A Project Report

On

IOT Analytics In Health Monitoring

By:

B SAIEESH KUMAR

SRIHITHA REDDY VINJAMURI

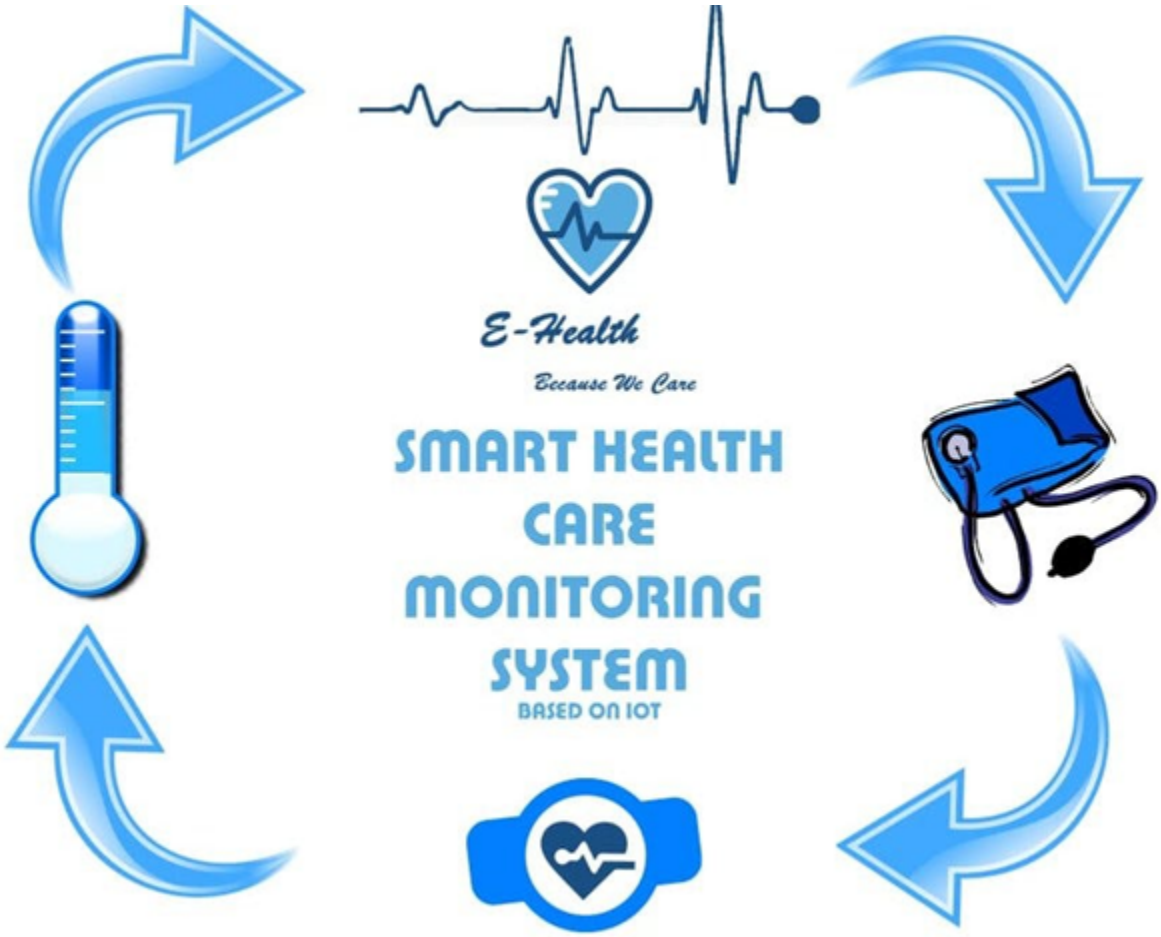
BOLLARAM PAVAN KALYAN

as an interns at

https://smartinternz.com/rsip2020

ON:

INTERNET OF THINGS


E-Health
Because We Care
SMART HEALTH
CARE
MONITORING
SYSTEM
BASED ON IOT

INTRODUCTION:

Overview:-

The objective of this report is to propose IOT analytics Health Monitoring facilitated to the world, Healthcare applications are most important. In general, IoT has been widely used to interconnect the advanced medical resources and to offer smart and effective healthcare services to the people.Real-time monitoring via connected devices can save lives in event of a medical emergency like heart failure, diabetes, asthma attacks, etc.

The IoT device collects and transfers health data: blood pressure, oxygen and blood sugar levels, weight, and ECGs.

Connectivity protocols: Bluetooth LE, Wi-Fi, Z-wave, ZigBee, and other modern protocols, healthcare personnel can change the way they spot illness and ailments in patients and can also innovate revolutionary ways of treatment.

Purpose:

IoT devices such as reduce much manual work which a doctor has to do during patient charting.

It is powered by voice commands and captures the patient's data.It makes the patient's data readily accessible for review.

It saves around doctors' work per week. IoT enables us to collect a massive amount of data about the patient's illness which would have taken many years if we collected it manually.

Scope:

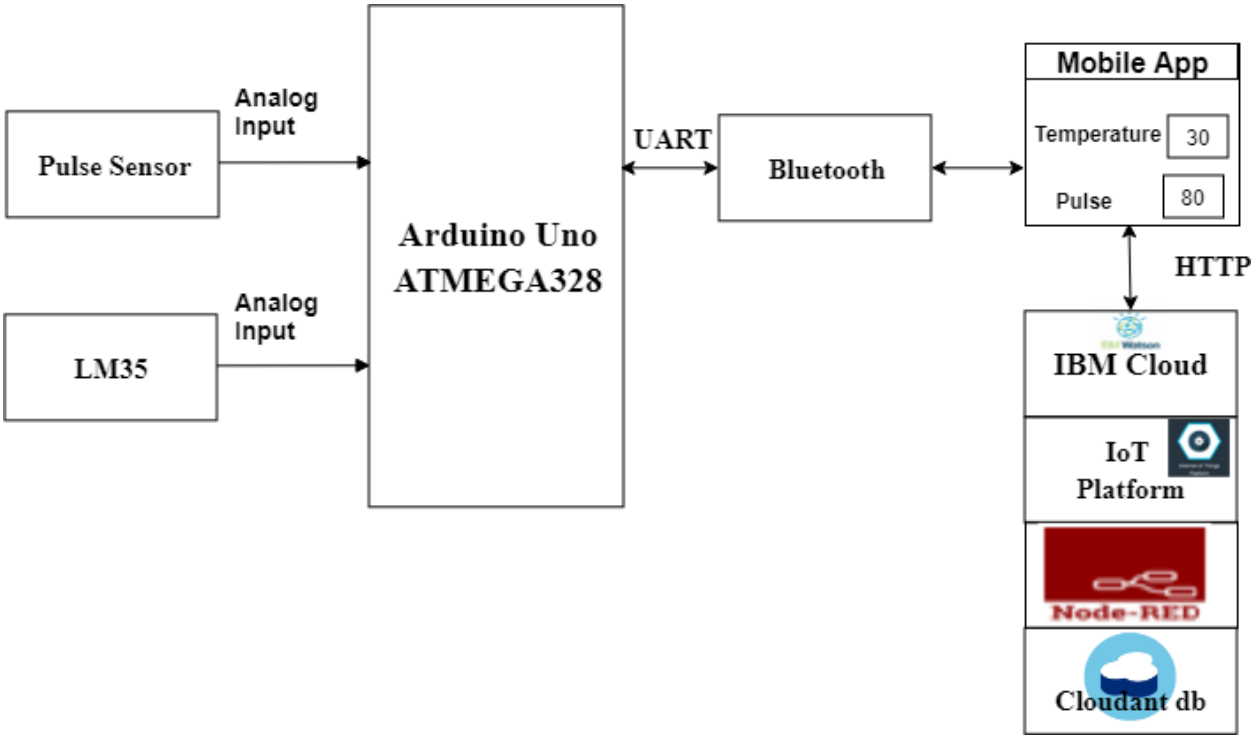
IoT can automate patient care workflow with the help healthcare mobility solution and other new technologies, and next-gen healthcare facilities.

IoT in healthcare enables interoperability, machine-to-machine communication, information exchange, and data movement that makes healthcare service delivery effective.

LITERATURE SURVEY:

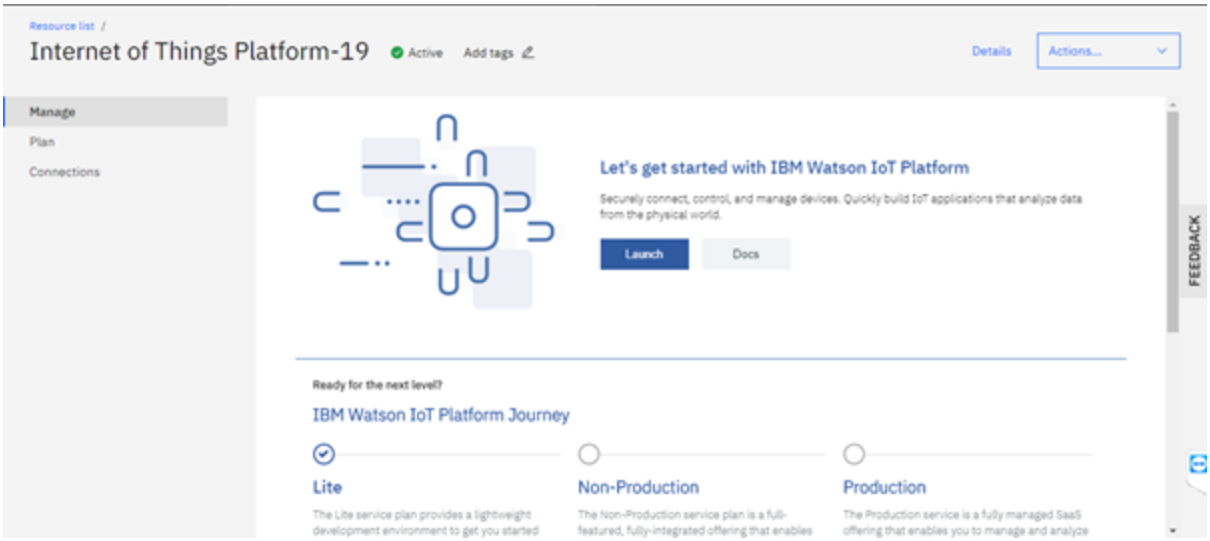
The major problems occurring in the health care are we may face many problems in checking BP, Pulse,Temperature at home but using this device we can get rid of those problems.IoT enables real-time alerting, tracking, and monitoring, which permits hands-on treatments, better accuracy, apt intervention by doctors and improve complete patient care delivery results.

THEORTICAL ANALYSIS:-

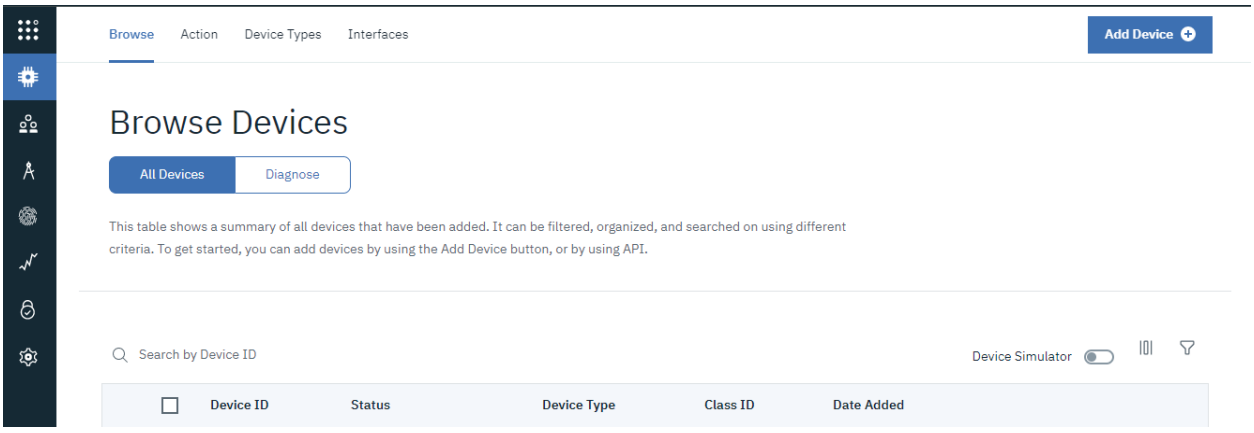


Designing Procedure:-

1. Sign-in to your IBM cloud account from the link cloud.ibm.com. There, go to Catalog and search for IoT in the search bar. Then select Internet of Things platform and subscribe for the desired plan and click create. Now, in the menu, go to Resource List - Services - Internet of Things Platform and then click Launch, as shown below:

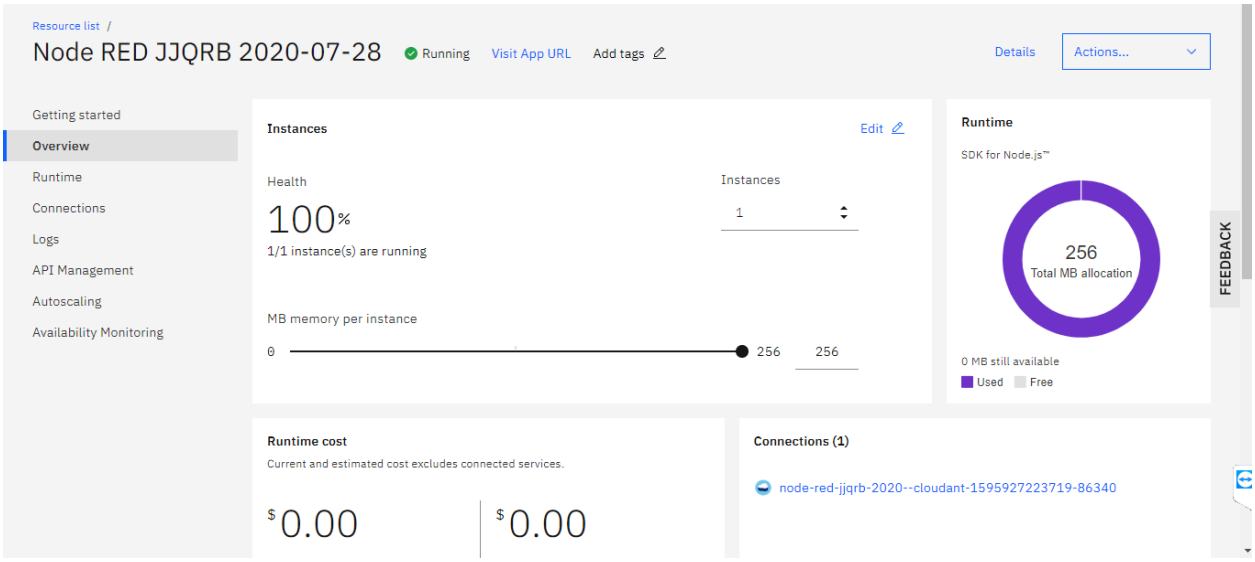


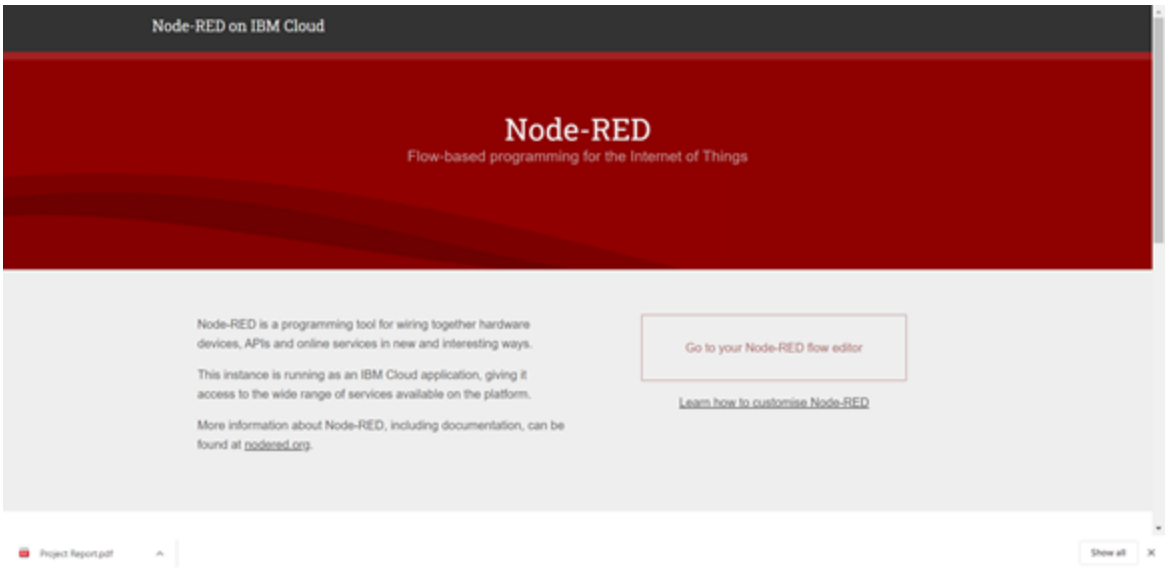
1. Now in the Watson IoT platform, click on the Add Device button at the top right corner, as shown below, to create a device to get the soil conditions from the sensor (simulator).



Make a note of the device credentials given during the device creation for further uses.

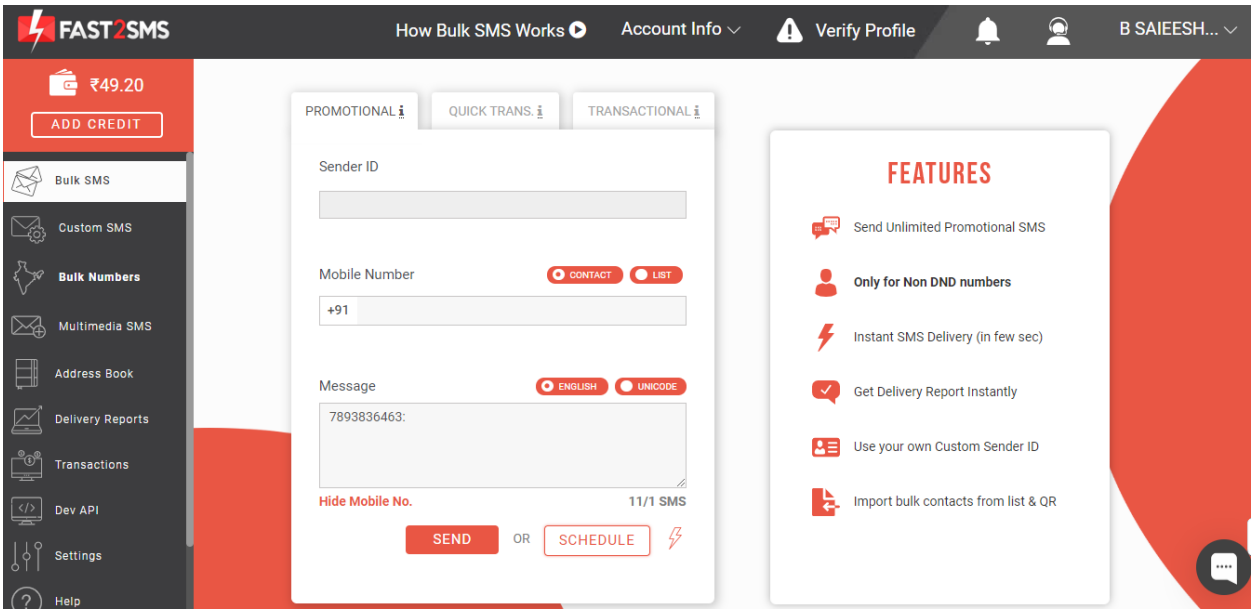
(3) Now, create Node-Red as shown here.And click on Visit App URL.Then you can see Main Page of Node-Red .





create a fast2sms account:(for sending alert messages):-
now create a fast2sms account to send alert messages touser

search fast2sms in bowser ,open that website and create an account there



code snippet for sending sensor data to the watsoniot platform and for sending the alert messages to the user

note:we dont have sensors to send the data to the cloud,so we send sensor data with python code

The following code is the code used for this task

IN THE BELOW CODE ENTER THE CREDENTIALS OF THE DEVICE THAT U CREATED IN THE WATSON IOT PLATFORM

PYTHON CODE:

```
#try to use jupyter notebook while executing the program wait for atleast 40 seconds for the entire program to run

import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
import requests
#Provide your IBM Watson Device Credentials
organization = "ytissp"
deviceType = "raspberrypi"
deviceId = "12345678"
authMethod = "token"
authToken = "12345678"


def myCommandCallback(cmd):
        print("Command received: %s" % cmd.data)#Commands


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() #try with different values



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

while True:
        temp=round(random.uniform(96,102),1)   # print temperature
        pul=random.randint(10,180)   # print pulse
        #enter your mobile number
        if (temp <= 97) or (temp >=100):
            r=requests.get('https://www.fast2sms.com/dev/bulk?authorization=sDNoOkViHbSXQAxwBU0q753IPKdJ6L4mYatgCFZy8fElnWGR1hPZK2UbvViosYwWEA0B9haIOLGjx4X3&sender_id=
        if (pul <= 50) or (pul >= 110):
            r=requests.get('https://www.fast2sms.com/dev/bulk?authorization=sDNoOkViHbSXQAxwBU0q753IPKdJ6L4mYatgCFZy8fElnWGR1hPZK2UbvViosYwWEA0B9haIOLGjx4X3&sender_id=
        data = { 'temp': temp , 'pul':pul}
        #print (data)
        def myOnPublishCallback():
            print ("Published  temp = %s " % temp,"pul = %s" % pul,"to IBM Watson")

        success = deviceCli.publishEvent("kitchen", "json", data, qos=0, on_publish=myOnPublishCallback)
        if not success:
            print("Not connected to IoTF")
        time.sleep(10)

        deviceCli.commandCallback = myCommandCallback


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

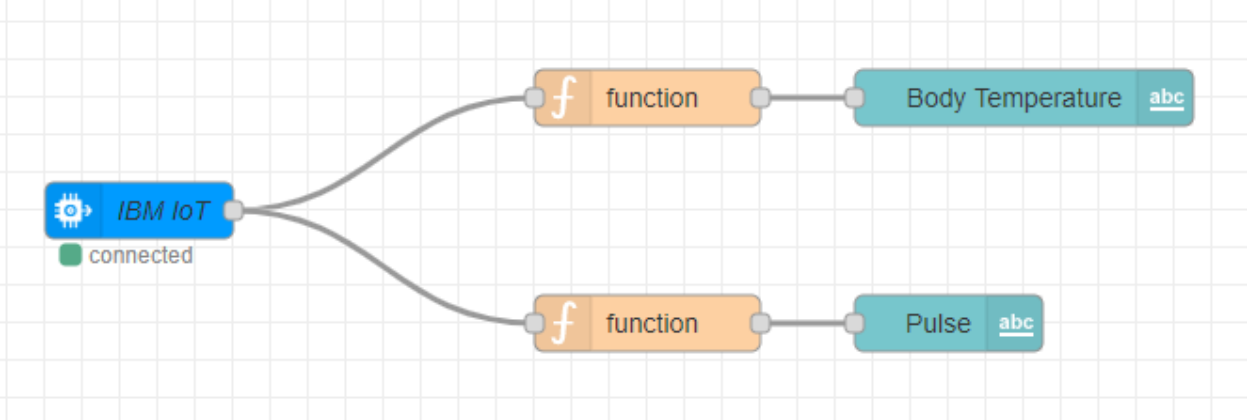
CREATE THE NODERED FLOW TO GET DATA FROM THE DEVICE AND HTTP REQUEST TO COMMUNICATE

WITH THE MOBILE APP

we need to create 2 flows to do this task

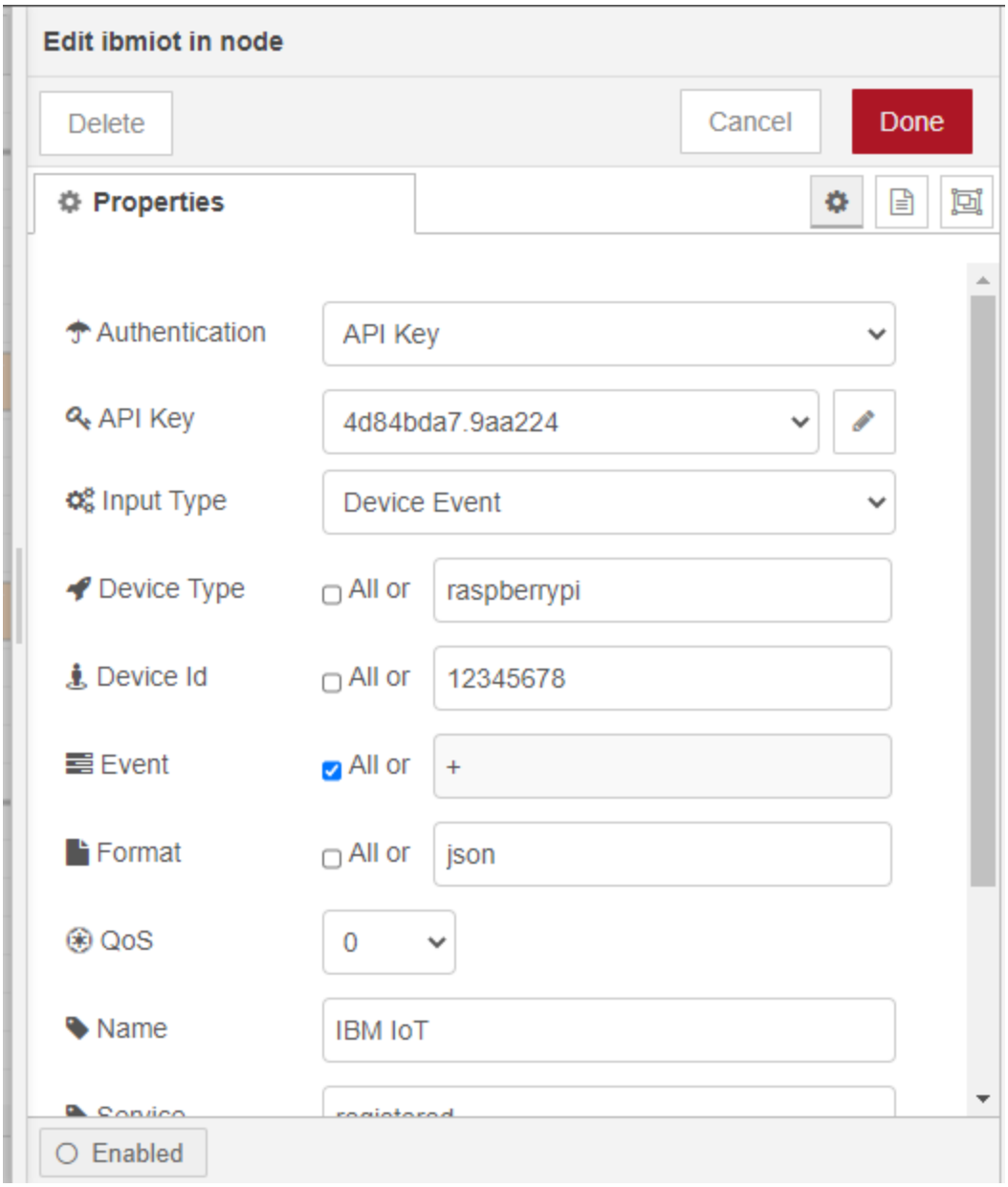
FLOW 1:

to get data from the device

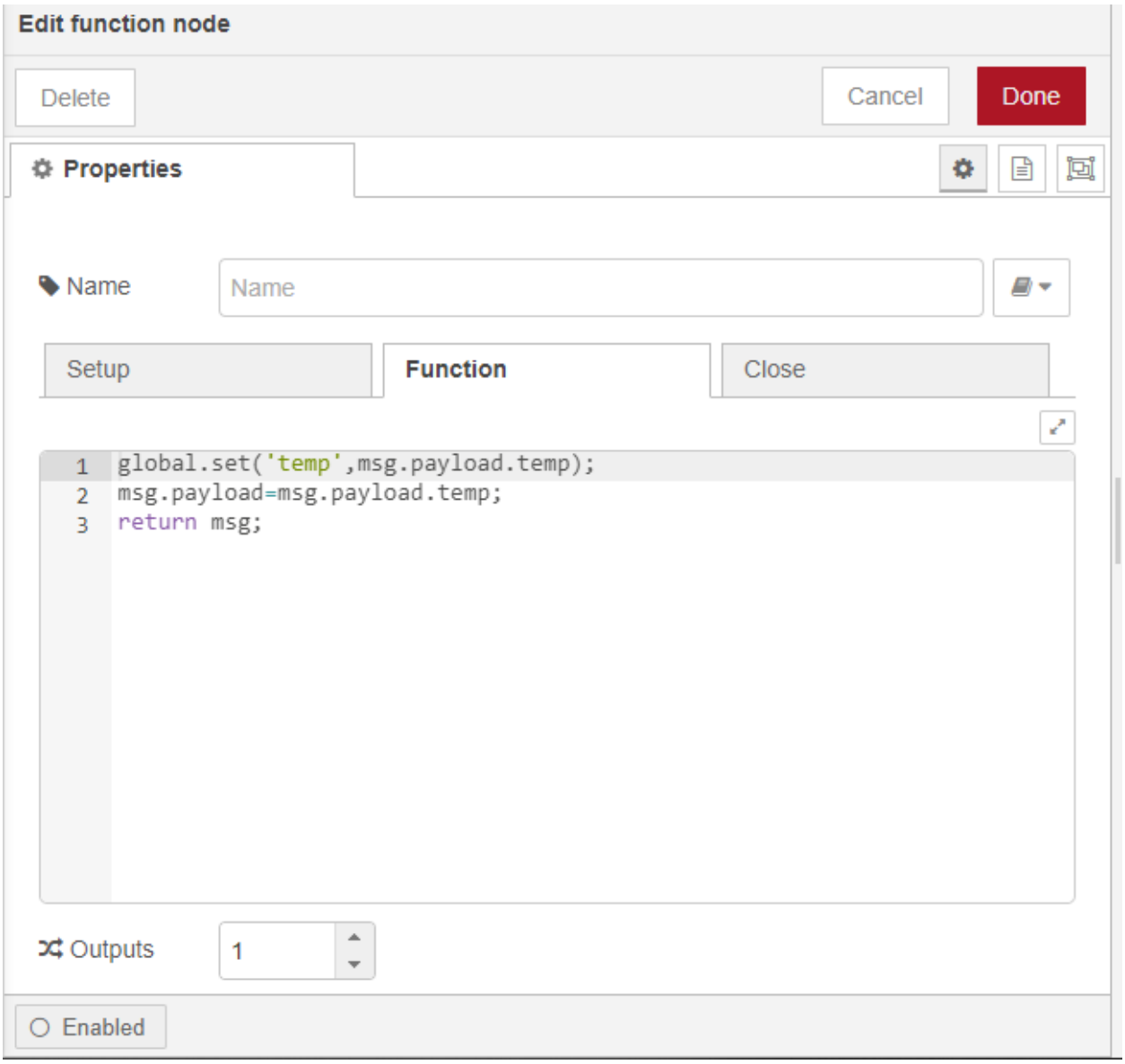


to connect ibmiot device node to the device double click on the node and enter the device credentials of the device that u have in your iot watson platform

like this

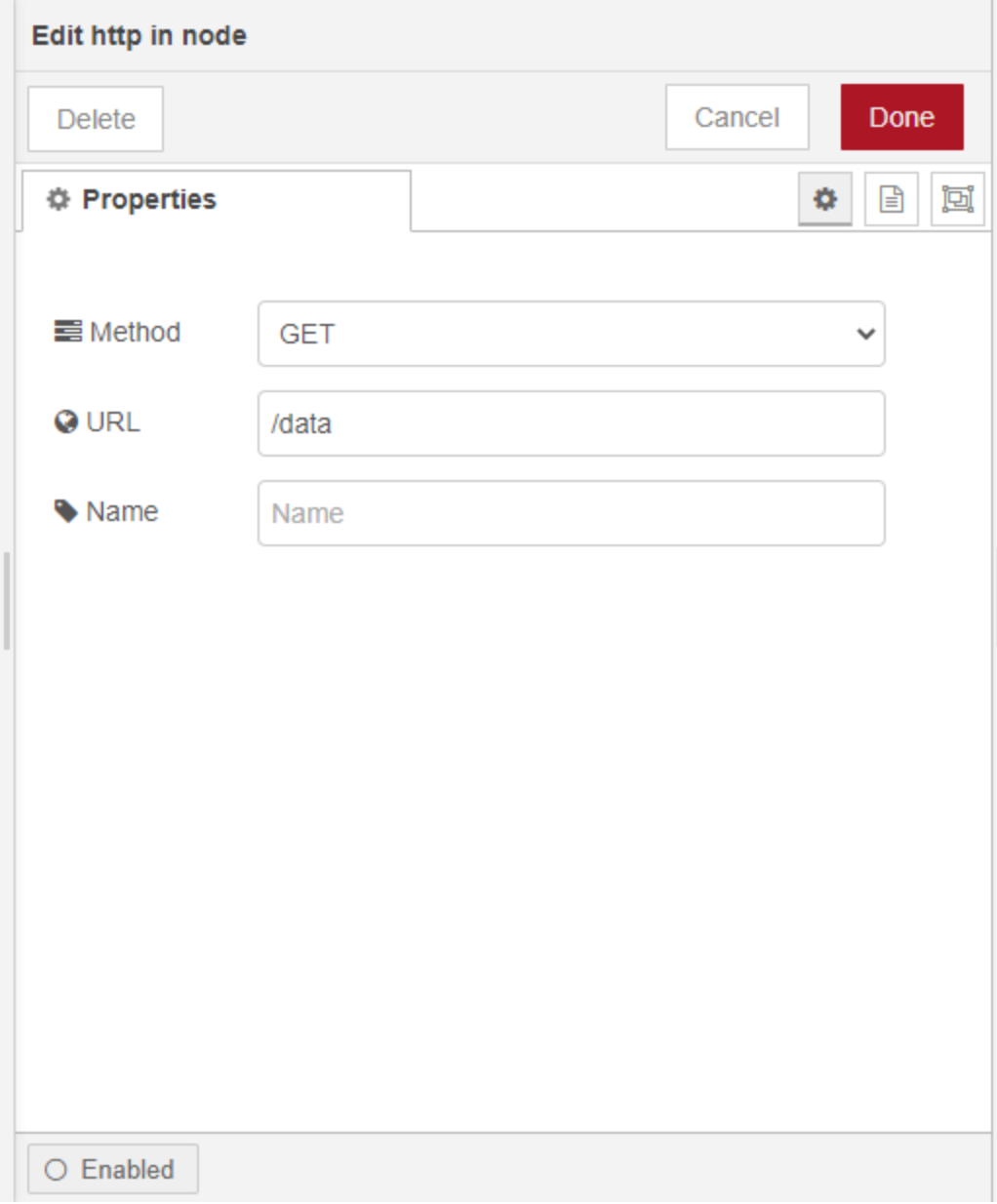


now the data that comes from the device is combined you need to parse the data and display data individually
code the function node like this

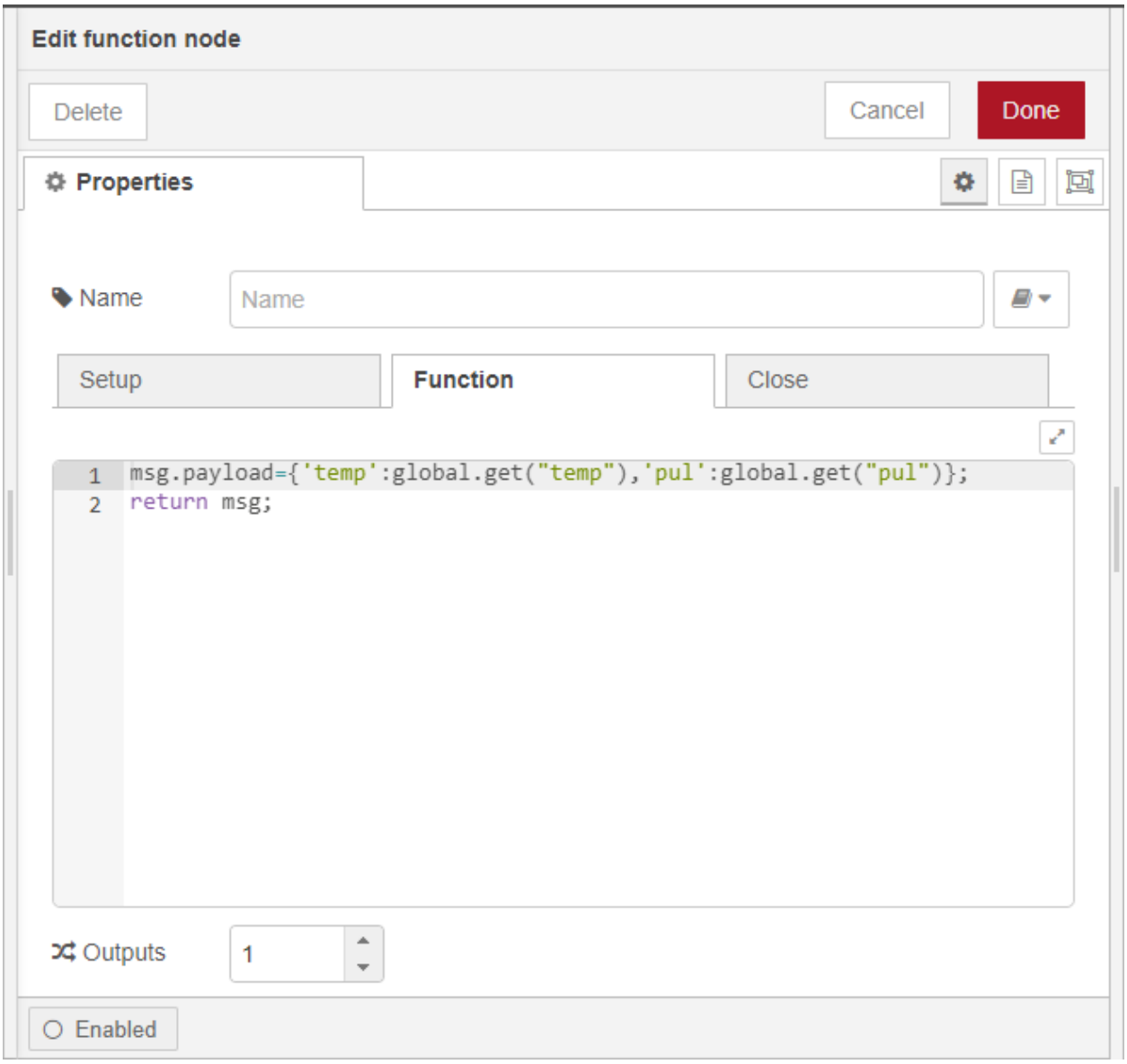


FLOW 2:

to create http request to communicate with mobile app

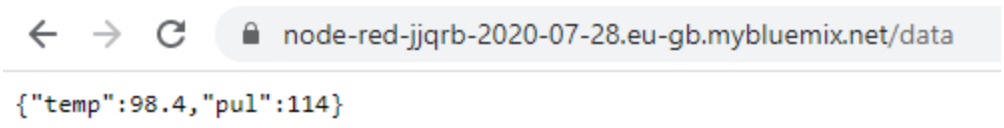


configure httpin node like this

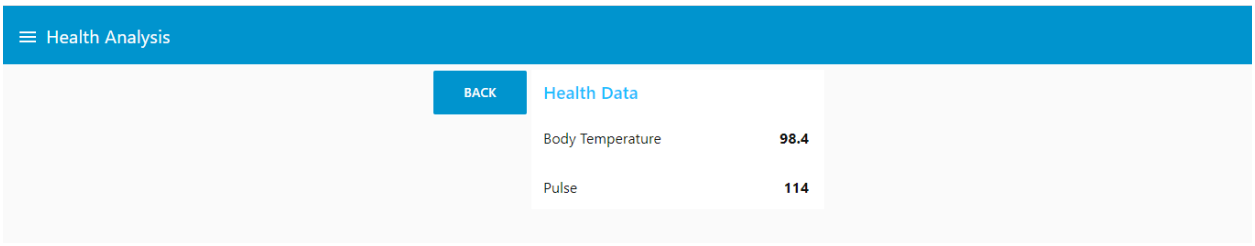


BY THIS FLOW WE ARE SENDING THE DATA TO SERVER

the data has been sent to the server looks like this



THE WEB APP UI WILL BE LIKE THIS



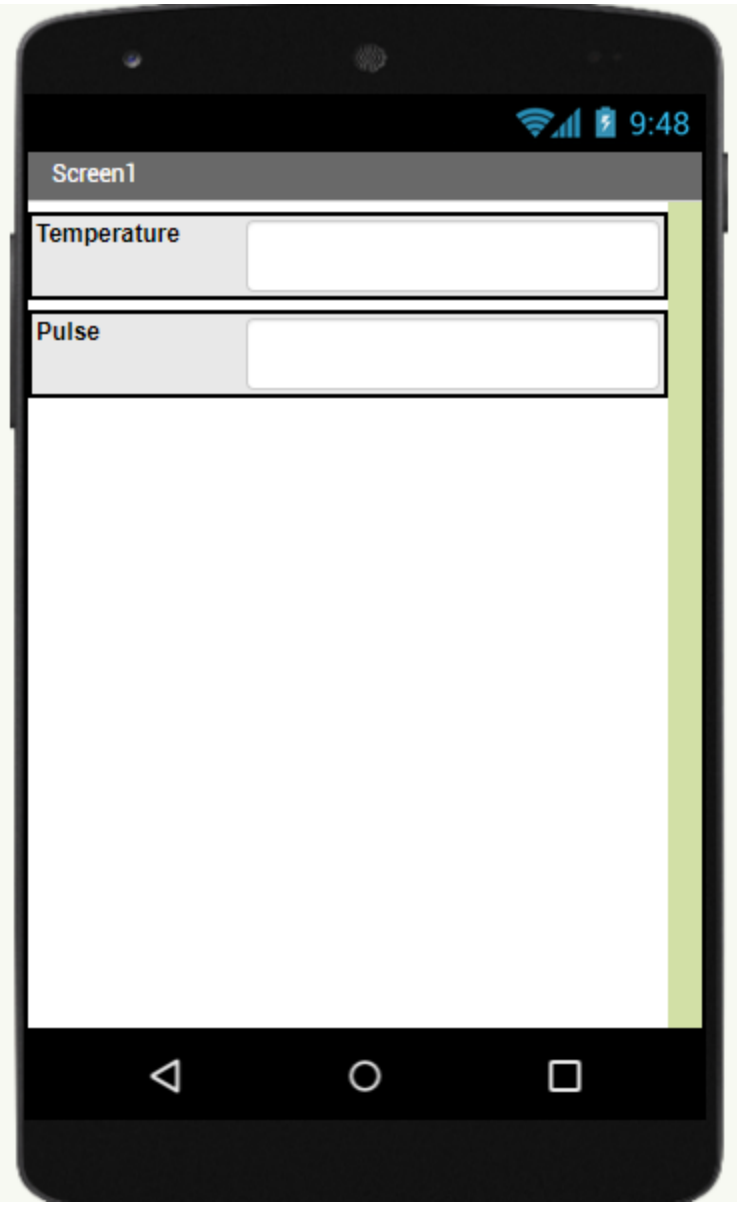
CREATE A MOBILE APP USING MIT APP INVENTOR & CONFIGURE IT TO GET DATA FROM CLOUD

search mit app inventor in browser and open it

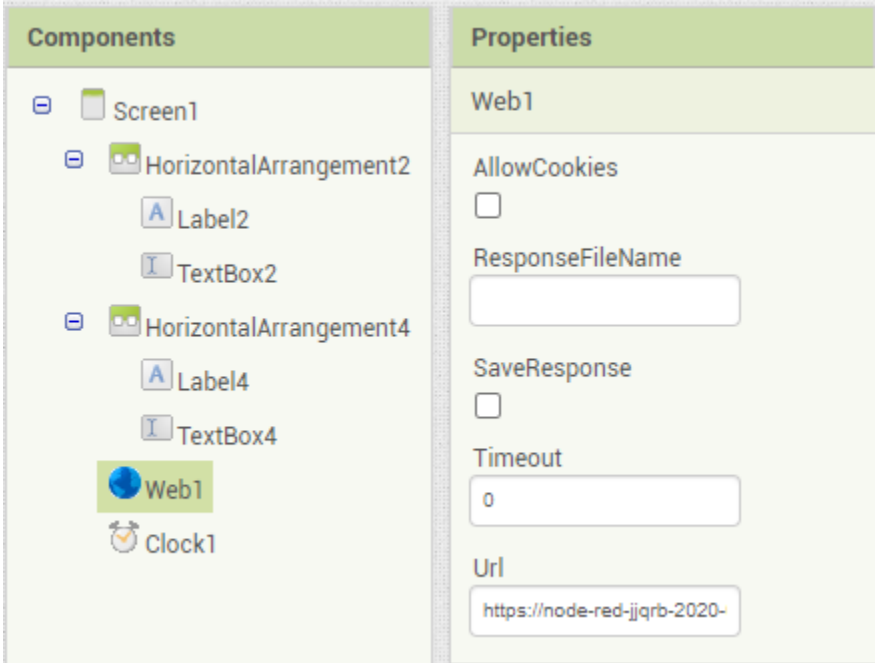
click on create apps on dashboard & login with ur google account

give the name of your project without spaces

configure the ui of your app like this

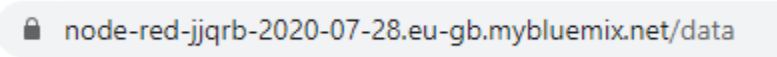


NOW DROP THE WEB FROM CONNECTIVITY ON TO THE BOARD & ENTER THE URL IN THE WEB

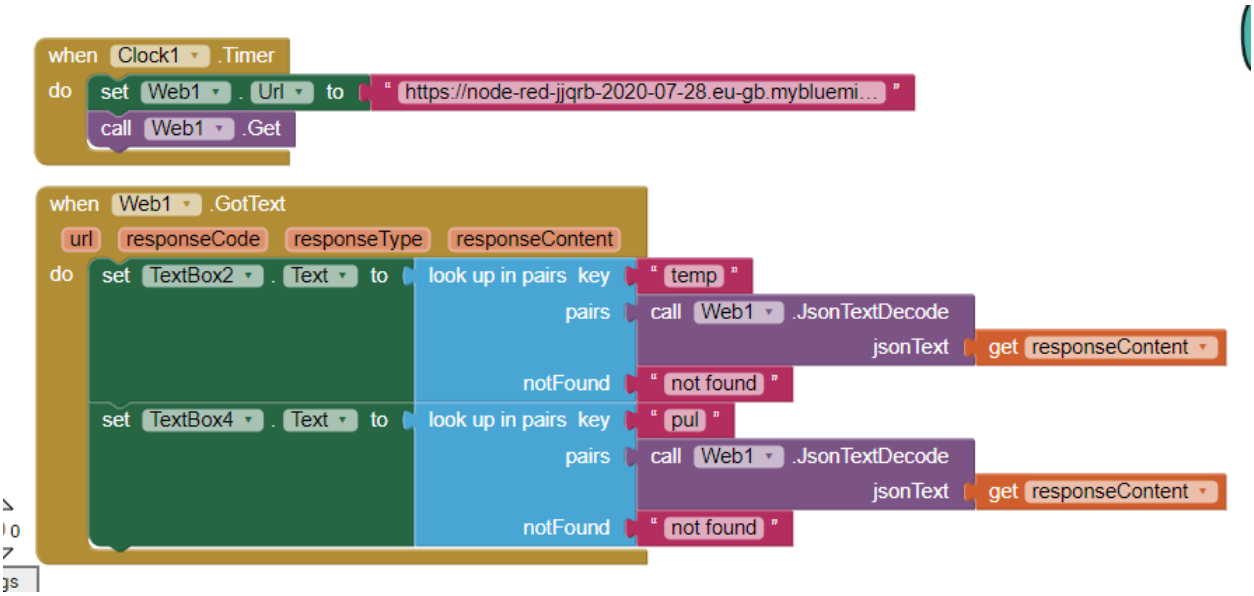


NOTE:
The app receives the data from the url that u enter In the web so you should enter the url that receives data from ibm device

enter this url



now click on the blocks on the top right corner of the screen and start arranging the blocks to create back end of the app
set the blocks in manner of text boxes



these blocks are there to decode data that is in the form of json & display then in their respective text boxes
now click on build option on the top of dashboard & download the apk file & install it in your mobile
app on mobile will be like this

PYTHON CODE:

```
#try to use jupyter notebook while executing the program wait for atleast 40 seconds for the entire program to run

import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
import requests
#Provide your IBM Watson Device Credentials
organization = "ytissp"
deviceType = "raspberrypi"
deviceId = "12345678"
authMethod = "token"
authToken = "12345678"


def myCommandCallback(cmd):
	print("Command received: %s" % cmd.data)#Commands


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() #try with different values
```

```
while True:
    temp=round(random.uniform(96,102),1)   # print temperature
    pul=random.randint(10,180)   # print pulse
    #enter your mobile number
    if (temp <= 97) or (temp >=100):

r=requests.get('https://www.fast2sms.com/dev/bulk?authorization=sDNoOkViHbSXQA
xwBU0q753IPKdJ6L4mYatgCFZy8fElnWGR1hPZK2UbvViosYwWEA0B9haIOLGjx4X3&
sender_id=FSTSMS&message=ABNORMAL TEMPERATURE
DETECTED&language=english&route=p&numbers=7893836463')
    if (pul <= 50) or (pul >= 110):

r=requests.get('https://www.fast2sms.com/dev/bulk?authorization=sDNoOkViHbSXQA
xwBU0q753IPKdJ6L4mYatgCFZy8fElnWGR1hPZK2UbvViosYwWEA0B9haIOLGjx4X3&
sender_id=FSTSMS&message=ABNORMAL PULSE
DETECTED&language=english&route=p&numbers=7893836463')
    data = { 'temp': temp , 'pul':pul}
    #print (data)
    def myOnPublishCallback():
      print ("Published  temp = %s " % temp,"pul = %s" % pul,"to IBM Watson")

    success = deviceCli.publishEvent("kitchen", "json", data, qos=0,
on_publish=myOnPublishCallback)
    if not success:
      print("Not connected to IoTF")
    time.sleep(10)

    deviceCli.commandCallback = myCommandCallback


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

OUTPUT:

```
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
====== RESTART: C:\Users\monica\Desktop\smartHealth\smartHealthAnalysis.py =====
2020-07-29 16:14:12,360   ibmiotf.device.Client      INFO    Connected successfully: d:ytissp:raspberrypi:12345678
Published  temp = 99.9  pul = 25 to IBM Watson
Published  temp = 98.8  pul = 91 to IBM Watson
Published  temp = 99.4  pul = 109 to IBM Watson
```

Conclusion:-

IoT Analytics in Health Monitoring for Live Monitoring of Temperature, BP, Pulse has been proposed using Node Red and MIT App. IoT Analytics in Health Monitoring being proposed via this report will assist Humans and take efficient care of Health as the System will always provide helping hand to Humans for getting Health data in MIT App

Future Scope:-

The Future Scope of Health Monitoring Device IoT got multiple benefits.The technology thus holds a strengthening **future** providing independent and mobile **health monitoring** while reducing the stress to visit doctors and **health** personal.

THE END