### **IoT Analytics In Health Monitoring System**

1. **Introduction:**
2. **Overview**:

This project is a health monitoring system used to detect user’s health attributes and it helps users to be aware of their health condition.It is also used to detect any health issues.

B.**Purpose:**

This project aims to build a mobile app which will visualize the health parameters and show the user's health level according to the predicted output.

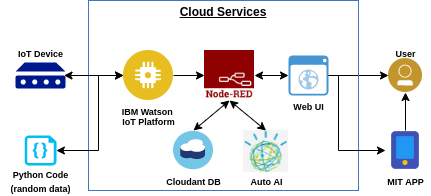
**II. Survey:**

1. **Existing Problem:** In the present day early detection of diseases can play a vital role in people's lives and may save them.This Health Monitoring System can reduce medical costs for the user and help them identify hidden diseases if they have any and also can store their vitals for future references.It is a quick and easy way to monitor our health.

1. **Proposed Solution**: The IoT device will sense the temperature, pulse and BP values of the person and upload it to IBM IoT platform(Python code can also be used). In the cloud the data will be sent to an AutoAI to predict users health status and alert the persons if their health condition is abnormal.Then a dashboard which will visualize the health parameters and send alerts or updates to the users phone via an app or text.

**III.Theoretical Analysis:**

1. BLOCK DIAGRAM:



1. Hardware/Software Designing:

1.Python: It is a general purpose programming language used for software development besides web development.

2.IBM Watson IoT Platform: A fully managed, cloud-hosted service with capabilities for device registration, connectivity, control and data storage.

3. Node-Red App: A flow-based development tool for visual programming developed by IBM for wiring together hardware devices, APIs and online services as part of IoT.

4.Cloudant DB: : An IBM service that enables you to create, train, and deploy self-learning models using an automated, collaborative workflow.

5.MIT App:It is an online platform designed to make computational thinking concepts through development of mobile applications.

6. FAST2SMS: It is an online application used to send SMS to users.

**Result:**

A simulation was made on IBM Watson IoT Platform. There were 5 generated values (Age, Name, Heart Rate, Blood Pressure and Temperature) . There are many values generated. We can see in the output on Node Red the predictions which in turn are displayed in the MIT App and any abnormalities detected are sent to the user by FAST2SMS.

**Advantages:**

1. Quick detection.

2. Fast ML model.

3. Easy implementation.

**Disadvantages:**

1.IoT simulator is different from a real IoT device.

**Application:**This solution is adaptable to the field of healthcare services and can be used by anyone having App store/Google Play.

**Conclusion:** This project is a solution to predicting the vital health levels of a user by using IBM Watson Machine Learning Service, IBM Watson IoT Platform, Node-RED,Python, AutoAI, MIT App developer and Fast2SMS.

**Future Scope:**It's a tool to anticipate an alarming health situation and can be applied to many different diseases. It is also crucial for health problems which can only be detected through precise measurements of the temperature,blood pressure, heart rate etc and could not be seen directly.

**Done by:**

BODA NIHARIKA

BOGADHI SHARMILA

AJAY

G SHREE CHANDANA

KALLA PRANEETH KUMAR

**Bibliography:**

Python IDLE

<https://cloud.ibm.com/>

[http://ai2.appinventor.mit.edu/#](http://ai2.appinventor.mit.edu/)

https://www.fast2sms.com/dashboard/sms/bulk

**Appendix**:

Python Code:

import wiotp.sdk.device

import time

import random

x=('male','female')

y=('southwest','northwest','northeast','southeast')

myConfig = {

"identity": {

"orgId": "b34ic0",

"typeId": "PROJECT",

"deviceId":"123456789"

},

"auth": {

"token": "987654321"

}

}

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:

age=random.randint(0,100)

gender=random.choice(x)

region=random.choice(y)

temperature=random.randint(0,100)

systolic=random.randint(0,100)

diastolic=random.randint(0,100)

pulserate=random.randint(0,100)

myData={'age':age,'gender':gender,'region':region,'temperature':temperature,'systolic':systolic,'diastolic':diastolic,'pulserate':pulserate}

client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)

print("Published data Successfully: %s", myData)

client.commandCallback = myCommandCallback

time.sleep(10)

client.disconnect()

UI Screenshot:

