SmartBridge IOT Project- Soldier Health Monitoring System Using IBM Cloud

Name: **Harshitha Munagala**

Registration number: **19BEC0565**

Table of Contents

1.INTRODUCTION

(a) Overview

Our project is Soldier Health Monitoring System Using IBM Cloud. We have obtained random location of the soldiers and their health conditions which includes temperature, pulse rate and blood pressure using python code and sent it to the cloud. We have used Node-RED to configure the flow to receive the data. Cloudant DB nodes are used to store the received data in the Cloudant DB. The data is published in the web application.

(b) Purpose

The purpose of this project is to track the location of the soldiers in a certain range and to check if their health condition is normal or abnormal. If the soldier is out of range, then a notification is sent to the authorities. If the received data of the health conditions of the soldiers is not normal, then a corresponding notification is sent to the authorities as well. This is a highly useful application as we can track and analyse the condition and the location of the soldier without direct communication.

2.LITERATURE SURVEY

(a) Existing Problem

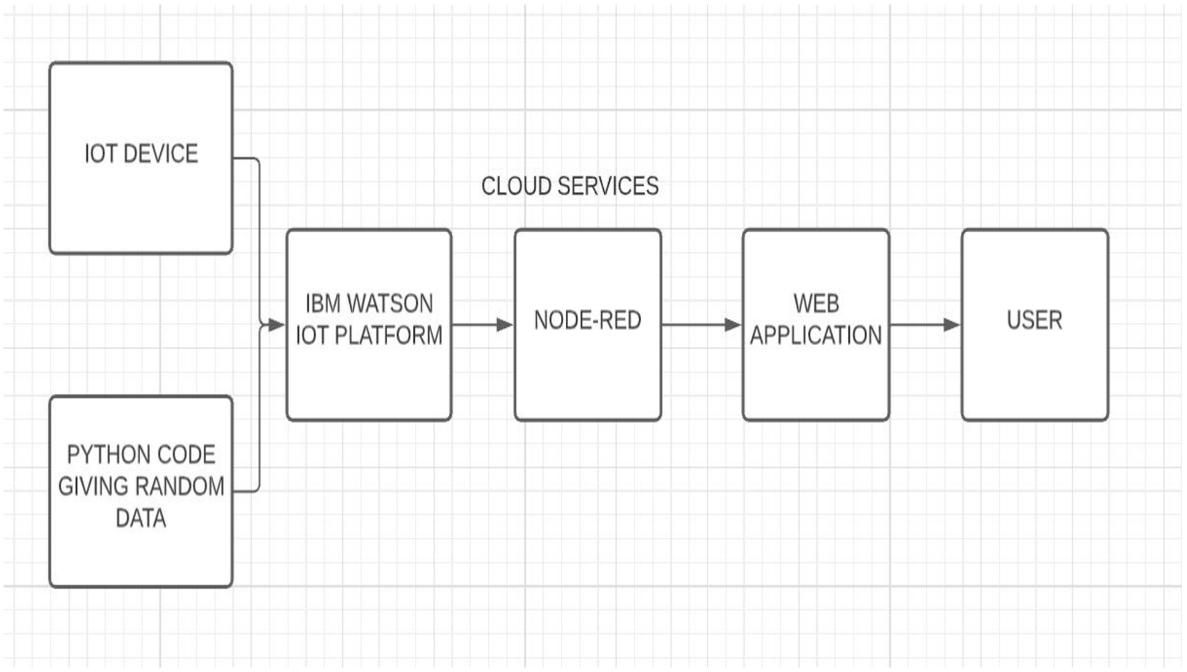
Tracking the soldiers is a pivotal function to carry out in the military. This will increase the efficiency and provide aid to the soldiers whose health conditions are sensed to be abnormal. At present only through communication can the authorities obtain information about the condition of their soldiers.

(b) Proposed Solution

We propose to solve this issue by obtaining the sensor data of the soldiers and analyse their health condition and location. The data will be displayed in the form of visual representation in the web application and if the parameters such as temperature, pulse and blood pressure are not within the normal range of a healthy person then the authorities are promptly notified with the corresponding message. The sensor data is also stored in the Cloudant DB.

3. THEORETICAL ANALYSIS

(a) Block Diagram



(b) Software Designing

We develop a python code to obtain random sensor data for location (latitude and longitude), body temperature, pulse rate and blood pressure.

We integrate the code with IBM Cloudant and store the received data in IBM Database.

Using dashboard nodes, we create a User Interface in Node-RED which displays the sensor data.

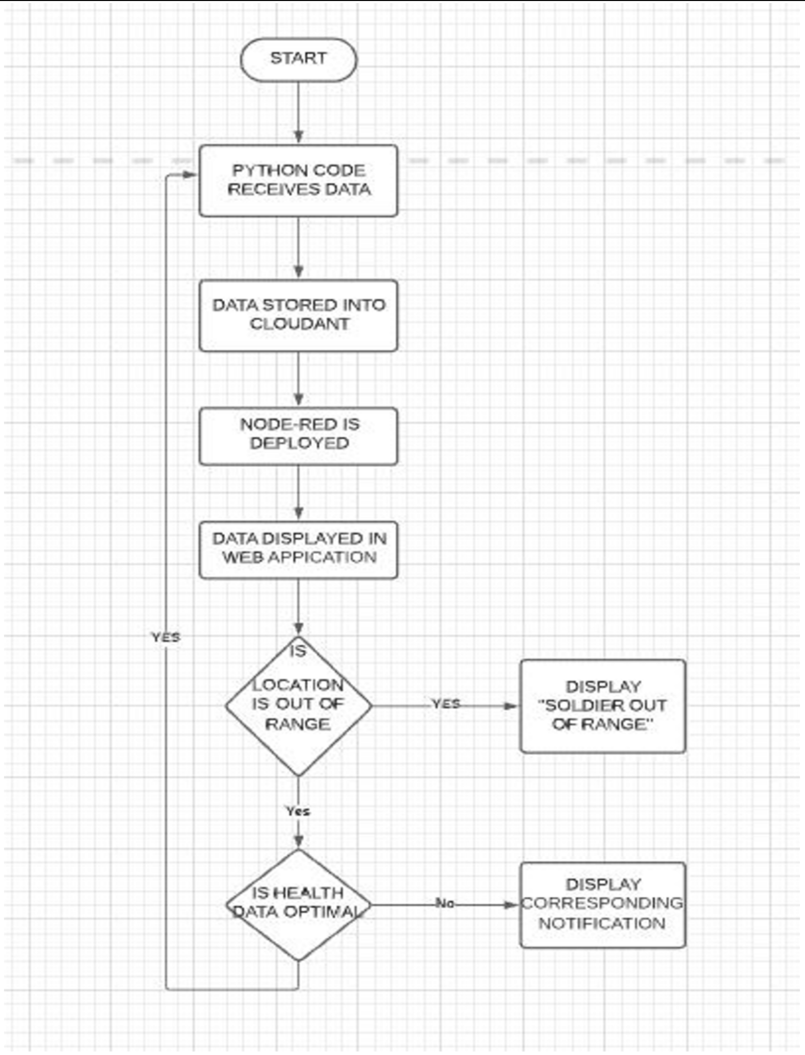
When the health conditions of the soldier is out of normal range, suitable alert notification is sent to the authorities.

4. EXPERIMENTAL INVESTIGATION

Python code was developed using in-built packages and functions and soldier data was generated.

Node-RED dashboards required was studied and installed for plotting the location of the soldier. We have collected the normal range of body temperature, pulse rate, and blood pressure to cross check with the received data from soldiers. The user interface is designed for understanding the data with ease using suitable charts for display.

5. FLOWCHART


START
PYTHON CODE RECEIVES DATA
DATA STORED INTO CLOUDANT
NODE-RED IS DEPLOYED
DATA DISPLAYED IN WEB APPICATION
IS LOCATION IS OUT OF RANGE
YES
DISPLAY "SOLDIER OUT OF RANGE"
Yes
IS HEALTH DATA OPTIMAL
No
DISPLAY CORRESPONDING NOTIFICATION
YES

6. RESULT

By following the above steps, we have displayed the data in the web UI and stored the data in IBM Cloud. The location of the soldiers is tracked and health condition is monitored. Notifications are sent to Authorities if soldier's health is abnormal and the data is stored in Cloudant DB.

7. ADVANTAGES AND DISADVANTAGES

Advantages:

The IoT based soldier health monitoring system allows the monitoring of a soldier’s body temperature, blood pressure levels and pulse rate. It also helps us track the location of the soldier, which is immensely useful during times of emergencies. The exact location and health status can be sent to the base station in real time, so that required actions can be taken in the base station. It is a highly reliable and cost-effective system. Due to the inclusion of IoT, it performs at high speed and transmits data faster.

Disadvantages:

One of the major drawbacks of this system is the short range it offers. It might be difficult to transmit accurate data when the soldier is far away from the base station. Data transmission can also be affected by rough weather and connectivity issues. In addition to this. wifi-based systems are always prone to hackers who might meddle with the data being sent to the bases.

8. APPLICATIONS

This system can be of great use in the military to track and monitor the location and health condition of soldiers. Since the system allows us to measure various health parameters, it can give the officials in the base the exact health status of the soldier without having to go to the field. This can be useful in detecting emergencies during times of extreme weather conditions to immediately rescue the soldier.

Apart from this, the accurate location data of the soldier can also be transmitted to the army base. This enables them to know the current position of the soldier and can be useful in alerting them if any incoming danger is detected. This is also useful in times of war to find out if the soldier is in close proximity with enemies and can be used to warn them to move to safer locations.

9. CONCLUSION

From the proposed system above, we conclude that transmission happens at fixed intervals of time and the readings are accurate. From the soldier unit the data is transmitted to the base camp unit and if any abnormalities are notified in the health conditions, the officials in the base station are immediately notified about it. This would really help in early tracking of danger and immediate actions can be taken to save lives of soldiers. This system helps in monitoring the health parameters of the soldier, atmospheric conditions and their locations.

The system also assists the soldier in getting help from the army base camp and or from another fellow soldier in situations of emergency. This would be very useful to military forces during war and rescue operations since the system can be used without any network restriction. Hence this system provides safety and protection to soldiers. The system has a very effective modelling design. The system is a reasonable one as it can be easily attached to the hand of the soldier.

10. FUTURE SCOPE

For any system, there is always scope for improvement with development in technology in upcoming times. With regard to the soldier health and location monitoring system, a compatible and better routing algorithm can be utilized to make this system much more reliable and efficient in terms of energy. New technologies can be incorporated in order to expand the capacity of data transmission. Methods to increase the number of health parameters being measured can be taken.

Additional features such as the attachment of a microphone can be done to the system in order to record external voices and also directly communicate with the soldier. A camera can also be added to take pictures of the location of the soldier so that tracking can be made easier in times of emergency. The range of communication of the system can also be expanded in order to expand the area of coverage and to ensure that data transmission is not stopped due to the increase in distance between the base station and the present location of the soldier.

11. BIBLIOGRAPHY

- Soldier Health and Position Tracking System using GPS and GSM Modem, by Deepa J, Ranjini, Sharanya Raj, Dr. Parameshachari B D Students, BE, Department of TCE, GSSSIETW, Mysuru, Karnataka, India 2 Professor and Head, Department of TCE, GSSSIETW, Mysuru, Karnataka, India.

- IoT Based Soldier Monitoring System Arya V Nair, Rani Raju, Tinsa Elsa Thomas, Vidya R Nair, Nidiya Habeeb #5
  Student , Dept of Electronics & Communication Engineering, Musaliar College of Engineering & Technology, Pathanamthitta
  5, Associate Professor, Dept of Electronics & Communication Engineering, Musaliar College of Engineering & Technology, Pathanamthitta

- SOLDIER TRACKING AND HEALTH MONITORING SYSTEMS 1.SHWETA SHELAR, 2.NIKHIL PATIL, 3.MANISH JAIN, 4.SAYALI CHAUDHARI, 5.SMITA HANDE
  1,2,3,4 B.E Student EXTC FCRIT Vashi
  5Assistant professor EXTC FCRIT Vashi.

- Development of Soldier Tracking with Real Time Health Monitoring and Control System

  1.Bonakruti Nitin Balraj, 2. Karale Shweta Suresh, 3.Gurav Sneha shrirang, 4.Shaikh Tahamina Rajak, 5.Kore Priyanka Suryakant, 6. Rajguru Vaibhav Vijay
  Trainee Engineer 1, 2, 3, 4, 5, 6
  ENTC Department of Engineering College of Brahmadevdada Mane Institute of Technology, Solapur, India 1, 2, 3, 4, 5
  College of A.G.Patil Institute of Technology, Solapur, India

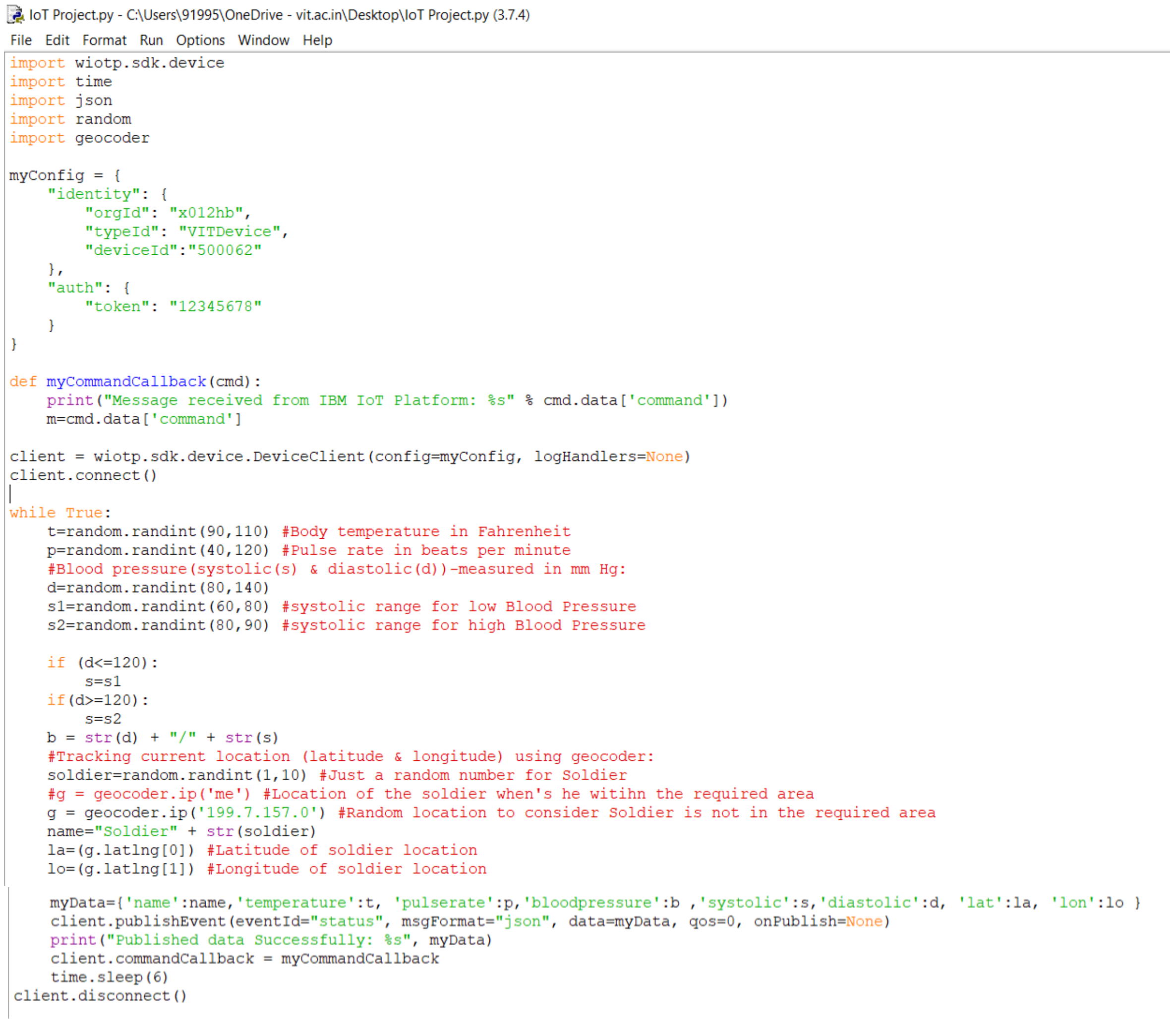
12. APPENDIX

(a) Source code

```
import wiotp.sdk.device
import time
import json
import random
import geocoder

myConfig = {
    "identity": {
        "orgId": "x012hb",
        "typeId": "VITDevice",
        "deviceId":"500062"
    },
    "auth": {
        "token": "12345678"
    }
}

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()
```

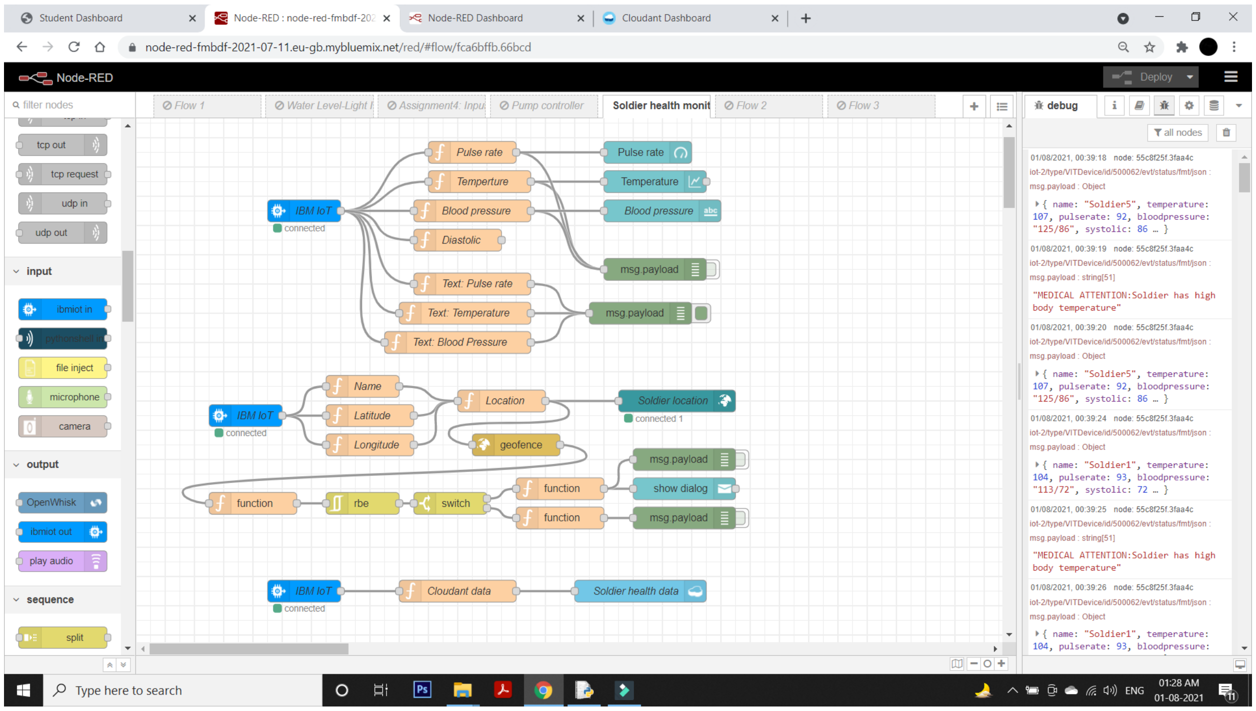
```
while True:
    t=random.randint(90,110) #Body temperature in Fahrenheit
    p=random.randint(40,120) #Pulse rate in beats per minute
    #Blood pressure(systolic(s) & diastolic(d))-measured in mm Hg:
    d=random.randint(80,140)
    s1=random.randint(60,80) #systolic range for low Blood Pressure
    s2=random.randint(80,90) #systolic range for high Blood Pressure


    if (d<=120):
        s=s1
    if(d>=120):
        s=s2
    b = str(d) + "/" + str(s)
    #Tracking current location (latitude & longitude) using geocoder:
    soldier=random.randint(1,10) #Just a random number for Soldier
    #g = geocoder.ip('me') #Location of the soldier when's he witihn the required
area
    g = geocoder.ip('199.7.157.0') #Random location to consider Soldier is not in the
required area
    name="Soldier" + str(soldier)
    la=(g.latlng[0]) #Latitude of soldier location
    lo=(g.latlng[1]) #Longitude of soldier location
    myData={'name':name,'temperature':t, 'pulserate':p,'bloodpressure':b
,'systolic':s,'diastolic':d, 'lat':la, 'lon':lo }
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
onPublish=None)
    print("Published data Successfully: %s", myData)
    client.commandCallback = myCommandCallback
    time.sleep(6)
client.disconnect()
```

(b) UI output Screenshot


IoT Project.py - C:\Users\91995\OneDrive - vit.ac.in\Desktop\IoT Project.py (3.7.4)
File Edit Format Run Options Window Help

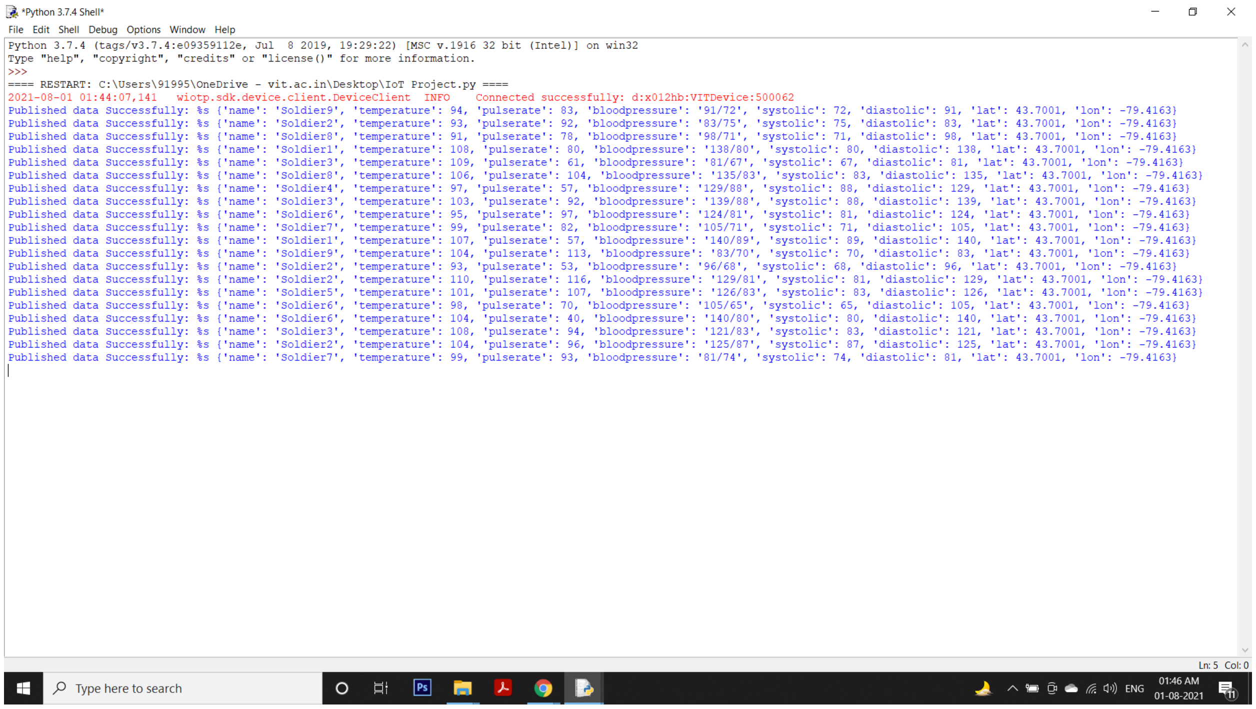
```
import wiotp.sdk.device
import time
import json
import random
import geocoder

myConfig = {
    "identity": {
        "orgId": "x012hb",
        "typeId": "VITDevice",
        "deviceId":"500062"
    },
    "auth": {
        "token": "12345678"
    }
}

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:
    t=random.randint(90,110) #Body temperature in Fahrenheit
    p=random.randint(40,120) #Pulse rate in beats per minute
    #Blood pressure(systolic(s) & diastolic(d))-measured in mm Hg:
    d=random.randint(80,140)
    s1=random.randint(60,80) #systolic range for low Blood Pressure
    s2=random.randint(80,90) #systolic range for high Blood Pressure

    if (d<=120):
        s=s1
    if(d>=120):
        s=s2
    b = str(d) + "/" + str(s)
    #Tracking current location (latitude & longitude) using geocoder:
    soldier=random.randint(1,10) #Just a random number for Soldier
    #g = geocoder.ip('me') #Location of the soldier when's he witihn the required area
    g = geocoder.ip('199.7.157.0') #Random location to consider Soldier is not in the required area
    name="Soldier" + str(soldier)
    la=(g.latlng[0]) #Latitude of soldier location
    lo=(g.latlng[1]) #Longitude of soldier location

    myData={'name':name,'temperature':t, 'pulserate':p,'bloodpressure':b ,'systolic':s,'diastolic':d, 'lat':la, 'lon':lo }
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
    print("Published data Successfully: %s", myData)
    client.commandCallback = myCommandCallback
    time.sleep(6)
client.disconnect()
```

Python code screenshot



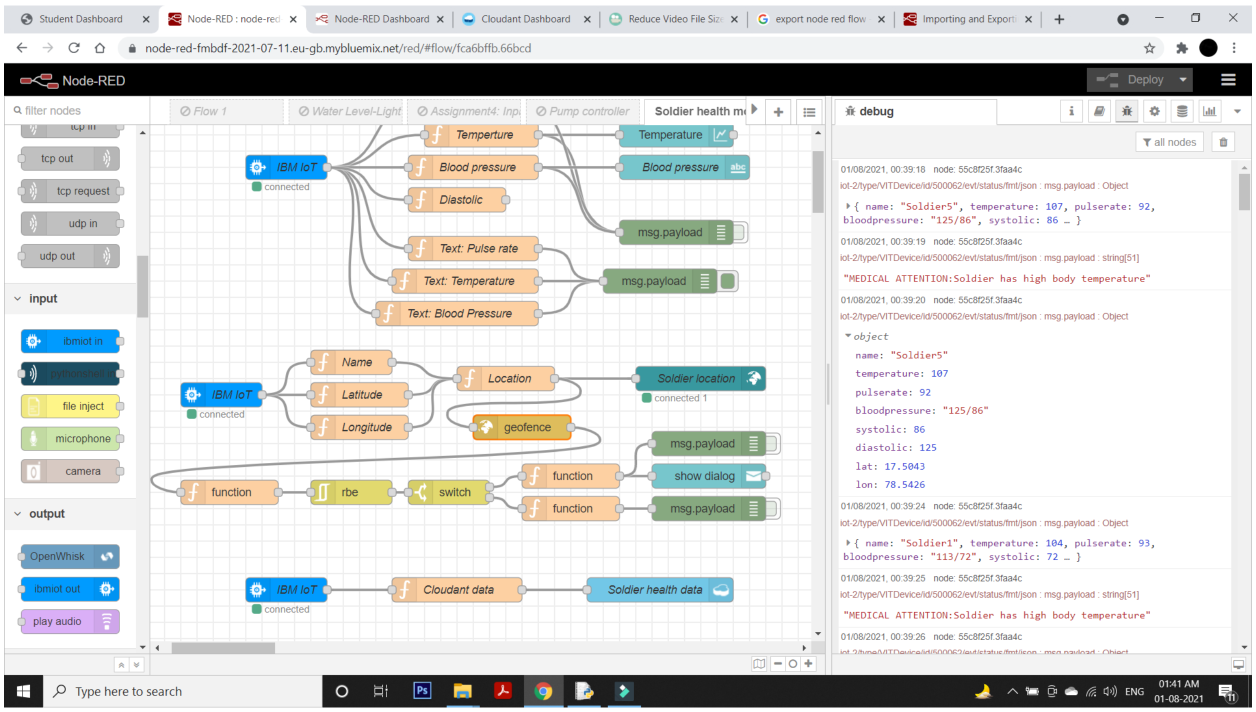
```
01/08/2021, 00:39:18   node: 55c8f25f.3faa4c
iot-2/type/VITDevice/id/500062/evt/status/fmt/json :
msg.payload : Object
▸{ name: "Soldier5", temperature:
107, pulserate: 92, bloodpressure:
"125/86", systolic: 86 … }
01/08/2021, 00:39:19   node: 55c8f25f.3faa4c
iot-2/type/VITDevice/id/500062/evt/status/fmt/json :
msg.payload : string[51]
"MEDICAL ATTENTION:Soldier has high
body temperature"
01/08/2021, 00:39:20   node: 55c8f25f.3faa4c
iot-2/type/VITDevice/id/500062/evt/status/fmt/json :
msg.payload : Object
▸{ name: "Soldier5", temperature:
107, pulserate: 92, bloodpressure:
"125/86", systolic: 86 … }
01/08/2021, 00:39:24   node: 55c8f25f.3faa4c
iot-2/type/VITDevice/id/500062/evt/status/fmt/json :
msg.payload : Object
▸{ name: "Soldier1", temperature:
104, pulserate: 93, bloodpressure:
"113/72", systolic: 72 … }
01/08/2021, 00:39:25   node: 55c8f25f.3faa4c
iot-2/type/VITDevice/id/500062/evt/status/fmt/json :
msg.payload : string[51]
"MEDICAL ATTENTION:Soldier has high
body temperature"
01/08/2021, 00:39:26   node: 55c8f25f.3faa4c
iot-2/type/VITDevice/id/500062/evt/status/fmt/json :
msg.payload : Object
▸{ name: "Soldier1", temperature:
104, pulserate: 93, bloodpressure:
```

Node red flow screenshot



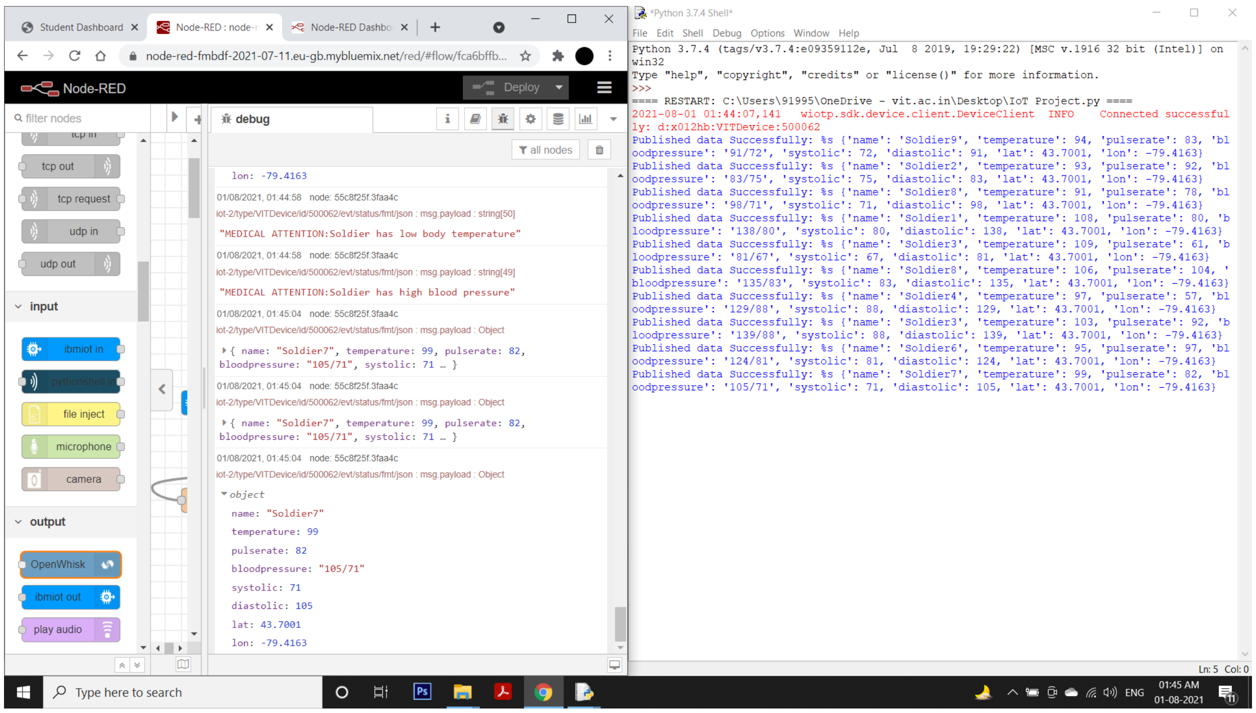
```
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul  8 2019, 19:29:22) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:\Users\91995\OneDrive - vit.ac.in\Desktop\IoT Project.py ====
2021-08-01 01:44:07,141  wiotp.sdk.device.client.DeviceClient  INFO   Connected successfully: d:x012hb:VITDevice:500062
Published data Successfully: %s {'name': 'Soldier9', 'temperature': 94, 'pulserate': 83, 'bloodpressure': '91/72', 'systolic': 72, 'diastolic': 91, 'lat': 43.7001, 'lon': -79.4163}
Published data Successfully: %s {'name': 'Soldier2', 'temperature': 93, 'pulserate': 92, 'bloodpressure': '83/75', 'systolic': 75, 'diastolic': 83, 'lat': 43.7001, 'lon': -79.4163}
Published data Successfully: %s {'name': 'Soldier8', 'temperature': 91, 'pulserate': 78, 'bloodpressure': '98/71', 'systolic': 71, 'diastolic': 98, 'lat': 43.7001, 'lon': -79.4163}
Published data Successfully: %s {'name': 'Soldier1', 'temperature': 108, 'pulserate': 80, 'bloodpressure': '138/80', 'systolic': 80, 'diastolic': 138, 'lat': 43.7001, 'lon': -79.4163}
Published data Successfully: %s {'name': 'Soldier3', 'temperature': 109, 'pulserate': 61, 'bloodpressure': '81/67', 'systolic': 67, 'diastolic': 81, 'lat': 43.7001, 'lon': -79.4163}
Published data Successfully: %s {'name': 'Soldier8', 'temperature': 106, 'pulserate': 104, 'bloodpressure': '135/83', 'systolic': 83, 'diastolic': 135, 'lat': 43.7001, 'lon': -79.4163}
Published data Successfully: %s {'name': 'Soldier4', 'temperature': 97, 'pulserate': 57, 'bloodpressure': '129/88', 'systolic': 88, 'diastolic': 129, 'lat': 43.7001, 'lon': -79.4163}
Published data Successfully: %s {'name': 'Soldier3', 'temperature': 103, 'pulserate': 92, 'bloodpressure': '139/88', 'systolic': 88, 'diastolic': 139, 'lat': 43.7001, 'lon': -79.4163}
Published data Successfully: %s {'name': 'Soldier6', 'temperature': 95, 'pulserate': 97, 'bloodpressure': '124/81', 'systolic': 81, 'diastolic': 124, 'lat': 43.7001, 'lon': -79.4163}
Published data Successfully: %s {'name': 'Soldier7', 'temperature': 99, 'pulserate': 82, 'bloodpressure': '105/71', 'systolic': 71, 'diastolic': 105, 'lat': 43.7001, 'lon': -79.4163}
Published data Successfully: %s {'name': 'Soldier1', 'temperature': 107, 'pulserate': 57, 'bloodpressure': '140/89', 'systolic': 89, 'diastolic': 140, 'lat': 43.7001, 'lon': -79.4163}
Published data Successfully: %s {'name': 'Soldier9', 'temperature': 104, 'pulserate': 113, 'bloodpressure': '83/70', 'systolic': 70, 'diastolic': 83, 'lat': 43.7001, 'lon': -79.4163}
Published data Successfully: %s {'name': 'Soldier2', 'temperature': 93, 'pulserate': 53, 'bloodpressure': '96/68', 'systolic': 68, 'diastolic': 96, 'lat': 43.7001, 'lon': -79.4163}
Published data Successfully: %s {'name': 'Soldier2', 'temperature': 110, 'pulserate': 116, 'bloodpressure': '129/81', 'systolic': 81, 'diastolic': 129, 'lat': 43.7001, 'lon': -79.4163}
Published data Successfully: %s {'name': 'Soldier5', 'temperature': 101, 'pulserate': 107, 'bloodpressure': '126/83', 'systolic': 83, 'diastolic': 126, 'lat': 43.7001, 'lon': -79.4163}
Published data Successfully: %s {'name': 'Soldier6', 'temperature': 98, 'pulserate': 70, 'bloodpressure': '105/65', 'systolic': 65, 'diastolic': 105, 'lat': 43.7001, 'lon': -79.4163}
Published data Successfully: %s {'name': 'Soldier6', 'temperature': 104, 'pulserate': 40, 'bloodpressure': '140/80', 'systolic': 80, 'diastolic': 140, 'lat': 43.7001, 'lon': -79.4163}
Published data Successfully: %s {'name': 'Soldier3', 'temperature': 108, 'pulserate': 94, 'bloodpressure': '121/83', 'systolic': 83, 'diastolic': 121, 'lat': 43.7001, 'lon': -79.4163}
Published data Successfully: %s {'name': 'Soldier2', 'temperature': 104, 'pulserate': 96, 'bloodpressure': '125/87', 'systolic': 87, 'diastolic': 125, 'lat': 43.7001, 'lon': -79.4163}
Published data Successfully: %s {'name': 'Soldier7', 'temperature': 99, 'pulserate': 93, 'bloodpressure': '81/74', 'systolic': 74, 'diastolic': 81, 'lat': 43.7001, 'lon': -79.4163}
```

Python shell



```
01/08/2021, 00:39:18   node: 55c8f25f.3faa4c
iot-2/type/VITDevice/id/500062/evt/status/fmt/json : msg.payload : Object
▸{ name: "Soldier5", temperature: 107, pulserate: 92,
bloodpressure: "125/86", systolic: 86 … }
01/08/2021, 00:39:19   node: 55c8f25f.3faa4c
iot-2/type/VITDevice/id/500062/evt/status/fmt/json : msg.payload : string[51]
"MEDICAL ATTENTION:Soldier has high body temperature"
01/08/2021, 00:39:20   node: 55c8f25f.3faa4c
iot-2/type/VITDevice/id/500062/evt/status/fmt/json : msg.payload : Object
▾object
  name: "Soldier5"
  temperature: 107
  pulserate: 92
  bloodpressure: "125/86"
  systolic: 86
  diastolic: 125
  lat: 17.5043
  lon: 78.5426
01/08/2021, 00:39:24   node: 55c8f25f.3faa4c
iot-2/type/VITDevice/id/500062/evt/status/fmt/json : msg.payload : Object
▸{ name: "Soldier1", temperature: 104, pulserate: 93,
bloodpressure: "113/72", systolic: 72 … }
01/08/2021, 00:39:25   node: 55c8f25f.3faa4c
iot-2/type/VITDevice/id/500062/evt/status/fmt/json : msg.payload : string[51]
"MEDICAL ATTENTION:Soldier has high body temperature"
01/08/2021, 00:39:26   node: 55c8f25f.3faa4c
```

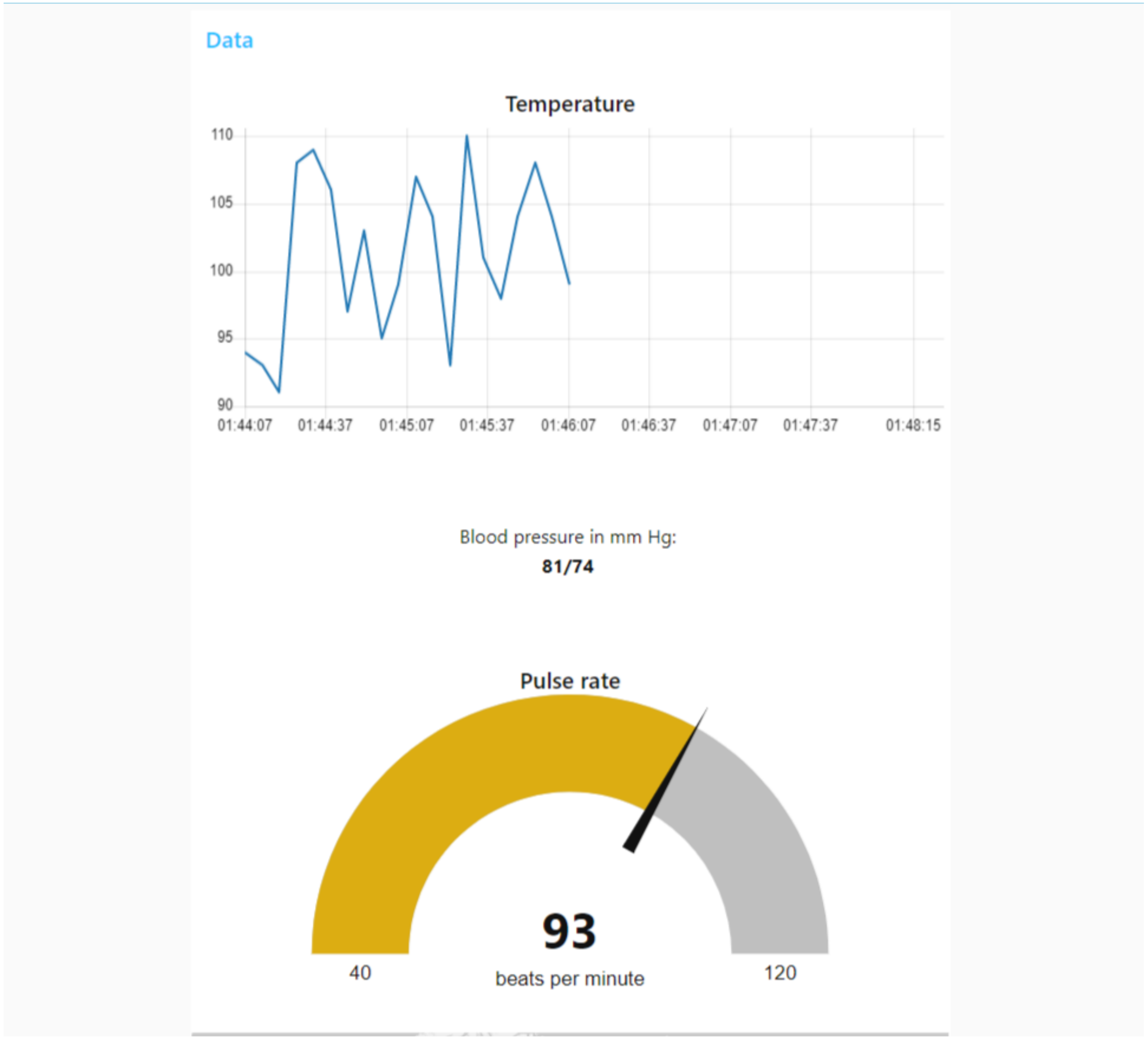
Node red debug messages



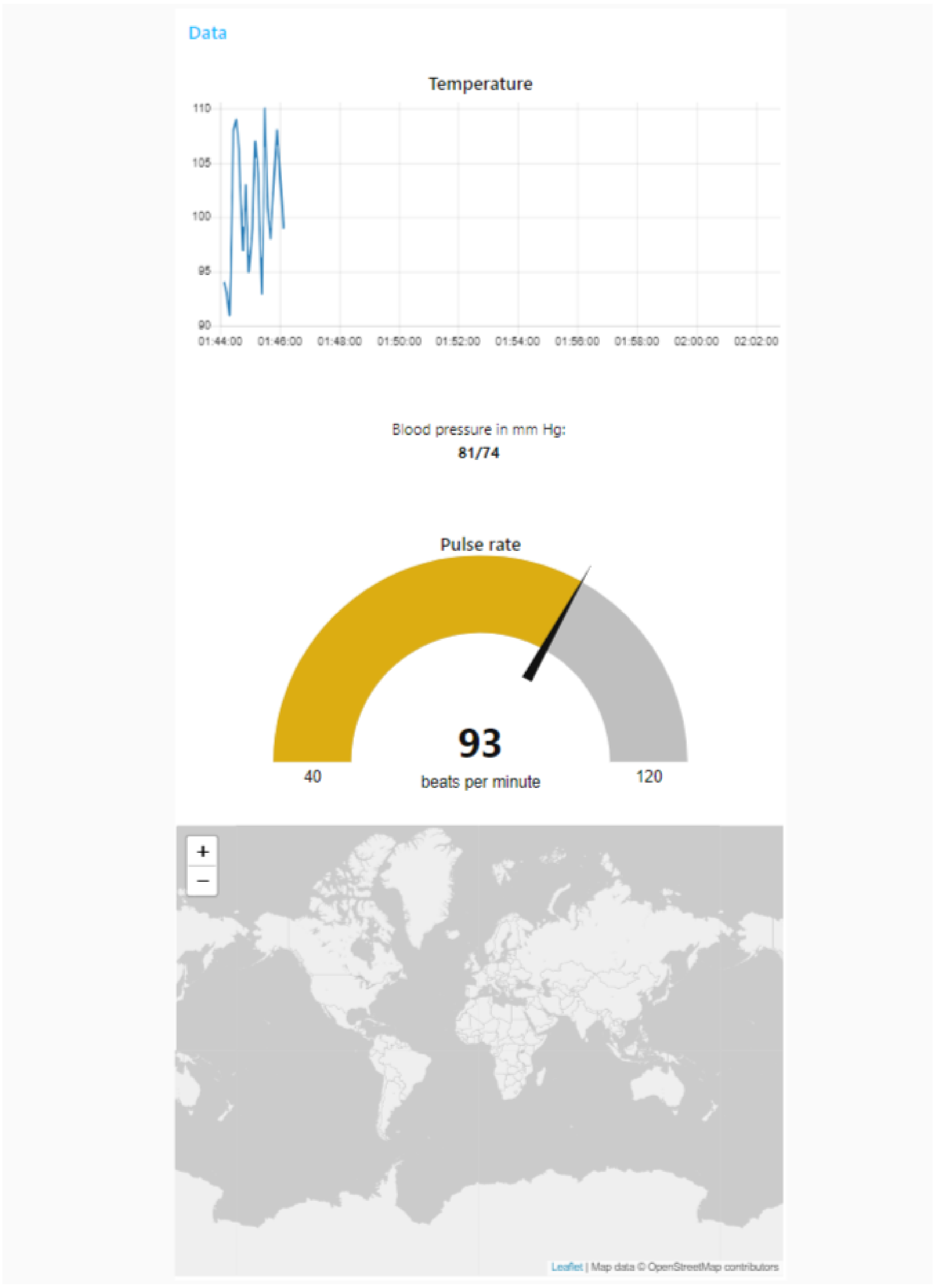
```
  lon: -79.4163
01/08/2021, 01:44:58   node: 55c8f25f.3faa4c
iot-2/type/VITDevice/id/500062/evt/status/fmt/json : msg.payload : string[50]
"MEDICAL ATTENTION:Soldier has low body temperature"
01/08/2021, 01:44:58   node: 55c8f25f.3faa4c
iot-2/type/VITDevice/id/500062/evt/status/fmt/json : msg.payload : string[49]
"MEDICAL ATTENTION:Soldier has high blood pressure"
01/08/2021, 01:45:04   node: 55c8f25f.3faa4c
iot-2/type/VITDevice/id/500062/evt/status/fmt/json : msg.payload : Object
▸{ name: "Soldier7", temperature: 99, pulserate: 82,
bloodpressure: "105/71", systolic: 71 … }
01/08/2021, 01:45:04   node: 55c8f25f.3faa4c
iot-2/type/VITDevice/id/500062/evt/status/fmt/json : msg.payload : Object
▸{ name: "Soldier7", temperature: 99, pulserate: 82,
bloodpressure: "105/71", systolic: 71 … }
01/08/2021, 01:45:04   node: 55c8f25f.3faa4c
iot-2/type/VITDevice/id/500062/evt/status/fmt/json : msg.payload : Object
▾object
  name: "Soldier7"
  temperature: 99
  pulserate: 82
  bloodpressure: "105/71"
  systolic: 71
  diastolic: 105
  lat: 43.7001
  lon: -79.4163
```

```
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul  8 2019, 19:29:22) [MSC v.1916 32 bit (Intel)] on
win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:\Users\91995\OneDrive - vit.ac.in\Desktop\IoT Project.py ====
2021-08-01 01:44:07,141   wiotp.sdk.device.client.DeviceClient  INFO    Connected successful
ly: d:x012hb:VITDevice:500062
Published data Successfully: %s {'name': 'Soldier9', 'temperature': 94, 'pulserate': 83, 'bl
oodpressure': '91/72', 'systolic': 72, 'diastolic': 91, 'lat': 43.7001, 'lon': -79.4163}
Published data Successfully: %s {'name': 'Soldier2', 'temperature': 93, 'pulserate': 92, 'bl
oodpressure': '83/75', 'systolic': 75, 'diastolic': 83, 'lat': 43.7001, 'lon': -79.4163}
Published data Successfully: %s {'name': 'Soldier8', 'temperature': 91, 'pulserate': 78, 'bl
oodpressure': '98/71', 'systolic': 71, 'diastolic': 98, 'lat': 43.7001, 'lon': -79.4163}
Published data Successfully: %s {'name': 'Soldier1', 'temperature': 108, 'pulserate': 80, 'b
loodpressure': '138/80', 'systolic': 80, 'diastolic': 138, 'lat': 43.7001, 'lon': -79.4163}
Published data Successfully: %s {'name': 'Soldier3', 'temperature': 109, 'pulserate': 61, 'b
loodpressure': '81/67', 'systolic': 67, 'diastolic': 81, 'lat': 43.7001, 'lon': -79.4163}
Published data Successfully: %s {'name': 'Soldier8', 'temperature': 106, 'pulserate': 104, '
bloodpressure': '135/83', 'systolic': 83, 'diastolic': 135, 'lat': 43.7001, 'lon': -79.4163}
Published data Successfully: %s {'name': 'Soldier4', 'temperature': 97, 'pulserate': 57, 'bl
oodpressure': '129/88', 'systolic': 88, 'diastolic': 129, 'lat': 43.7001, 'lon': -79.4163}
Published data Successfully: %s {'name': 'Soldier3', 'temperature': 103, 'pulserate': 92, 'b
loodpressure': '139/88', 'systolic': 88, 'diastolic': 139, 'lat': 43.7001, 'lon': -79.4163}
Published data Successfully: %s {'name': 'Soldier6', 'temperature': 95, 'pulserate': 97, 'bl
oodpressure': '124/81', 'systolic': 81, 'diastolic': 124, 'lat': 43.7001, 'lon': -79.4163}
Published data Successfully: %s {'name': 'Soldier7', 'temperature': 99, 'pulserate': 82, 'bl
oodpressure': '105/71', 'systolic': 71, 'diastolic': 105, 'lat': 43.7001, 'lon': -79.4163}
```

Python shell + Node red debug messages

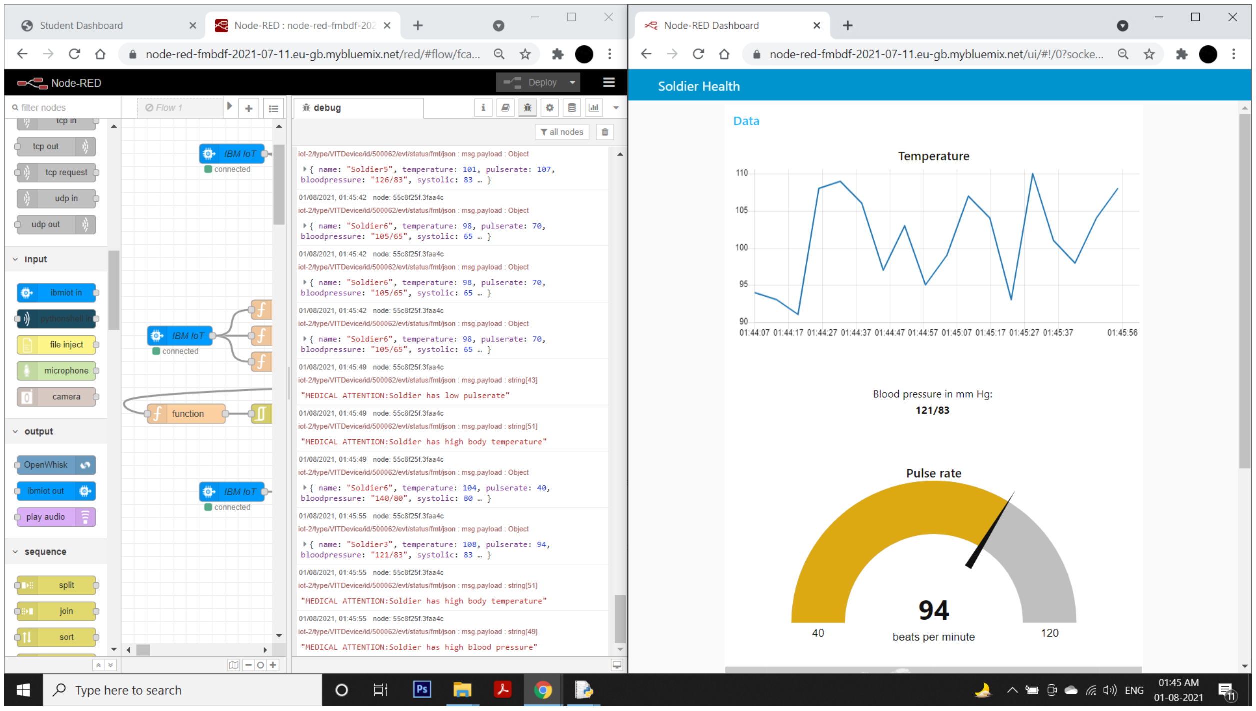


Health parameters (Temperature, Blood Pressure, Pulse rate) visualization


Data

Blood pressure in mm Hg:
81/74

Data visualization (Health paramters + location tracking)



```
iot-2/type/VITDevice/id/500062/evt/status/fmt/json : msg.payload : Object
{ name: "Soldier5", temperature: 101, pulserate: 107,
bloodpressure: "126/83", systolic: 83 … }
01/08/2021, 01:45:42   node: 55c8f25f.3faa4c
iot-2/type/VITDevice/id/500062/evt/status/fmt/json : msg.payload : Object
{ name: "Soldier6", temperature: 98, pulserate: 70,
bloodpressure: "105/65", systolic: 65 … }
01/08/2021, 01:45:42   node: 55c8f25f.3faa4c
iot-2/type/VITDevice/id/500062/evt/status/fmt/json : msg.payload : Object
{ name: "Soldier6", temperature: 98, pulserate: 70,
bloodpressure: "105/65", systolic: 65 … }
01/08/2021, 01:45:42   node: 55c8f25f.3faa4c
iot-2/type/VITDevice/id/500062/evt/status/fmt/json : msg.payload : Object
{ name: "Soldier6", temperature: 98, pulserate: 70,
bloodpressure: "105/65", systolic: 65 … }
01/08/2021, 01:45:49   node: 55c8f25f.3faa4c
iot-2/type/VITDevice/id/500062/evt/status/fmt/json : msg.payload : string[43]
"MEDICAL ATTENTION:Soldier has low pulserate"
01/08/2021, 01:45:49   node: 55c8f25f.3faa4c
iot-2/type/VITDevice/id/500062/evt/status/fmt/json : msg.payload : string[51]
"MEDICAL ATTENTION:Soldier has high body temperature"
01/08/2021, 01:45:49   node: 55c8f25f.3faa4c
iot-2/type/VITDevice/id/500062/evt/status/fmt/json : msg.payload : Object
{ name: "Soldier6", temperature: 104, pulserate: 40,
bloodpressure: "140/80", systolic: 80 … }
01/08/2021, 01:45:55   node: 55c8f25f.3faa4c
iot-2/type/VITDevice/id/500062/evt/status/fmt/json : msg.payload : Object
{ name: "Soldier3", temperature: 108, pulserate: 94,
bloodpressure: "121/83", systolic: 83 … }
01/08/2021, 01:45:55   node: 55c8f25f.3faa4c
iot-2/type/VITDevice/id/500062/evt/status/fmt/json : msg.payload : string[51]
"MEDICAL ATTENTION:Soldier has high body temperature"
01/08/2021, 01:45:55   node: 55c8f25f.3faa4c
iot-2/type/VITDevice/id/500062/evt/status/fmt/json : msg.payload : string[49]
"MEDICAL ATTENTION:Soldier has high blood pressure"
```

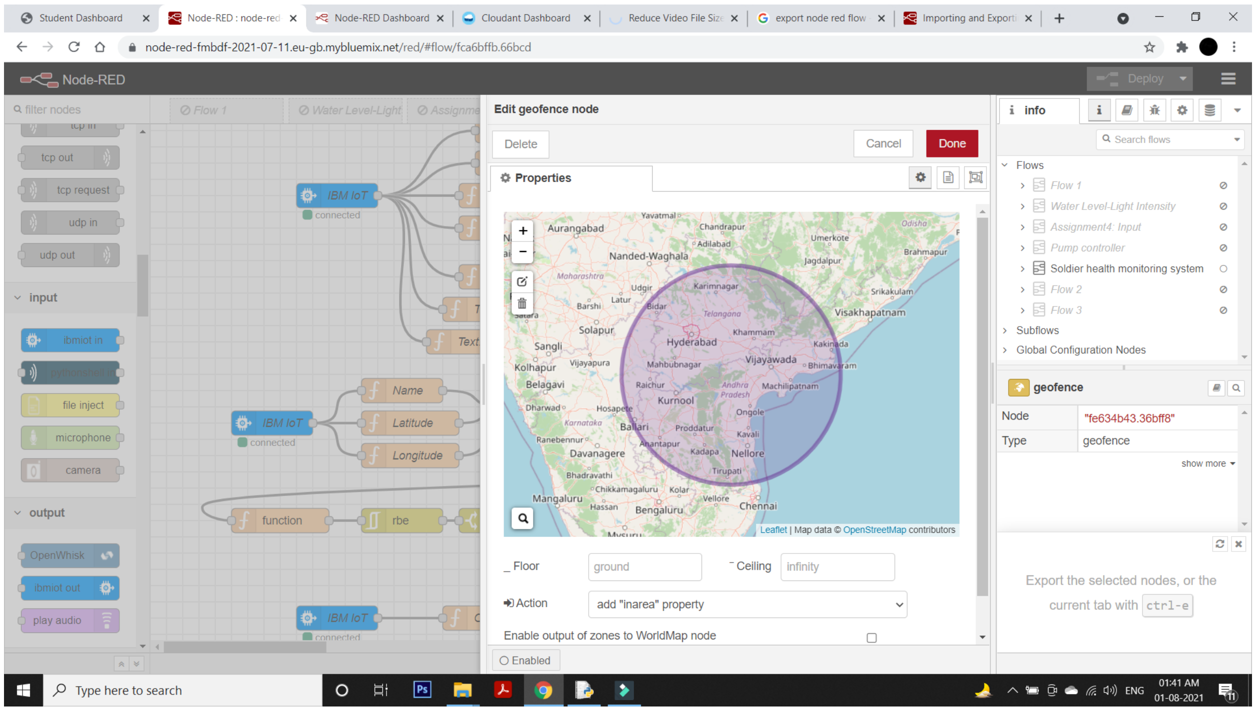
Soldier Health

Data

Blood pressure in mm Hg:
121/83

Pulse rate
94
beats per minute

Node red debug messages + Health parameters visualization



Edit geofence node

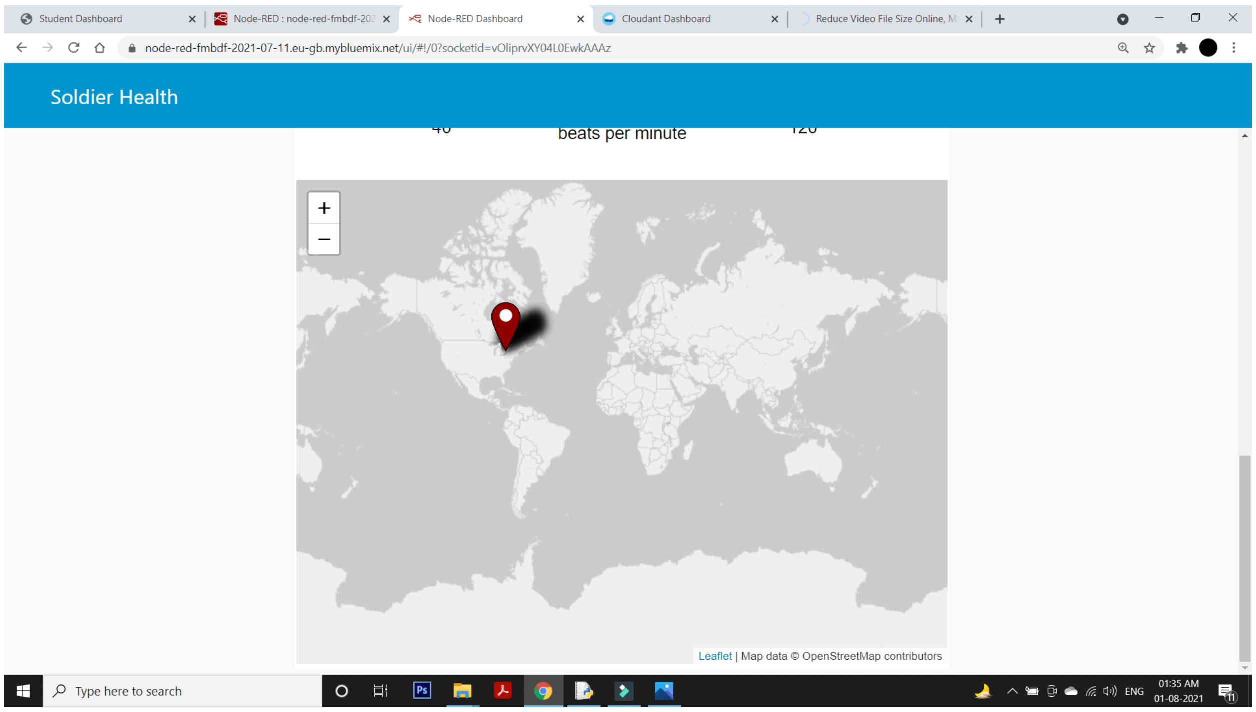
Properties

Floor: ground
Ceiling: infinity
Action: add "inarea" property
Enable output of zones to WorldMap node

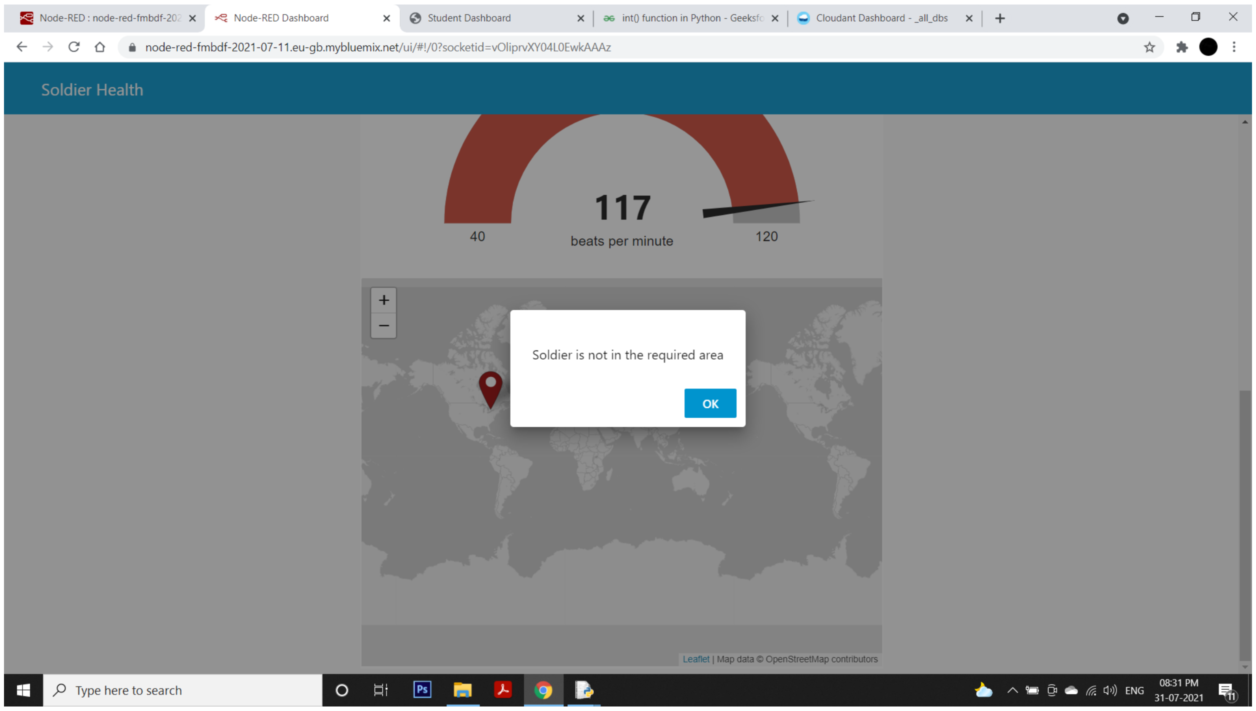
Node: "fe634b43.36bff8"
Type: geofence

Delete
Cancel
Done
Enabled
info
Flows
Flow 1
Water Level-Light Intensity
Assignment4: Input
Pump controller
Soldier health monitoring system
Flow 2
Flow 3
Subflows
Global Configuration Nodes
geofence
show more
Export the selected nodes, or the current tab with ctrl-e

Geofence location

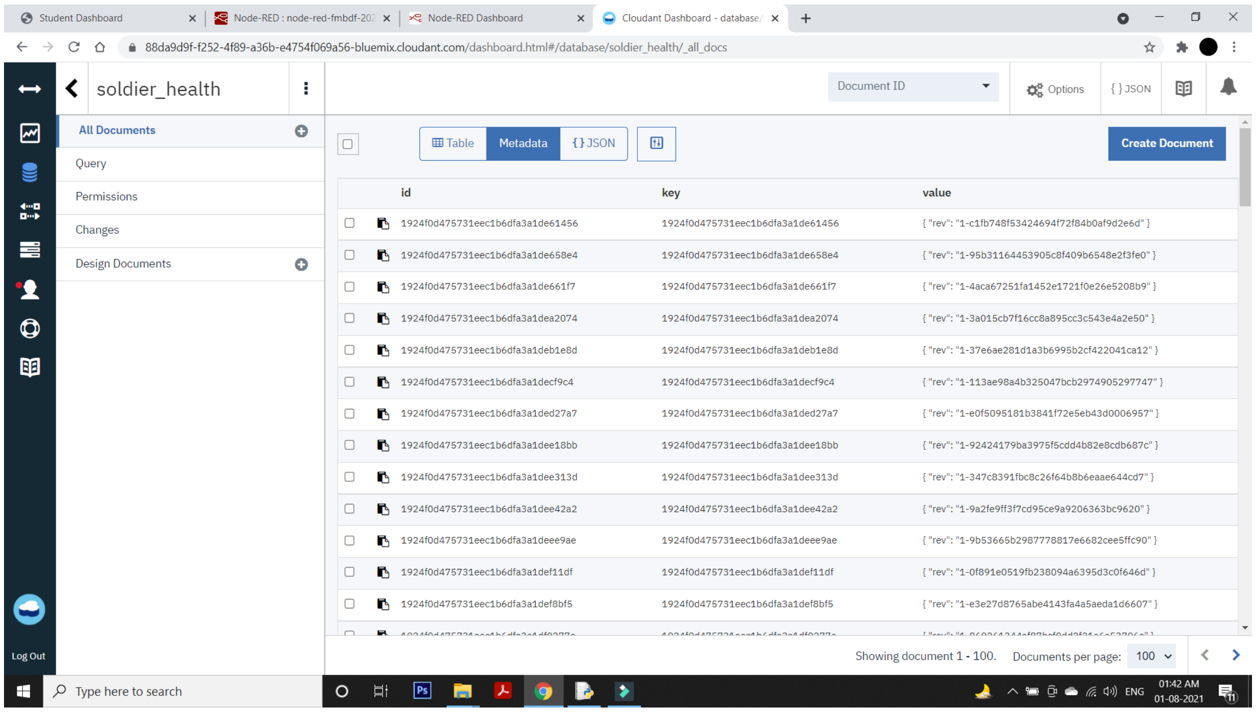


Location tracking visualization



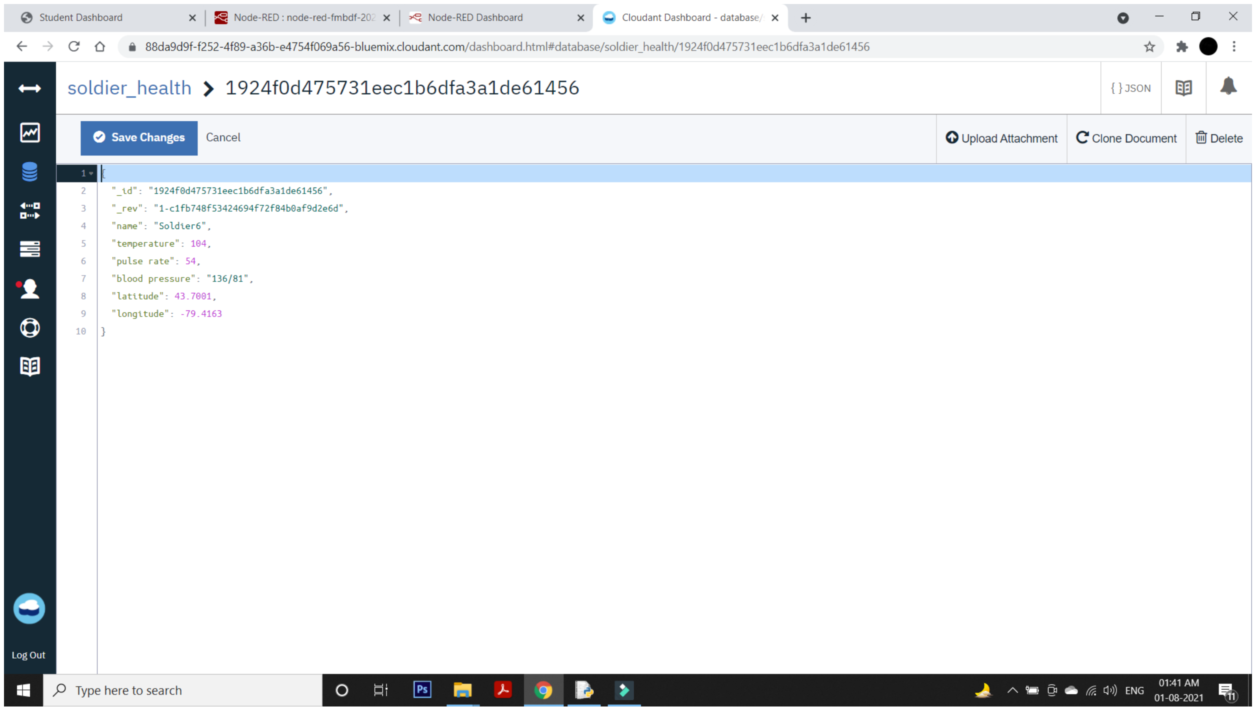
Soldier is not in the required area

Soldier tracking notification message



id	key	value
1924f0d475731eec1b6dfa3a1de61456	1924f0d475731eec1b6dfa3a1de61456	{ "rev": "1-c1fb748f53424694f72f84b0af9d2e6d" }
1924f0d475731eec1b6dfa3a1de658e4	1924f0d475731eec1b6dfa3a1de658e4	{ "rev": "1-95b31164453905c8f409b6548e2f3fe0" }
1924f0d475731eec1b6dfa3a1de661f7	1924f0d475731eec1b6dfa3a1de661f7	{ "rev": "1-4aca67251fa1452e1721f0e26e5208b9" }
1924f0d475731eec1b6dfa3a1dea2074	1924f0d475731eec1b6dfa3a1dea2074	{ "rev": "1-3a015cb7f16cc8a895cc3c543e4a2e50" }
1924f0d475731eec1b6dfa3a1deb1e8d	1924f0d475731eec1b6dfa3a1deb1e8d	{ "rev": "1-37e6ae281d1a3b6995b2cf422041ca12" }
1924f0d475731eec1b6dfa3a1decf9c4	1924f0d475731eec1b6dfa3a1decf9c4	{ "rev": "1-113ae98a4b325047bcb2974905297747" }
1924f0d475731eec1b6dfa3a1ded27a7	1924f0d475731eec1b6dfa3a1ded27a7	{ "rev": "1-e0f5095181b3841f72e5eb43d0006957" }
1924f0d475731eec1b6dfa3a1dee18bb	1924f0d475731eec1b6dfa3a1dee18bb	{ "rev": "1-92424179ba3975f5cdd4b82e8cdb687c" }
1924f0d475731eec1b6dfa3a1dee313d	1924f0d475731eec1b6dfa3a1dee313d	{ "rev": "1-347c8391fbc8c26f64b8b6eaae644cd7" }
1924f0d475731eec1b6dfa3a1dee42a2	1924f0d475731eec1b6dfa3a1dee42a2	{ "rev": "1-9a2fe9ff3f7cd95ce9a9206363bc9620" }
1924f0d475731eec1b6dfa3a1deee9ae	1924f0d475731eec1b6dfa3a1deee9ae	{ "rev": "1-9b53665b2987778817e6682cee5ffc90" }
1924f0d475731eec1b6dfa3a1def11df	1924f0d475731eec1b6dfa3a1def11df	{ "rev": "1-0f891e0519fb238094a6395d3c0f646d" }
1924f0d475731eec1b6dfa3a1def8bf5	1924f0d475731eec1b6dfa3a1def8bf5	{ "rev": "1-e3e27d8765abe4143fa4a5aeda1d6607" }

Soldier_health database in cloudant DB


Student Dashboard
Node-RED : node-red-fmbdf-202
Node-RED Dashboard
Cloudant Dashboard - database/
88da9d9f-f252-4f89-a36b-e4754f069a56-bluemix.cloudant.com/dashboard.html#database/soldier_health/1924f0d475731eec1b6dfa3a1de61456
soldier_health > 1924f0d475731eec1b6dfa3a1de61456
{ } JSON
Save Changes
Cancel
Upload Attachment
Clone Document
Delete
Log Out
Type here to search
ENG
01:41 AM
01-08-2021

```
{
  "_id": "1924f0d475731eec1b6dfa3a1de61456",
  "_rev": "1-c1fb748f53424694f72f84b0af9d2e6d",
  "name": "Soldier6",
  "temperature": 104,
  "pulse rate": 54,
  "blood pressure": "136/81",
  "latitude": 43.7001,
  "longitude": -79.4163
}
```

A document in soldier_health database