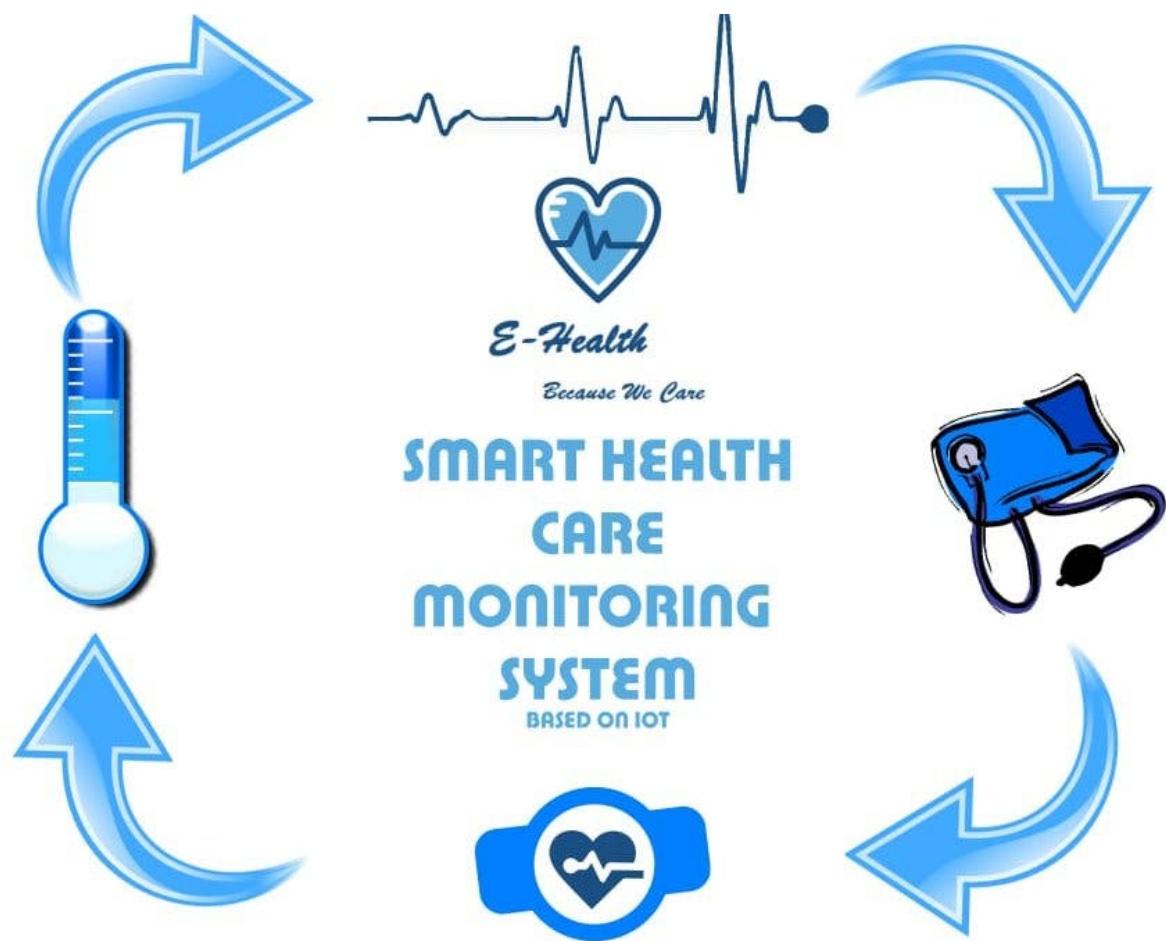A Project Report

On

IOT Analytics In Health Monitoring

By:

Sandiri Sowmya

Banda Sai Shivani

Vanamala Vaishnavi

Vamshi Krishna Siddaveni

as an interns at

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:-
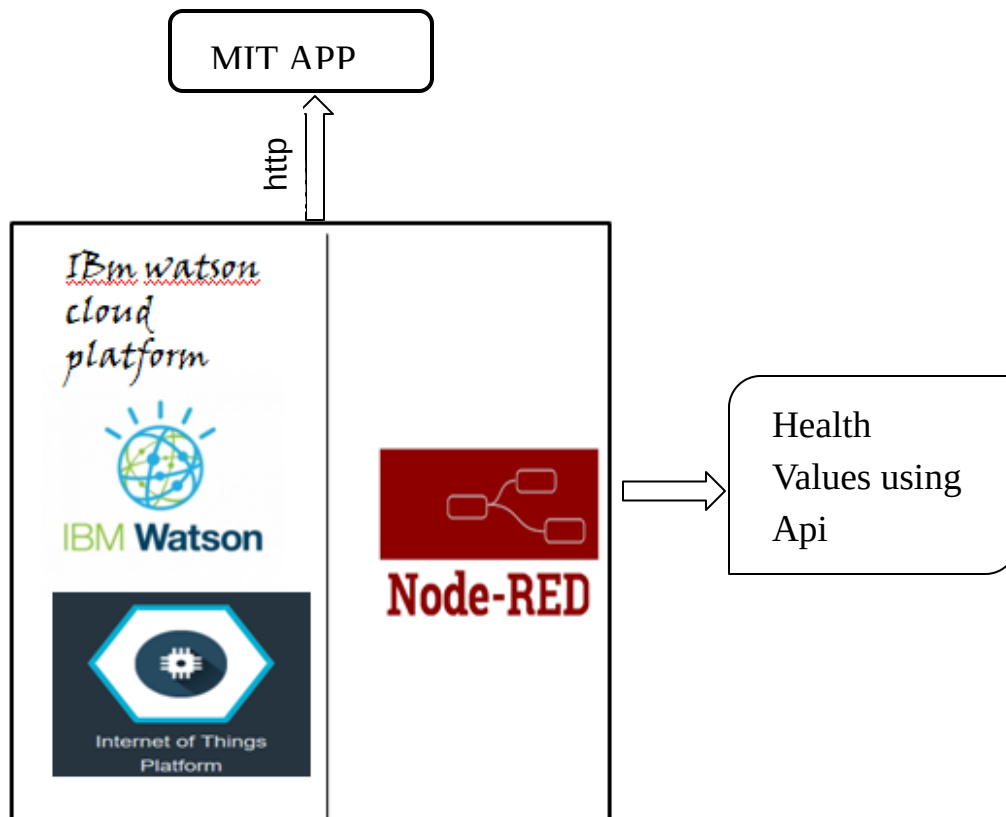
Designing Procedure:-

(1) Sign-in to your IBM cloud account from the link https://cloud.ibm.com/login. 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 in below:
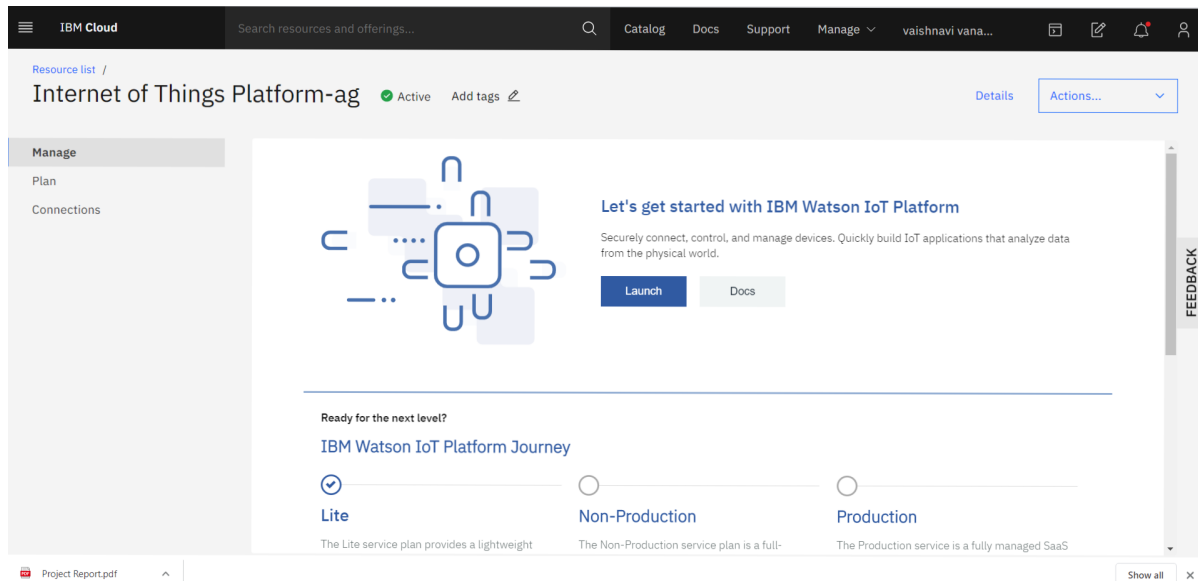
(2) Now,create IBM Watson IOT Platform and also create device.
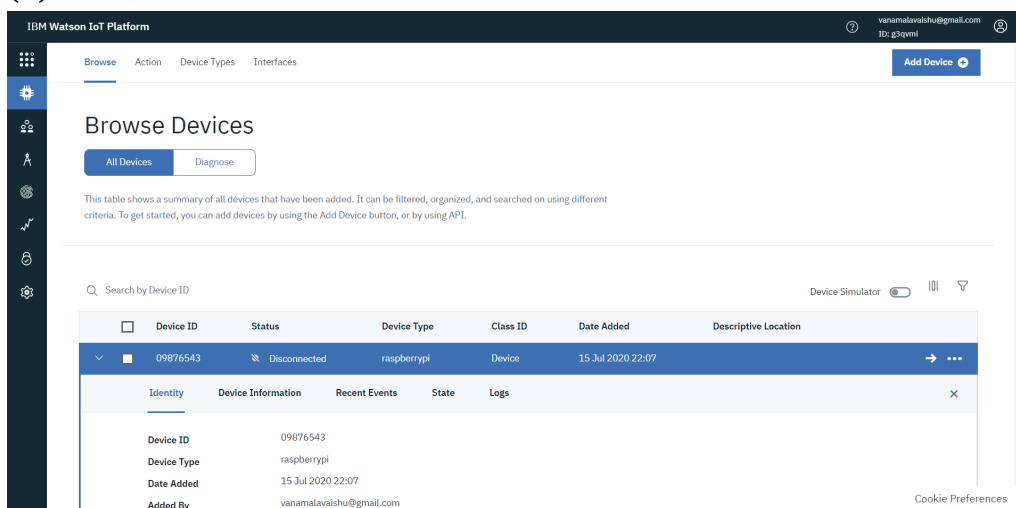
(3) Now, create Node-Red as shown here.And click on Visit App URL.Then you can see Main Page of Node-Red .

Search resources and offerings...

Catalog   Docs   Support   Manage ⌄   vaishnavi vana...

Resource list /

Node RED VBFMF   ● Running   Visit App URL   Add tags ✎

Details   Actions... ⌄

Getting started

Overview

Runtime

Connections

Logs

API Management

Autoscaling

Availability Monitoring

Instances   Edit ✎

Health

100%

1/1 instance(s) are running

Instances

1 ⇕

MB memory per instance

0 ————————————————●— 256   256

Runtime

SDK for Node.js™

256
Total MB allocation

0 MB still available
■ Used ■ Free

Runtime cost

Current and estimated cost excludes connected services.

$0.00 | $0.00

Connections (1)

🔵 node-red-vbfmf-cloudant-1594802753802-73060

📄 Project Report.pdf   ∧

Show all   ✕

Node-RED on IBM Cloud

Node-RED

Flow-based programming for the Internet of Things

Node-RED is a programming tool for wiring together hardware
devices, APIs and online services in new and interesting ways.

This instance is running as an IBM Cloud application, giving it
access to the wide range of services available on the platform.

More information about Node-RED, including documentation, can be
found at nodered.org.

Go to your Node-RED flow editor

Learn how to customise Node-RED

📄 Project Report.pdf   ∧

Show all   ✕

(4) Now, create MIT App using your MailID as shown below.

Flow:-

To obtain the health details from the API. Then, go to the API Keys tab and generate a key and make a note of the key. Now, in the API tab, select the API Docs and make a note of the API call accordingly.
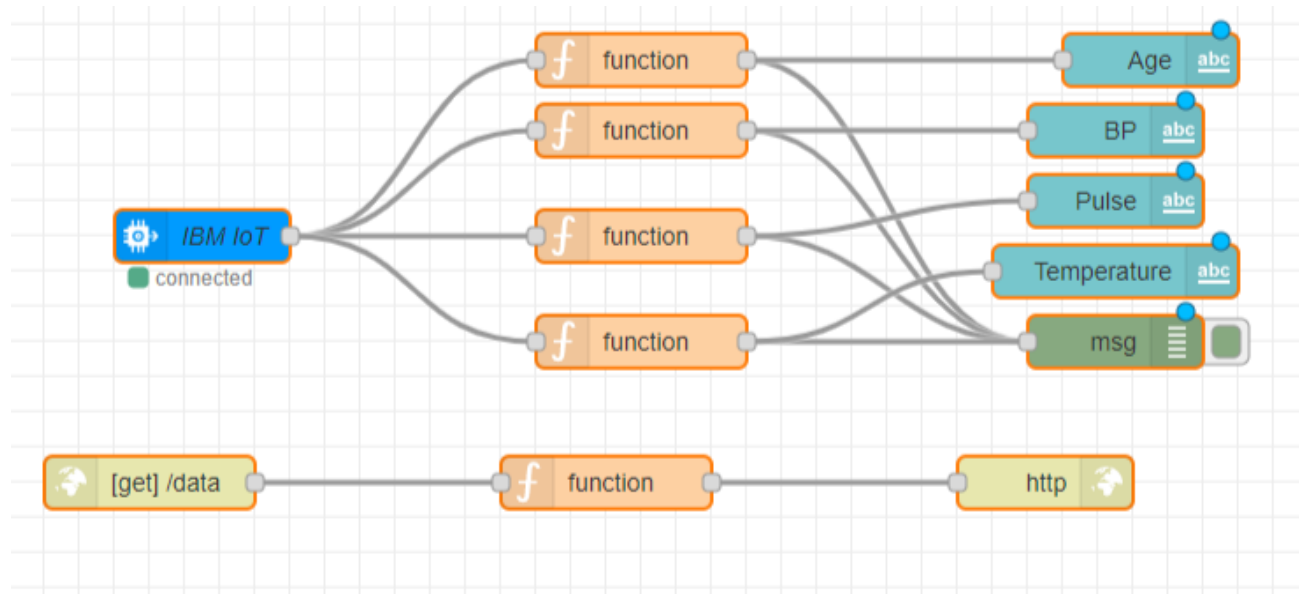
Edit function node

Delete   Cancel   Done

⚙ Properties   ⚙ 📄 🔲

🏷 Name   [Name   ]   📖▾

| Setup | **Function** | Close |

```
1  global.set('Age' , msg.payload.age);
2  msg.payload=msg.payload.age;
3  return msg;
```

⤢ Outputs   [1] ▲▼

Edit function node

Delete   Cancel   Done

⚙ **Properties**   ⚙ 📄 ▣

🏷 Name   [Name   ]   📖▾

| Setup | **Function** | Close |

```
1  global.set('BP' , msg.payload.systolic +'/' + msg.payload.diastolic);
2  msg.payload=msg.payload.systolic +'/' + msg.payload.diastolic;
3  return msg;
```

⤨ Outputs   [1   ] ▲▼

○ Enabled

Edit function node

Delete   Cancel   Done

⚙ **Properties**   ⚙ 📄 🔲

🏷 Name | Name | 📖 ▾

| Setup | **Function** | Close |

```
1  global.set('Pulse' , msg.payload.pulse);
2  msg.payload=msg.payload.pulse;
3  return msg;
```

⤬ Outputs | 1 | ▲▼ |

Edit function node

Delete   Cancel   Done

⚙ Properties   ⚙ 📄 🔲

🏷 Name   [Name]   📓▾

Setup	Function	Close

↗

```
1  global.set('temperature' , msg.payload.temperature);
2  msg.payload=msg.payload.temperature;
3  return msg;
```

Edit function node > **JavaScript editor**

Cancel   Done

```
1  msg.payload={'Age':global.get("Age"),'BP':global.get("BP"),'Pulse':global.get("Pulse"),'temperature':global.get("temperatur
2  return msg;
```
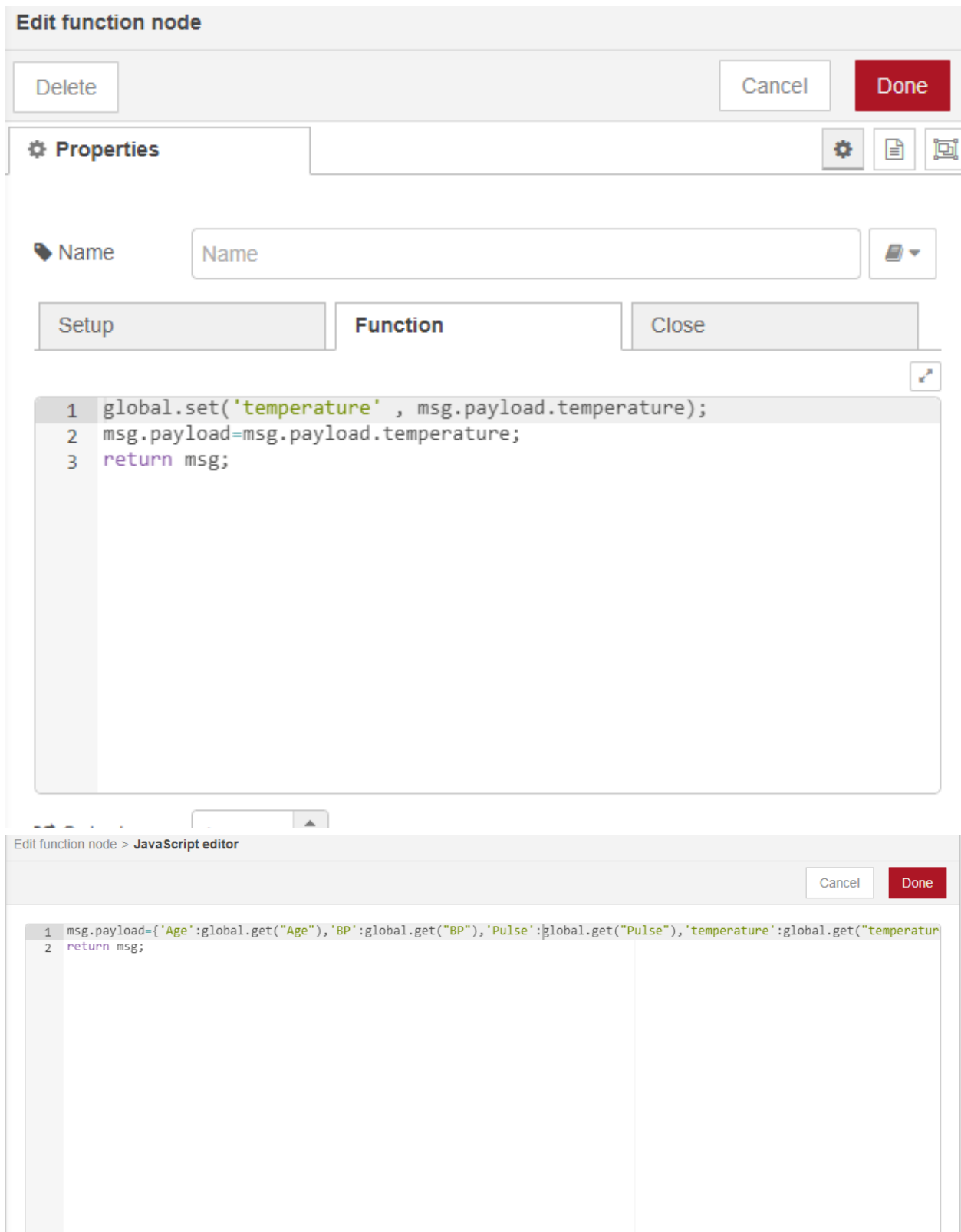
RESULT Of Node Red :-

Home		
Age		57
BP		100/99
Pulse		53
Temperature		37

node-red-vbfmf.eu-gb.mybluemix.net/data

{"Age":57,"BP":"100/99","Pulse":53,"temperature":37}

MITApp:
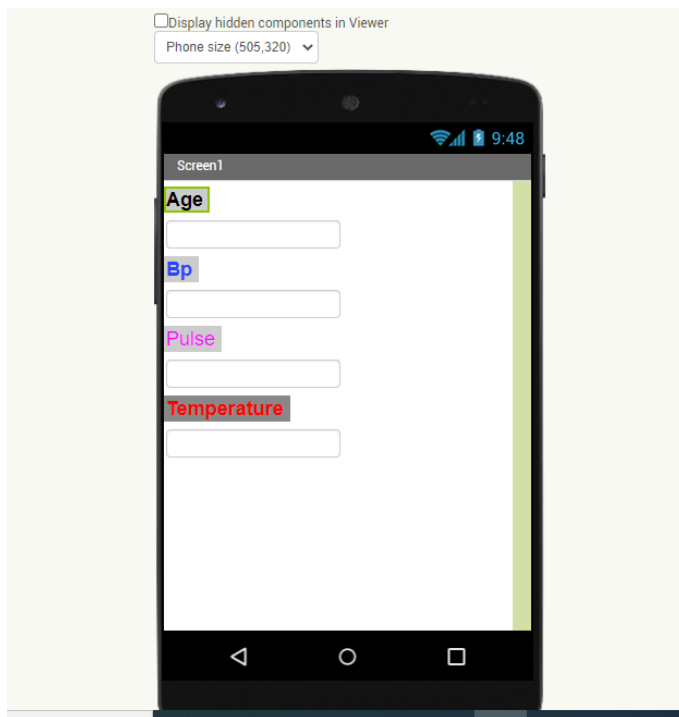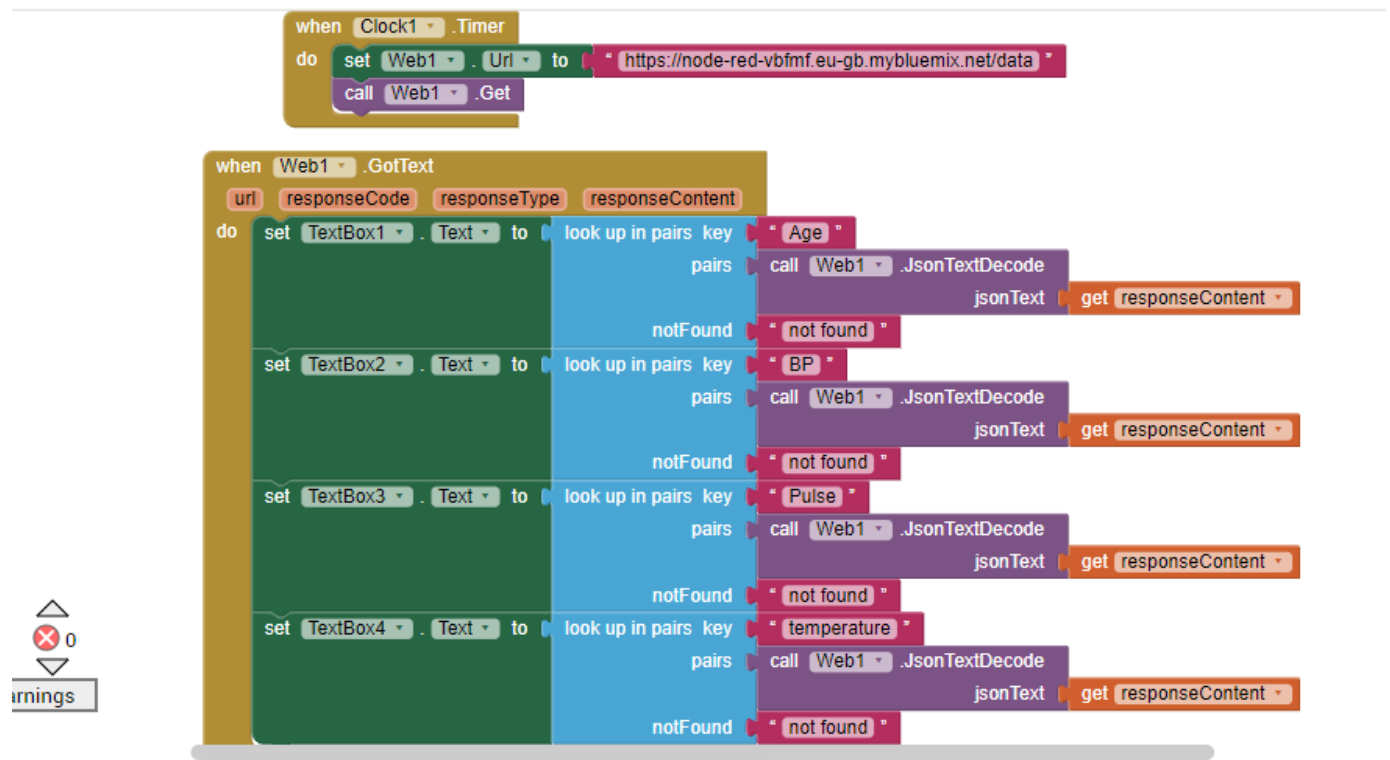
Result Of MIT App:-

FLOW CHART DESCRIBING THE WORKING OF IOT ANALYTICS IN HEALTH MONITORING:-

```
┌─────────────────┐
│     START       │
└─────────────────┘
         │
         ▼
┌─────────────────────┐
│ connected to network│
└─────────────────────┘
         │
         ▼
┌──────────────────────────────┐
│ Get details from the health API│◄────────┐
└──────────────────────────────┘          │
         │                                 │
         ▼                                 │
┌──────────────────────┐                   │
│  Send data to Web APP │                   │
└──────────────────────┘                   │
         │                                 │
         ▼                                 │
┌──────────────────┐                       │
│   Time Delay      │───────────────────────┘
└──────────────────┘
```
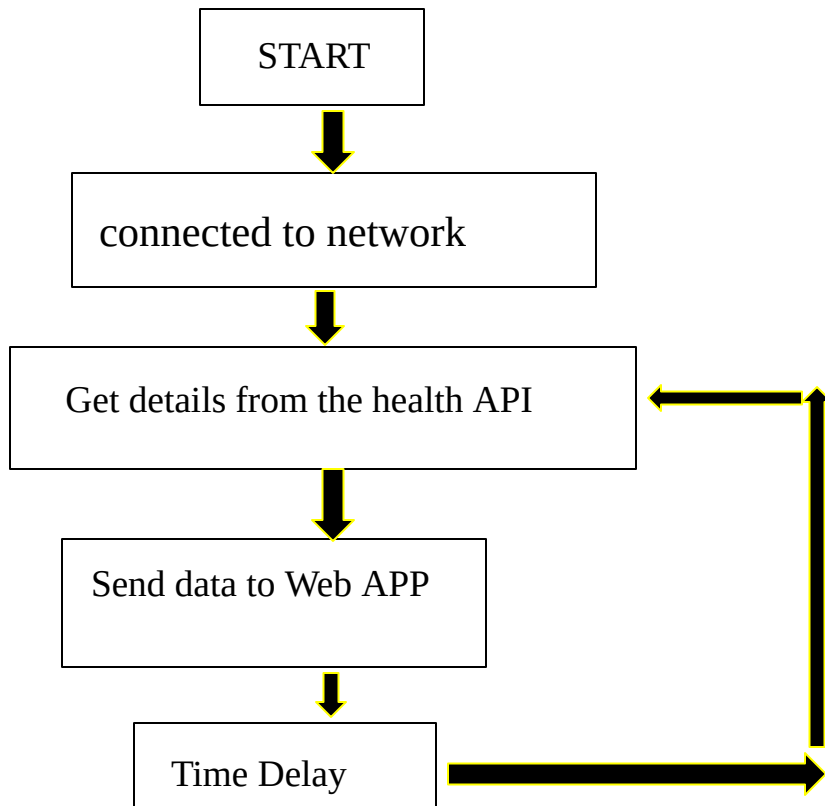
Python Code:

```python
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
#Provide your IBM Watson Device Credentials
organization = "g3qvmi"
deviceType = "raspberrypi"
deviceId = "09876543"
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()
```

```
while True:

    age=random.randint(18,78)
    #print(age)
    temp=random.randint(35,40)
    #print(temp)
    sys=random.randint(100,150)
    #print(sys)
    dia=random.randint(60,100)
    #print(dia)
    pul=random.randint(10,180)
    #print(pul)
    #Send age,temperature,systolic,diastolic,pulse to IBM Watson
    data = {'age':age,'temperature':temp,'systolic':sys,'diastolic':dia,'pulse':pul}
    #print (data)
    def myOnPublishCallback():
        print ("Published age= %s %%" %age,"temperature = %s %%" % temp,"systolic= %s
%%"%sys,"diastolic=%s %%"%dia,"pulse=%s %%"%pul, "to IBM Watson")

    success = deviceCli.publishEvent("Health", "json", data, qos=0,
on_publish=myOnPublishCallback)
    if not success:
        print("Not connected to IoTF")
    time.sleep(2)

    deviceCli.commandCallback = myCommandCallback

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

Output:

File   Edit   Shell   Debug   Options   Window   Help

```
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:20:19) [MSC v.1925 32 bit (In
tel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==================== RESTART: C:\Users\DELL\Desktop\health.py ====================
2020-07-17 19:48:58,522   ibmiotf.device.Client      INFO     Connected successfu
lly: d:g3qvmi:raspberrypi:09876543
Published age= 76 % temperature = 35 % systolic= 100 % diastolic=91 % pulse=132
% to IBM Watson
Published age= 54 % temperature = 40 % systolic= 129 % diastolic=62 % pulse=58 %
 to IBM Watson
Published age= 77 % temperature = 38 % systolic= 118 % diastolic=65 % pulse=140
% to IBM Watson
Published age= 49 % temperature = 36 % systolic= 130 % diastolic=66 % pulse=124
% 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