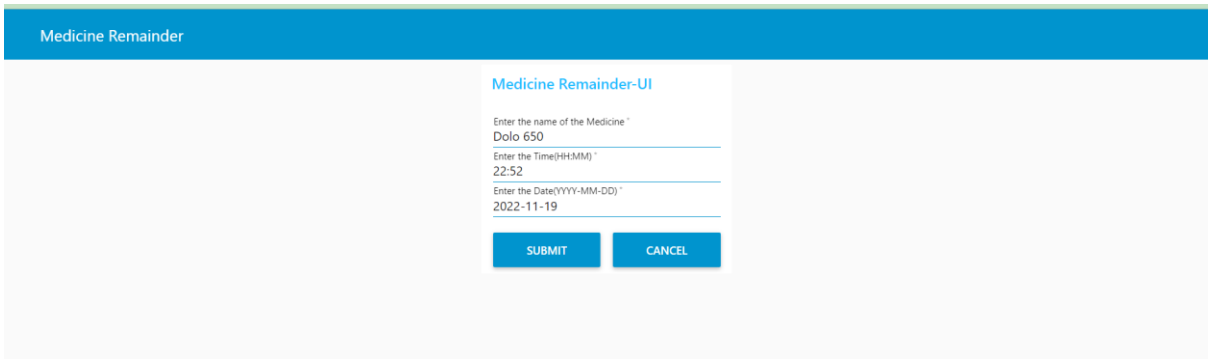


3)Updated nodered flow diagram

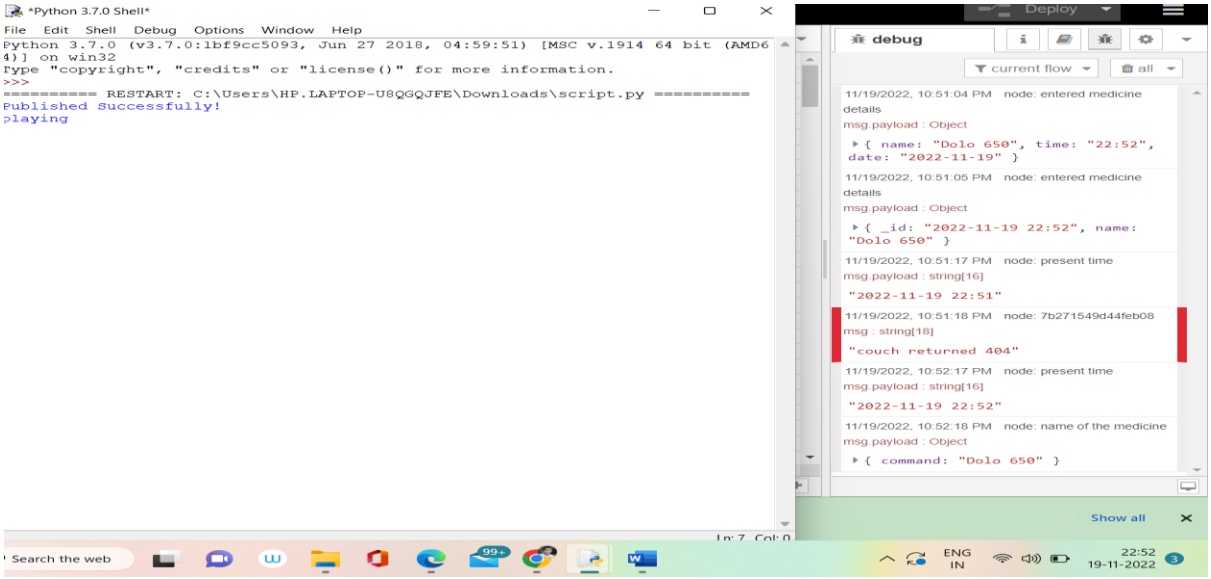


Result:

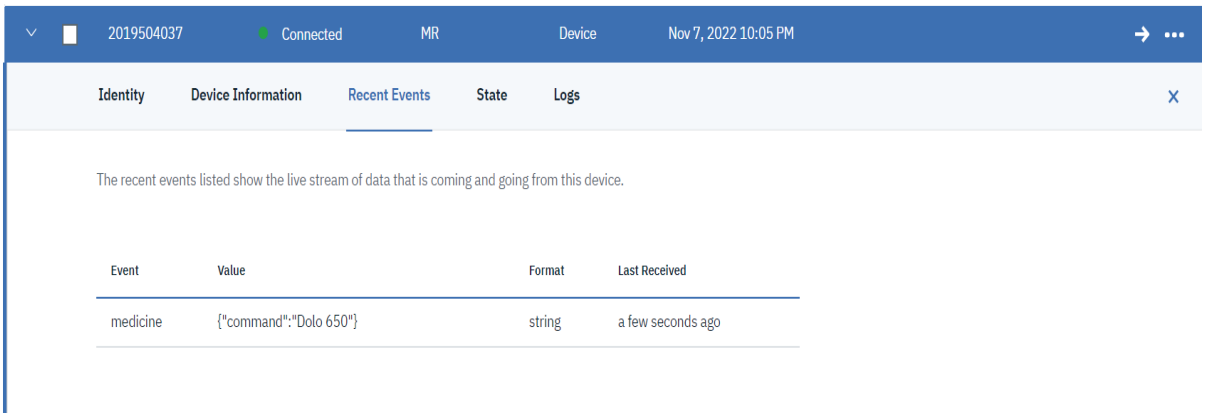
1)Medicine details are uploaded by the care-taker using the link http://169.51.206.114:32641/ui



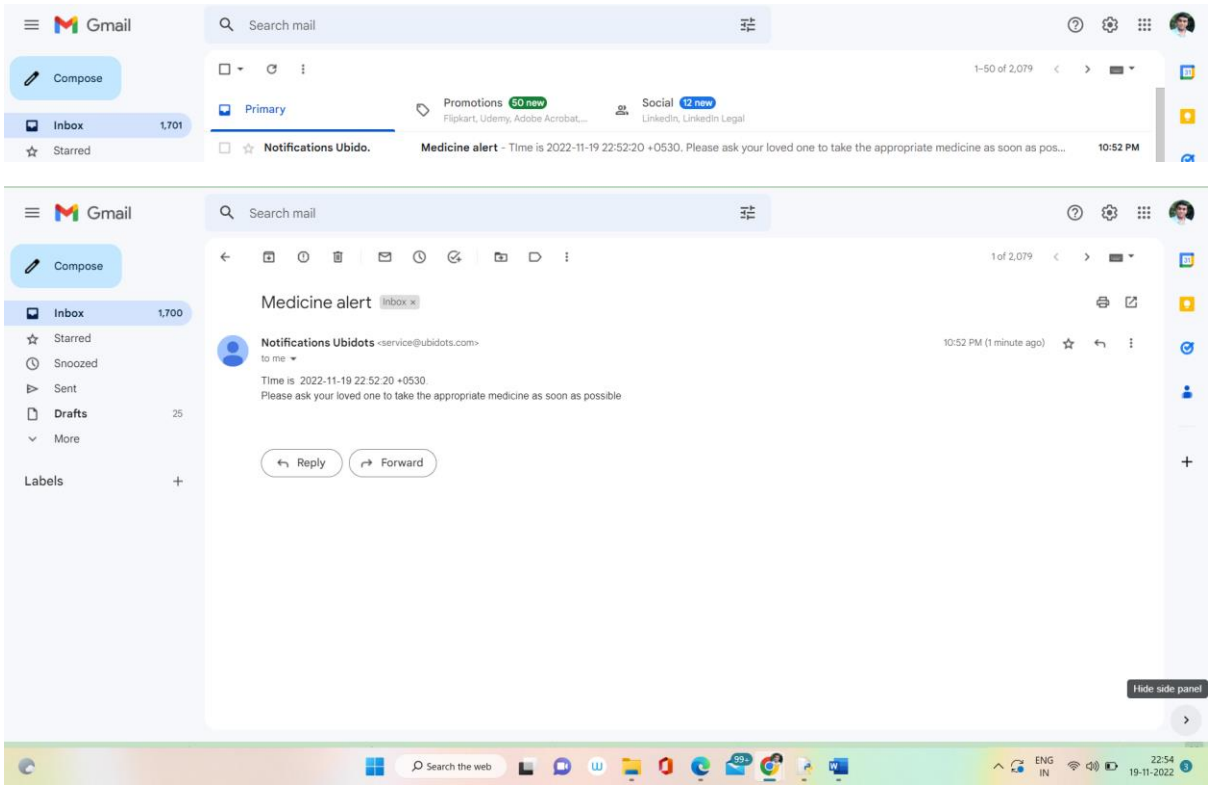
2)When the time arrives, the python code will send the name of the medicine to the IBM IoT Watson platform



3)The following is the screen shot of the recent events in the IoT Watson platform



4)An email is sent to the care taker asking the person to intimate the elderly patient



5) Text to Speech command signals to the user : The file is uploaded in the google drive.

https://drive.google.com/file/d/190BT9642_KT801JEqu5nEtIwwkkUql1Y/view?usp=sharing

Summary of Sprint 4:

1)Some features are added to the python code to enable TTS mechanism

2)An email is sent to the care taker to intimate the time of medicine intake of the patient

3)Time of medicine is intimated to the user via voice commands

****************************END OF REPORT************************************