8/12/2020

IOT ANALYTICS IN HEALTH MONITORING

DONE BY,
NEHA SATISH

1) INTRODUCTION

1.1 Overview

Health Monitor is used to check if a person is following a healthy regime. It is used by collecting data on regular components of health such as body temperature or water level in the body.

1.2 Purpose

More skilful patient administration can help use the assets of the clinic all the more astutely and set aside cash. It is simpler to utilize the framework for patients and clinical experts. The checking framework is particularly helpful to screen patients with interminable sicknesses. Most ailments are serious, so it is important to screen the condition of the patient while at home, and rapidly react if well-being markers compound.

2) LITERATURE SURVEY

2.1 Existing problem

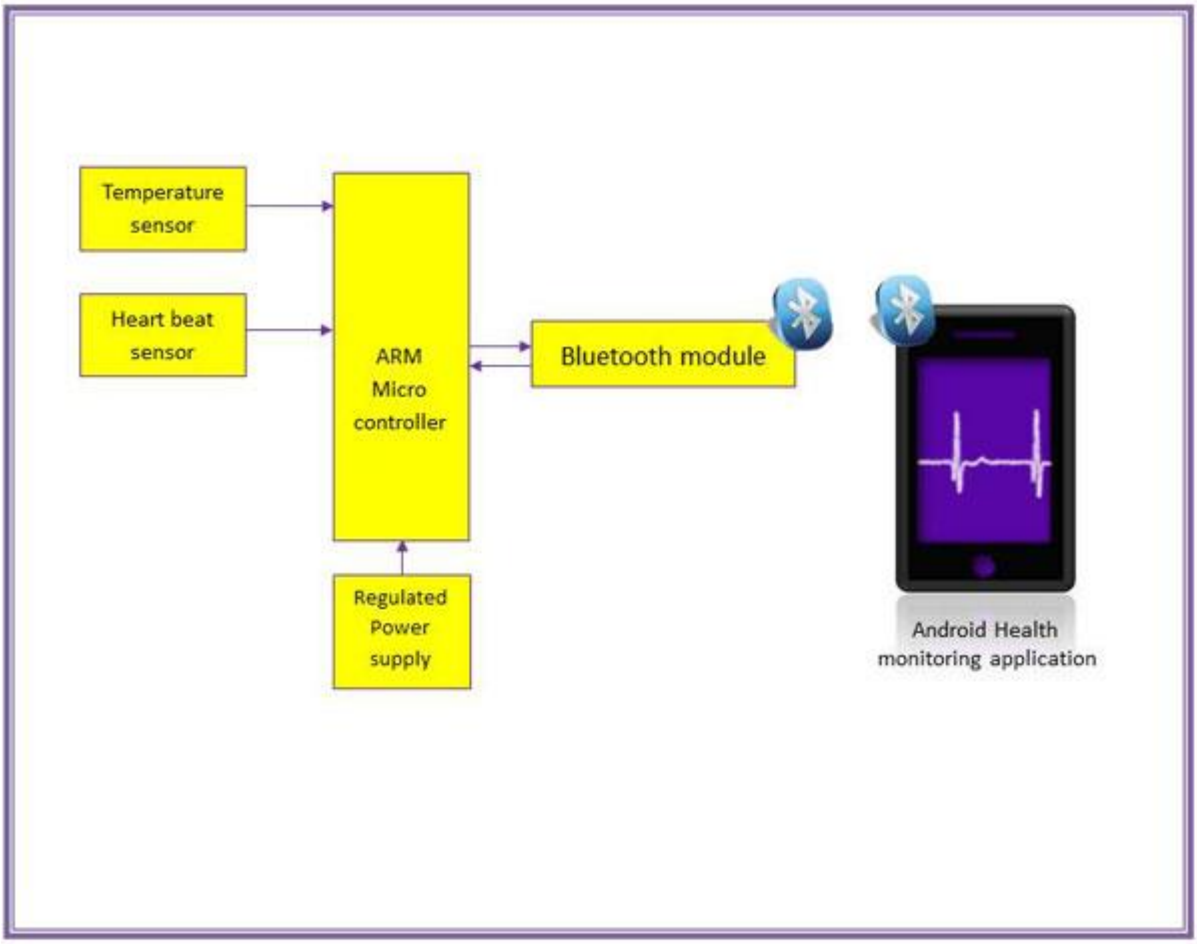
Many people run to the hospitals for a regular health check-up. Due to the Covid19 Pandemic, most of the patients couldn’t visit the hospital for regular check-ups. This has increased panic in people and also ailments which people do not find cure to.

2.2 Proposed solution

The health Monitor is built to give people a better exposure to hoe they can have a regular check-up at home. Using the health monitor they can check their body temperature and know if their water levels are proper, and if not get appropriate cure.

3) THEORITICAL ANALYSIS

3.1 Block diagram



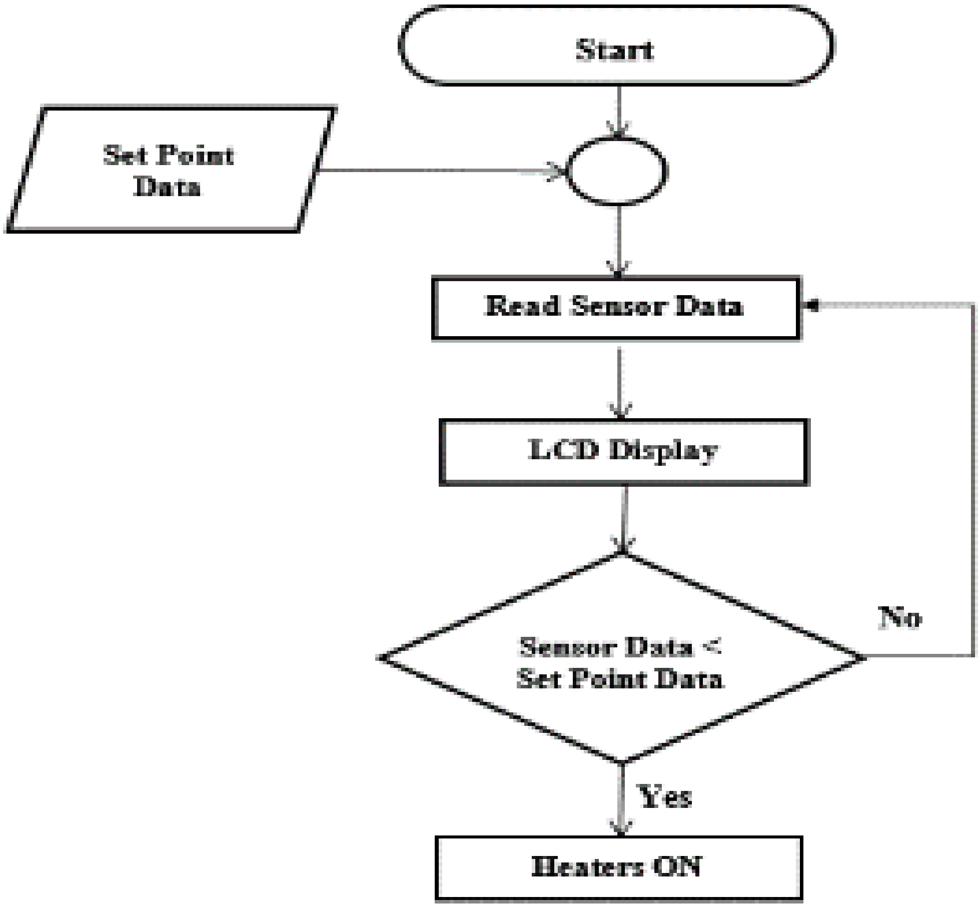
3.2 Software designing

Software components used are : 1.IBM cloud 2.IBM IOT platform 3.IBM Watson 4.node-red 5.Tinker cad 6.Python IDLE 7.MIT app inventor

4) EXPERIMENTAL INVESTIGATIONS

People who felt sick, immediately checked their temperature using the health monitor and found cure.

5) FLOWCHART



6) RESULT

Health monitor was built using IBM watson assistant, IOT sensor simulator to check the health parameters of a person.

7) ADVANTAGES & DISADVANTAGES

PROS:

- Many people can be helped at the same time in a short while.
- Health Monitor cannot spread virus and diseases to the people.

CONS:

- The Health monitor requires Internet and power supply. In most rural and remote areas, reliable source of power is a major challenge.

8) APPLICATIONS

- Allows sending data from patients to health professionals in real time
- Improves patients’ lifestyle.
- Makes healthcare more available
- Saves money.
- Timely detection and action for specific conditions which require quick attention.
- Assisted and rapid diagnoses that may help arrive at logical conclusions.
- Reduction in hospitalization and related time, effort, and costs.
- Better adherence to the medication schedule.
- Home or familiar premises may be more amenable for several patients than hospitals.

9) CONCLUSION

Health monitor app is available on your phones. Any time, any place it can be accessible. It is very feasible and eco-friendly. This helps a lot of people to get their regular medical check-ups done and follow a healthy regime.

10) FUTURE SCOPE

With more High-end hardware and software, the Health Monitor can be customized and can be upgraded and improved for more efficiency and success rate.

11) BIBILOGRAPHY

a.Github

b.ubuntupit

c.youtube

d.smart internz

12) APPENDIX

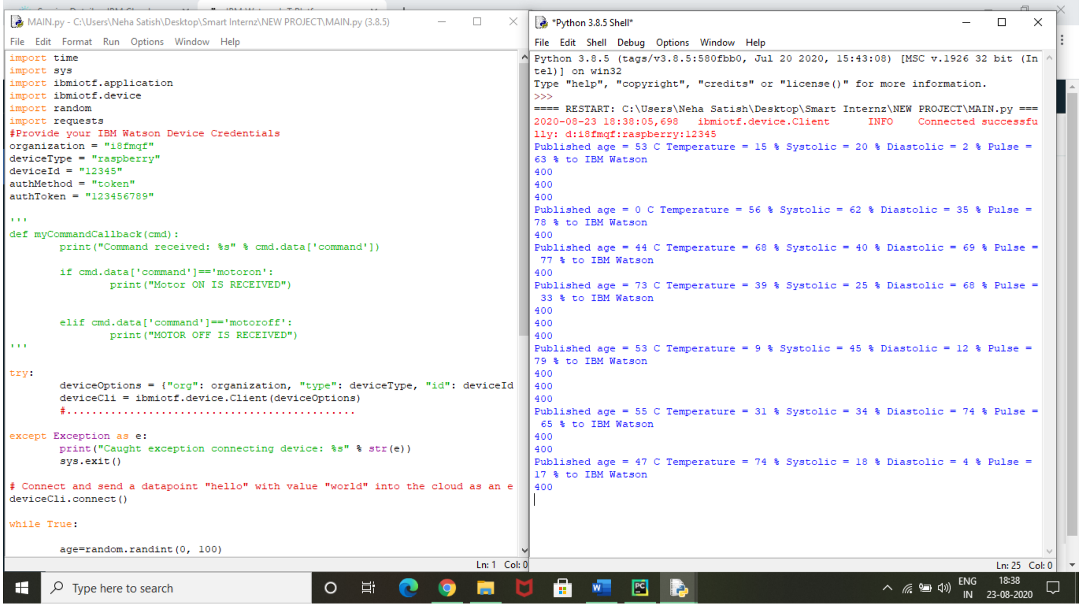
SOUURCE CODE:

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
import requests
#Provide your IBM Watson Device Credentials
organization = "i8fmqf"
deviceType = "raspberry"
deviceId = "12345"
authMethod = "token"
authToken = "123456789"

'''
def myCommandCallback(cmd):
      print("Command received: %s" % cmd.data['command'])

      if cmd.data['command']=='motoron':
            print("Motor ON IS RECEIVED")


      elif cmd.data['command']=='motoroff':
            print("MOTOR OFF IS RECEIVED")
'''

try:
  deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method": authMethod,
"auth-token": authToken}
  deviceCli = ibmiotf.device.Client(deviceOptions)
  #..............................................

except Exception as e:
  print("Caught exception connecting device: %s" % str(e))
  sys.exit()

# Connect and send a datapoint "hello" with value "world" into the cloud as an event of type "greeting" 10
times
deviceCli.connect()

while True:

      age=random.randint(0, 100)
      Temperature =random.randint(0, 100)
      Systolic =random.randint(0, 100)
      Diastolic =random.randint(0, 100)
      Pulse =random.randint(0, 100)
      #Send Temperature & Humidity to IBM Watson
      data = { 'age' : age, 'Temperature' : Temperature, 'Systolic':Systolic, 'Diastolic':Diastolic, 'Pulse':Pulse }
      #print (data)
      def myOnPublishCallback():
         print ("Published age = %s C" % age, "Temperature = %s %%" % Temperature, "Systolic = %s %%"
% Systolic, "Diastolic = %s %%" % Diastolic, "Pulse = %s %%" % Pulse, "to IBM Watson")

      success = deviceCli.publishEvent("health", "json", data, qos=0, on_publish=myOnPublishCallback)
```

```
    if(Temperature<50):
        r =
requests.get('https://www.fast2sms.com/dev/bulk?authorization=q5lnG4atKdro1TQyMLiCgpFSJfVPBe
mw8uEkRWU9IAc3Y2Xxbhpwz6Nl0iu3bf9EoQJFW2C7VsMG4XID&sender_id=FSTSMS&message=Tempe
rature is low... please switch on motor.&language=english&route=p&numbers=9611405758')
        print(r.status_code)
    if(Systolic<50):
        r =
requests.get('https://www.fast2sms.com/dev/bulk?authorization=q5lnG4atKdro1TQyMLiCgpFSJfVPBe
mw8uEkRWU9IAc3Y2Xxbhpwz6Nl0iu3bf9EoQJFW2C7VsMG4XID&sender_id=FSTSMS&message=Systoli
c is low... please switch on motor.&language=english&route=p&numbers=9611405758')
        print(r.status_code)
    if(Diastolic<50):
        r =
requests.get('https://www.fast2sms.com/dev/bulk?authorization=q5lnG4atKdro1TQyMLiCgpFSJfVPBe
mw8uEkRWU9IAc3Y2Xxbhpwz6Nl0iu3bf9EoQJFW2C7VsMG4XID&sender_id=FSTSMS&message=Diastol
ic is low... please switch on motor.&language=english&route=p&numbers=9611405758')
        print(r.status_code)
    if(Pulse<50):
        r =
requests.get('https://www.fast2sms.com/dev/bulk?authorization=q5lnG4atKdro1TQyMLiCgpFSJfVPBe
mw8uEkRWU9IAc3Y2Xxbhpwz6Nl0iu3bf9EoQJFW2C7VsMG4XID&sender_id=FSTSMS&message=Pulse is
low... please switch on motor.&language=english&route=p&numbers=9611405758')
        print(r.status_code)
    if not success:
      print("Not connected to IoTF")
    time.sleep(2)

    #deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud
deviceCli.disconnect()
```

B. UI output Screenshot.

Code:



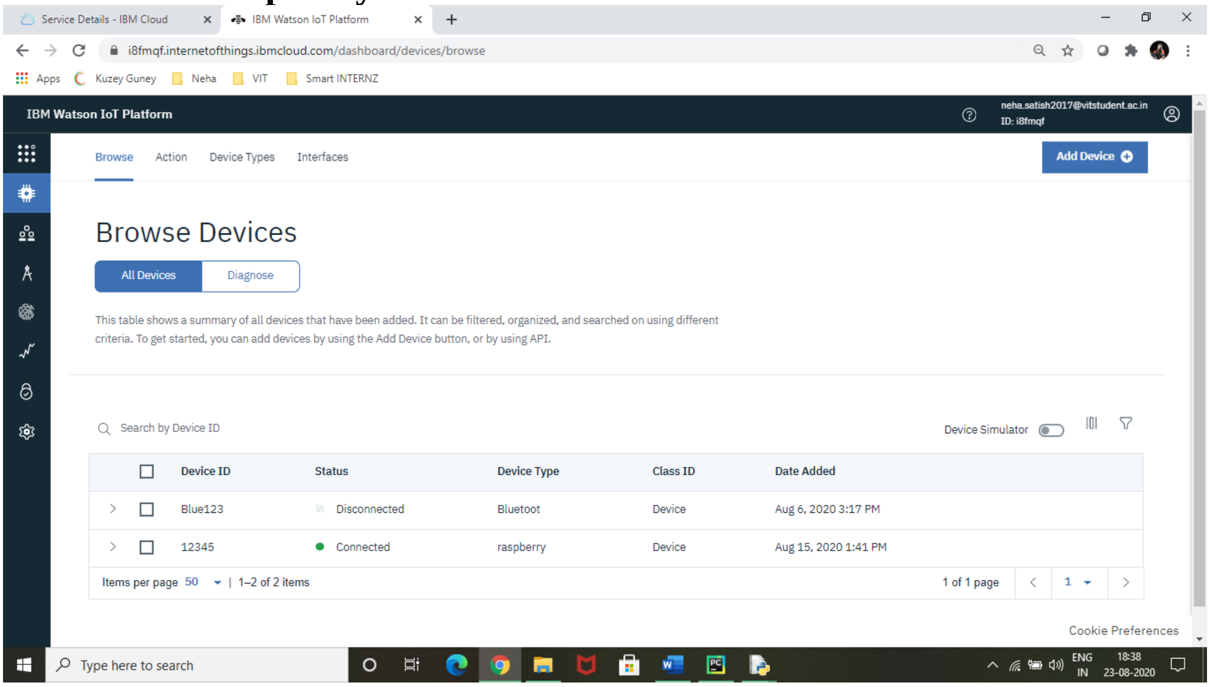
MAIN.py - C:\Users\Neha Satish\Desktop\Smart Internz\NEW PROJECT\MAIN.py (3.8.5)

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
import requests
#Provide your IBM Watson Device Credentials
organization = "i8fmqf"
deviceType = "raspberry"
deviceId = "12345"
authMethod = "token"
authToken = "123456789"

'''
def myCommandCallback(cmd):
        print("Command received: %s" % cmd.data['command'])

        if cmd.data['command']=='motoron':
                print("Motor ON IS RECEIVED")


        elif cmd.data['command']=='motoroff':
                print("MOTOR OFF IS RECEIVED")
'''

try:
        deviceOptions = {"org": organization, "type": deviceType, "id": deviceId
        deviceCli = ibmiotf.device.Client(deviceOptions)
        #..............................................

except Exception as e:
        print("Caught exception connecting device: %s" % str(e))
        sys.exit()

# Connect and send a datapoint "hello" with value "world" into the cloud as an e
deviceCli.connect()

while True:

        age=random.randint(0, 100)
```

Python 3.8.5 Shell

```
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (In
tel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:\Users\Neha Satish\Desktop\Smart Internz\NEW PROJECT\MAIN.py ===
2020-08-23 18:38:05,698   ibmiotf.device.Client      INFO    Connected successfu
lly: d:i8fmqf:raspberry:12345
Published age = 53 C Temperature = 15 % Systolic = 20 % Diastolic = 2 % Pulse =
63 % to IBM Watson
400
400
400
Published age = 0 C Temperature = 56 % Systolic = 62 % Diastolic = 35 % Pulse =
78 % to IBM Watson
400
Published age = 44 C Temperature = 68 % Systolic = 40 % Diastolic = 69 % Pulse =
 77 % to IBM Watson
400
Published age = 73 C Temperature = 39 % Systolic = 25 % Diastolic = 68 % Pulse =
 33 % to IBM Watson
400
400
400
Published age = 53 C Temperature = 9 % Systolic = 45 % Diastolic = 12 % Pulse =
79 % to IBM Watson
400
400
400
Published age = 55 C Temperature = 31 % Systolic = 34 % Diastolic = 74 % Pulse =
 65 % to IBM Watson
400
400
Published age = 47 C Temperature = 74 % Systolic = 18 % Diastolic = 4 % Pulse =
17 % to IBM Watson
400
```

IBM Device: raspberry


Service Details - IBM Cloud
IBM Watson IoT Platform
i8fmqf.internetofthings.ibmcloud.com/dashboard/devices/browse
Apps
Kuzey Guney
Neha
VIT
Smart INTERNZ
neha.satish2017@vitstudent.ac.in
ID: i8fmqf
Type here to search
ENG
IN
18:38
23-08-2020

IBM Watson IoT Platform

Browse Action Device Types Interfaces

Add Device

Browse Devices

All Devices Diagnose

This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.

Search by Device ID

Device Simulator

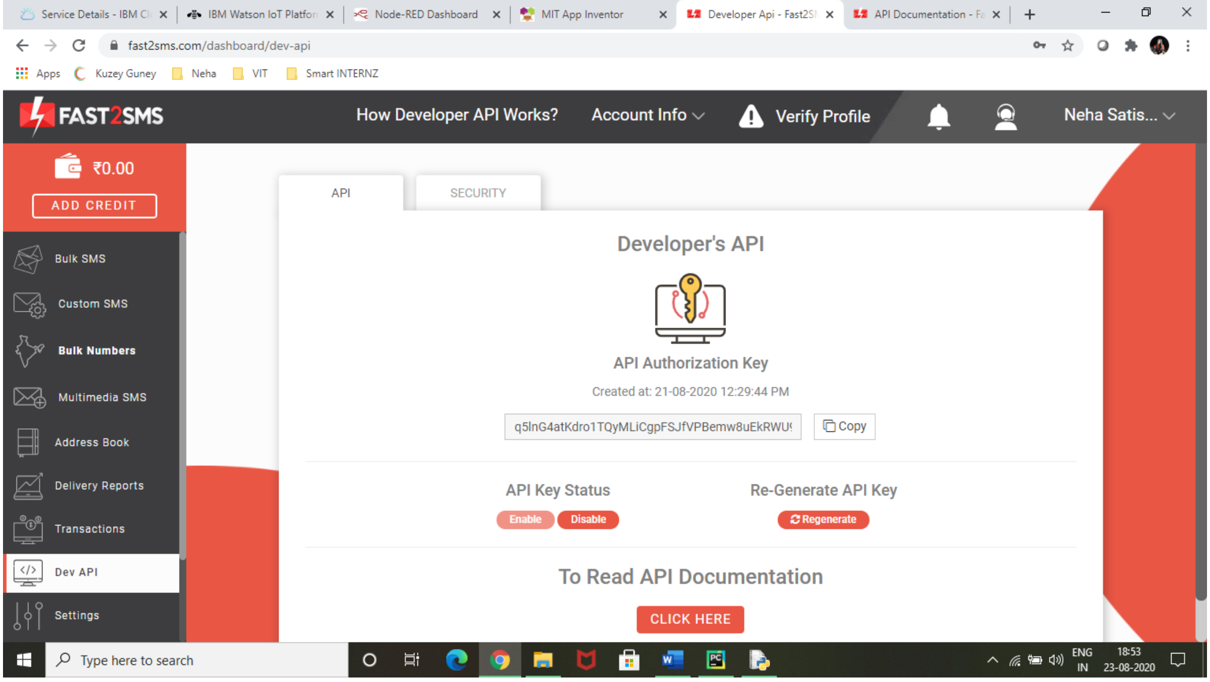
	Device ID	Status	Device Type	Class ID	Date Added
□	Blue123	Disconnected	Bluetoot	Device	Aug 6, 2020 3:17 PM
□	12345	● Connected	raspberry	Device	Aug 15, 2020 1:41 PM

Items per page 50 | 1–2 of 2 items

1 of 1 page 1

Cookie Preferences

Fast2sms: to send sms to the patients mobile number



API SECURITY

Developer's API

API Authorization Key

Created at: 21-08-2020 12:29:44 PM

q5lnG4atKdro1TQyMLiCgpFSJfVPBemw8uEkRWU(Copy

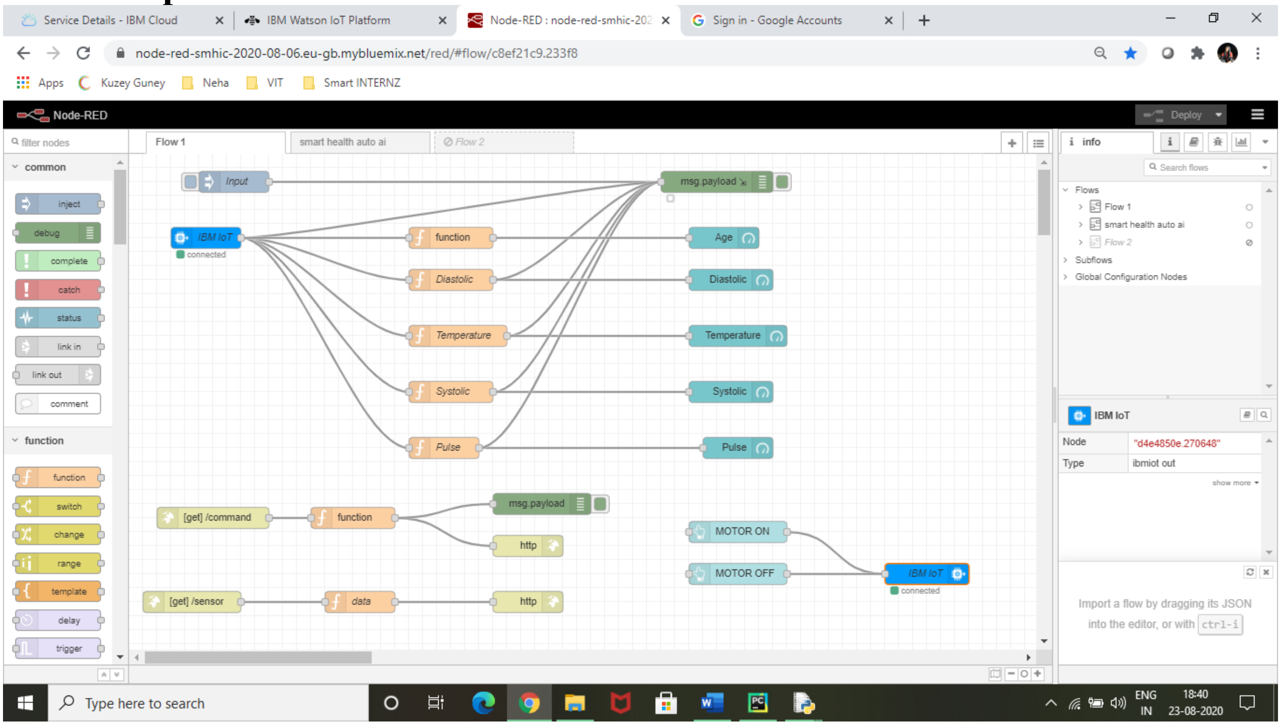
API Key Status: Enable Disable

Re-Generate API Key: Regenerate

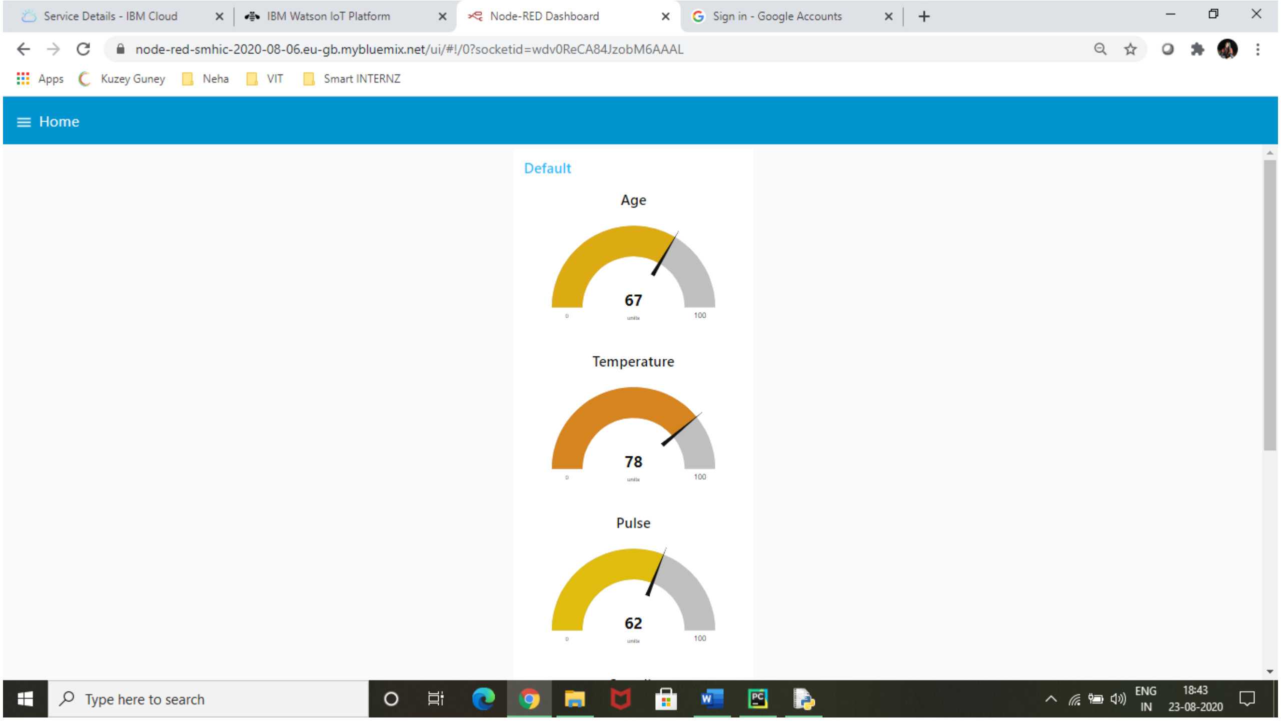
To Read API Documentation

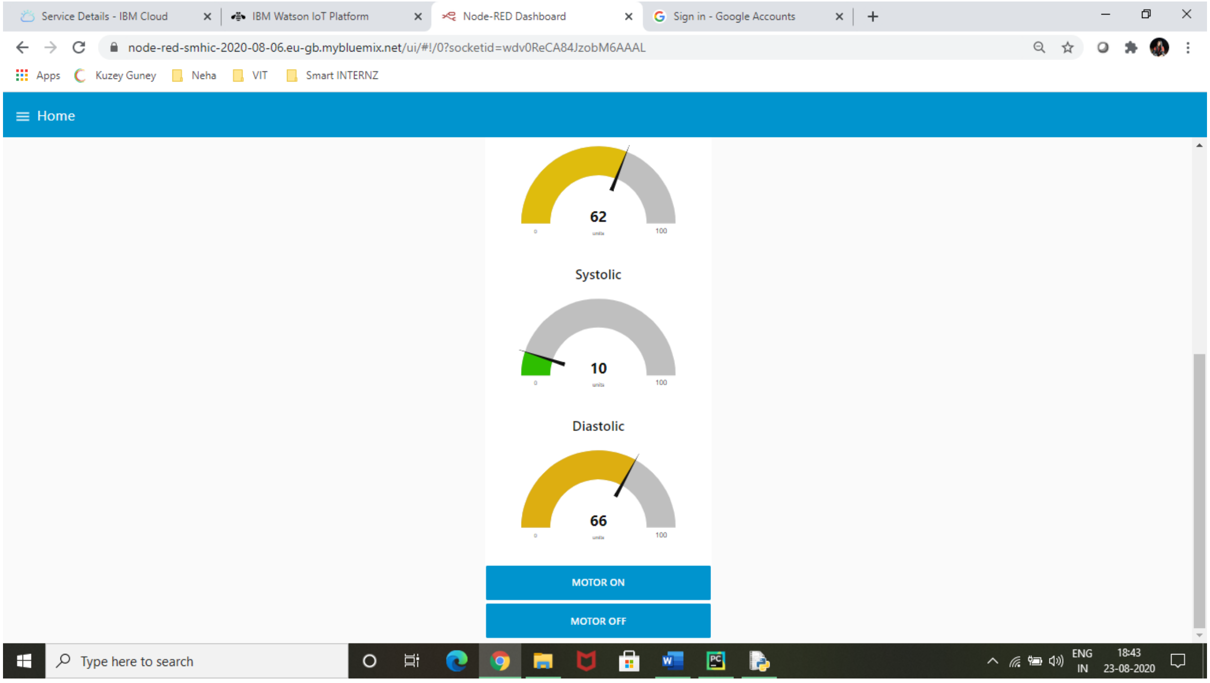
CLICK HERE

Node-red platform:

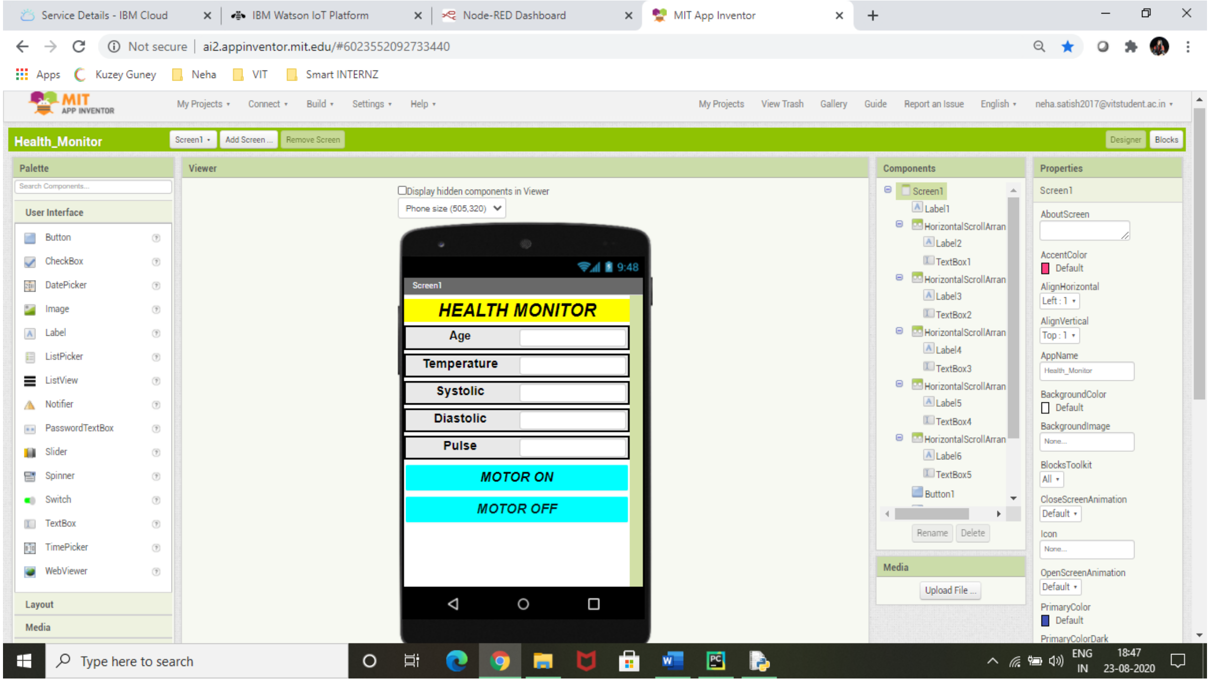


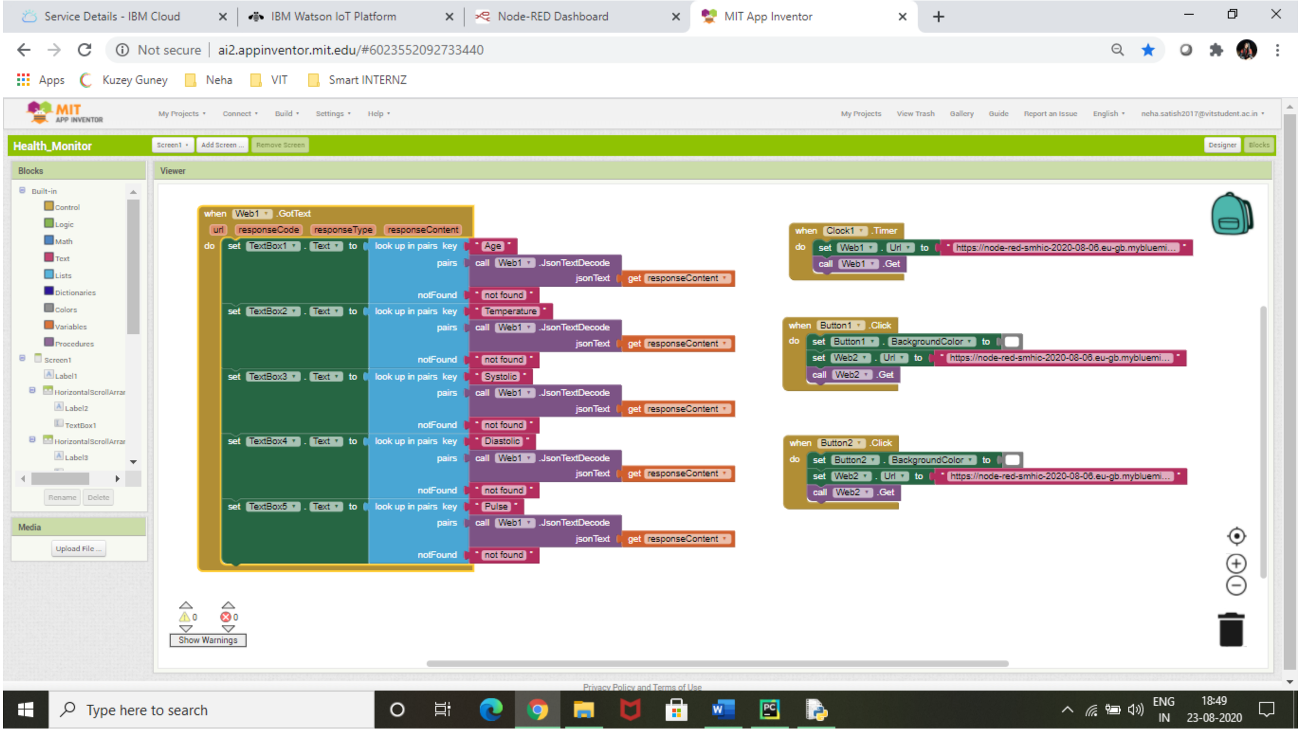
Node-red UI:




Service Details - IBM Cloud
IBM Watson IoT Platform
Node-RED Dashboard
Sign in - Google Accounts
node-red-smhic-2020-08-06.eu-gb.mybluemix.net/ui/#!/0?socketid=wdv0ReCA84JzobM6AAAL
Home
62
Systolic
10
Diastolic
66
MOTOR ON
MOTOR OFF
Type here to search
18:43
23-08-2020

MIT APP:


Service Details - IBM Cloud
IBM Watson IoT Platform
Node-RED Dashboard
MIT App Inventor
ai2.appinventor.mit.edu/#6023552092733440
Health_Monitor
Palette
Viewer
Components
Properties
Screen1
HEALTH MONITOR
Age
Temperature
Systolic
Diastolic
Pulse
MOTOR ON
MOTOR OFF
Type here to search
18:47
23-08-2020


Service Details - IBM Cloud
IBM Watson IoT Platform
Node-RED Dashboard
MIT App Inventor
Not secure | ai2.appinventor.mit.edu/#6023552092733440
Apps Kuzey Guney Neha VIT Smart INTERNZ
MIT APP INVENTOR
My Projects Connect Build Settings Help
My Projects View Trash Gallery Guide Report an Issue English neha.satish2017@vitstudent.ac.in
Health_Monitor
Screen1 Add Screen ... Remove Screen
Designer Blocks
Blocks
Built-in
Control
Logic
Math
Text
Lists
Dictionaries
Colors
Variables
Procedures
Screen1
Label1
HorizontalScrollArrar
Label2
TextBox1
HorizontalScrollArrar
Label3
Rename Delete
Media
Upload File ...
Viewer
0 0
Show Warnings
Privacy Policy and Terms of Use
Type here to search
ENG IN 18:49 23-08-2020

```
when Web1 .GotText
  url  responseCode  responseType  responseContent
do  set TextBox1 . Text to  look up in pairs key  " Age "
                                           pairs  call Web1 .JsonTextDecode
                                                               jsonText  get responseContent
                                        notFound  " not found "
    set TextBox2 . Text to  look up in pairs key  " Temperature "
                                           pairs  call Web1 .JsonTextDecode
                                                               jsonText  get responseContent
                                        notFound  " not found "
    set TextBox3 . Text to  look up in pairs key  " Systolic "
                                           pairs  call Web1 .JsonTextDecode
                                                               jsonText  get responseContent
                                        notFound  " not found "
    set TextBox4 . Text to  look up in pairs key  " Diastolic "
                                           pairs  call Web1 .JsonTextDecode
                                                               jsonText  get responseContent
                                        notFound  " not found "
    set TextBox5 . Text to  look up in pairs key  " Pulse "
                                           pairs  call Web1 .JsonTextDecode
                                                               jsonText  get responseContent
                                        notFound  " not found "
```

```
when Clock1 .Timer
do  set Web1 . Url to  " https://node-red-smhic-2020-08-06.eu-gb.mybluemi... "
    call Web1 .Get

when Button1 .Click
do  set Button1 . BackgroundColor to  □
    set Web2 . Url to  " https://node-red-smhic-2020-08-06.eu-gb.mybluemi... "
    call Web2 .Get

when Button2 .Click
do  set Button2 . BackgroundColor to  □
    set Web2 . Url to  " https://node-red-smhic-2020-08-06.eu-gb.mybluemi... "
    call Web2 .Get
```