

Project Title: Remote Health Monitoring System with Analytics Dashboard

1. INTRODUCTION

1.1 Overview: Health monitoring systems play a vital role which will help in early detection of the diseases which can reduce the suffering and medical costs. In this Health monitoring system we will be detecting the level of ill health of the person and would recommend few medications that can be taken by him by which he can recover a bit than before using Machine learning and Internet of things.

1.2 Purpose: Health monitoring systems will help in early detection of the diseases which can reduce the suffering and medical costs. In this Health monitoring system we will be detecting the level of ill health of the person and would recommend few medications that can be taken by him by which he can recover a bit than before using Machine learning and Internet of things.

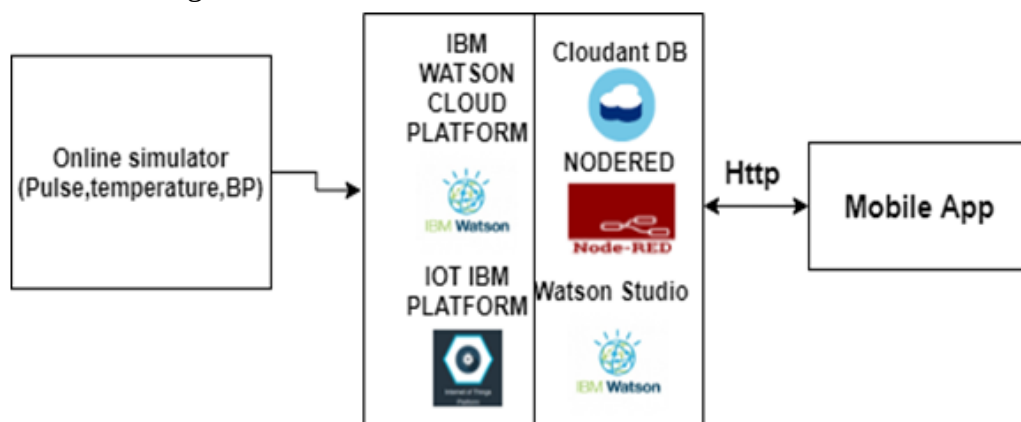
2. LITERATURE SURVEY

2.1 Existing problem: A majority of older adults are challenged by chronic and acute illnesses and/or injuries. Remote Health monitoring systems have been shown to be effective in helping to manage chronic disease, post-acute care, and monitoring the safety of the older adult population.

2.2 Proposed solution: Our aim in this project will be to gather patient data using various sensors and send it to cloud using a mobile application for storage and analysis. At the cloud end, various machine learning techniques will analyze the data and in case of any emergency, it will automatically inform the respective agencies/services. Thus, it will help in providing a quick and reliable health system.

3. THEORITICAL ANALYSIS

3.1 Block diagram



3.2 Project Flow:

- Send the health parameters (Temperature, BP and Pulse) to IBM IoT platform using an online simulator.
- Create a machine learning model using Watson studio and Auto AI experiment which will analyze the health status like “critical”, “fever” etc. and according to that status he/she needs to consult a doctor.
- Create a Node-RED flow to get data from IBM IoT devices and which will communicate with mobile apps using HTTP requests.
- Create a mobile app through which the person can register along with their name, age, and gender.
- Store the entire data in the Cloudant DB using Node-RED
- Configure the Node-RED to give the sensor input to the ML model and predict the illness of the person.
- Configure the mobile app which will visualize the health parameters and show the suggestions according to the predicted output.

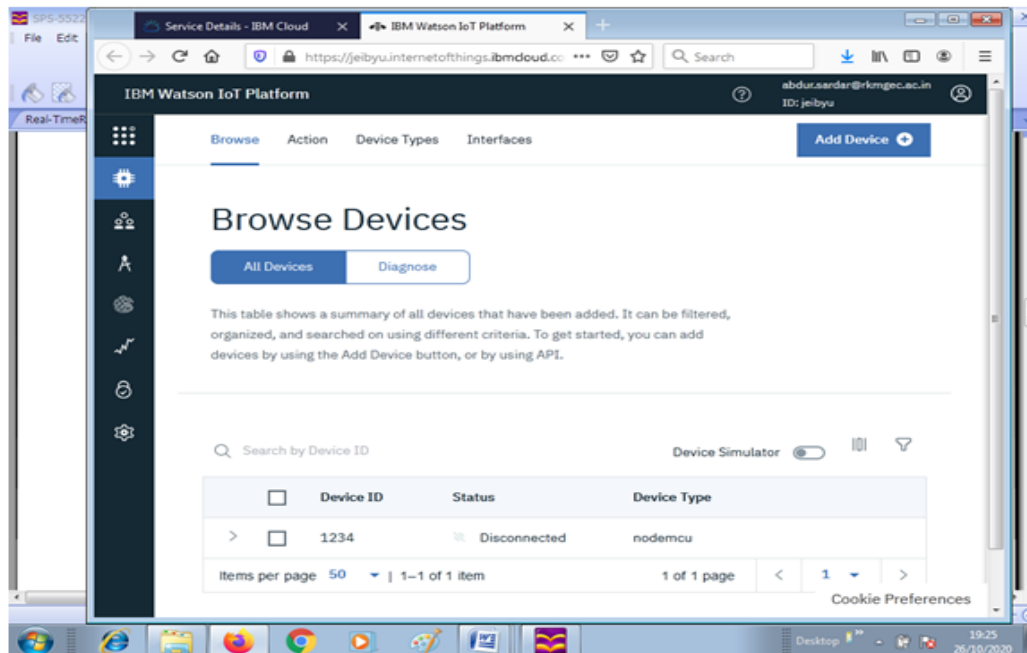
4. REQUIRED SERVICES:

- a) IBM IoT Cloud Platform
- b) IBM Watson Studio
- c) IBM AutoAI/Machine learning service
- d) Node-Red
- e) IBM Cloudant DB

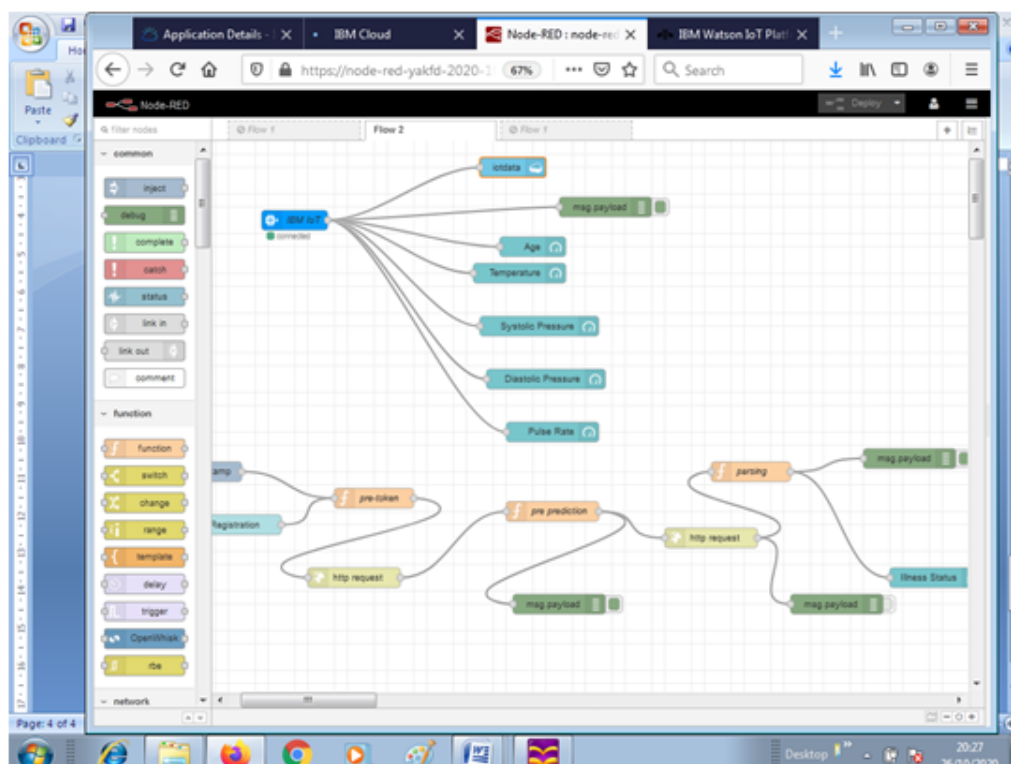
5. DESIGN DETAILS OF THE PROJECT:

The details workflow of the project is given below:

1. First we have created a device with IBM IoT platform which can simulate all the sensor data like Temperature, Systolic, Diastolic and Pulse.



2. To visualize the sensor data we have created flow in the Node-RED.



3. Next we have created a database using IBM cloudant and attached it with IBM IoT devices in the Node-RED flow diagram and all the simulated data is stored in that database.

Application Details

Service Details

Cloudant Dashboard

Node-RED : node

IBM Watson IoT

Home

Clipboard

Log Out

← → ↺ 🏠

https://e2df1b66-3e12-4171-8a47-9d9f0... 🔒

🔍 Search

📄 📁 📌

☰

🔍

🏠

📄

📁

📌

🔍

🏠

📄

📁

📌

🔍

🏠

📄

📁

📌

Document ID

🔍

{ }

📄

🔍

	Age	Diastolic	Pulse	Systolic	Temperature
78	112	69	52	80	
31	105	81	82	98	
35	100	97	71	94	
33	122	91	69	118	
34	110	113	66	117	
97	106	53	64	114	
1	88	104	69	103	
11	127	86	60	91	
79	139	55	52	106	
3	84	78	74	104	
57	108	88	72	97	
15	137	85	78	87	
88	106	103	63	101	

Showing 5 of 7 columns. ☐ Show all columns.

← →

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

🌐

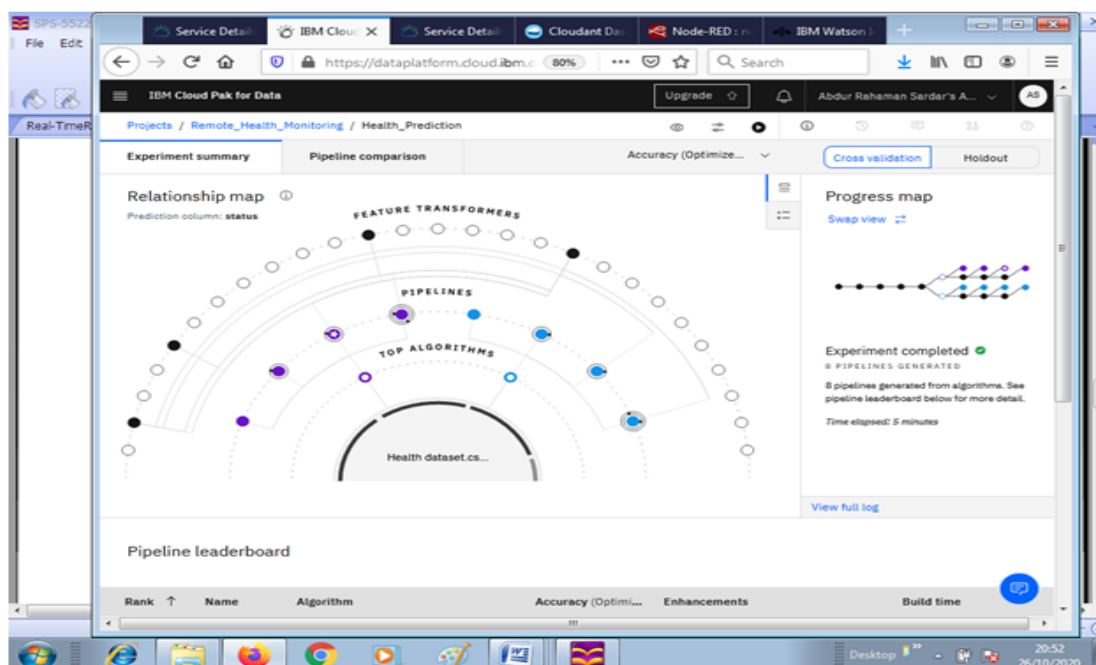
🌐

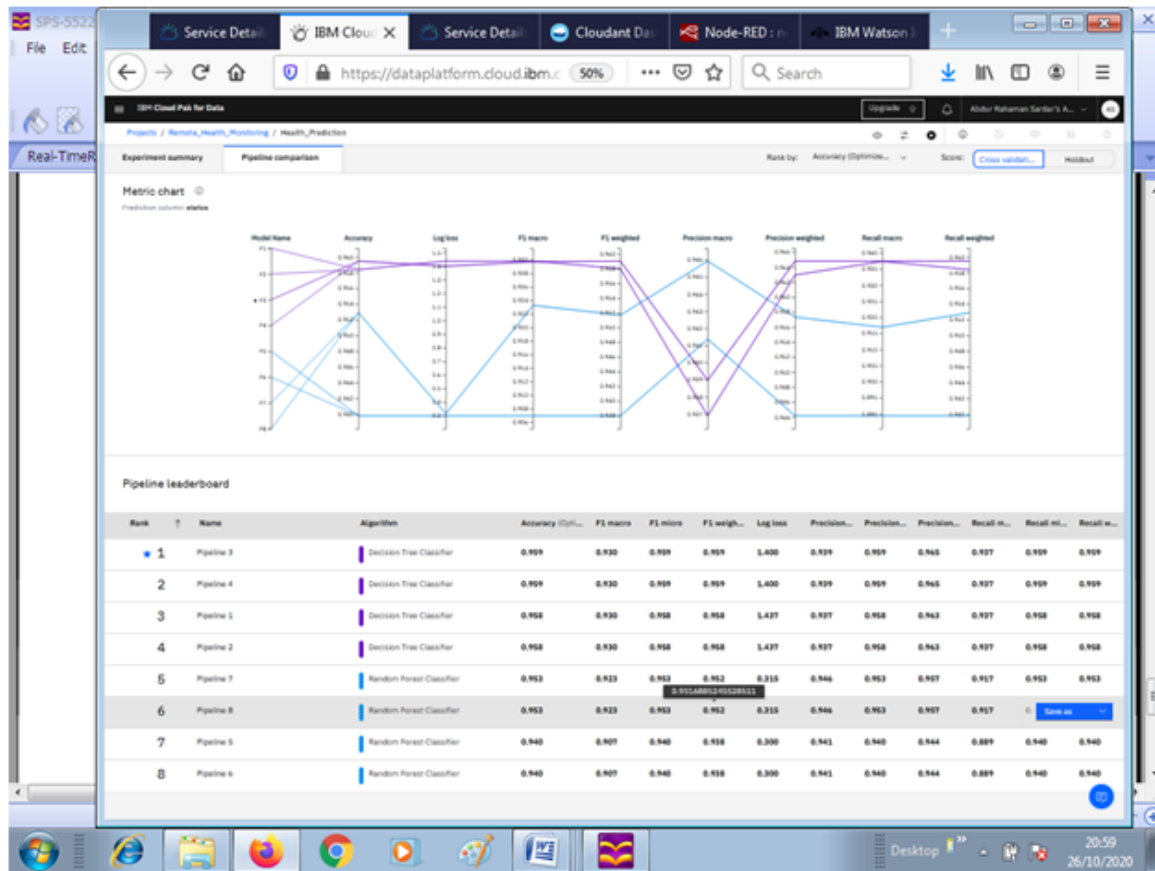
🌐

🌐

🌐

4. After that a machine learning instance is created an AutoAI using IBM Watson studio and trained the model using Auto AI.





The screenshot displays the IBM Cloud Pak for Data interface, specifically the "Remote_Health_Monitoring" API reference and test results. The "Enter input data" section contains fields for Age, Temperature, Systolic, Diastolic, and Pulse. The "Result" section shows the output of the API call, which is a JSON object containing predictions and values.

Enter input data:

- Age: 25
- Temperature: 37.5
- Systolic: 46
- Diastolic: 66
- Pulse: 100

Result:

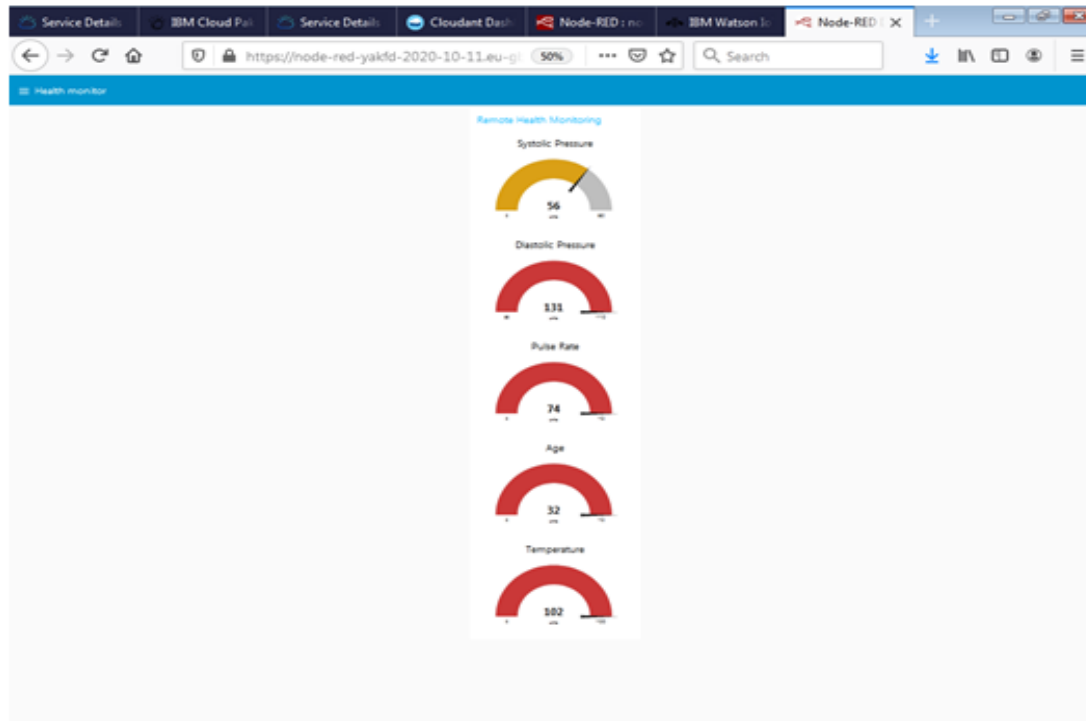
```

{
  "predictions": [
    {
      "fields": {
        "prediction": "Normal",
        "probability": 0.0
      },
      "values": {
        "Normal": 0.0
      }
    }
  ]
}

```


Output:

The first picture shows the simulated sensor data from web application and the second picture shows the present health status.



The screenshot displays a web application interface titled 'Home'. It features a form for user information and a section for illness status. The form includes fields for Name, Age, Gender, Temperature, Systolic, Diastolic, and Pulse, each with a corresponding 'SUBMIT' button. The illness status is currently 'Hypotension'.

Field	Value
Name	abcd
Age	25
Gender	male
Temperature	37
Systolic	66
Diastolic	46
Pulse	100

Illness Status: **Hypotension**

SUBMIT CANCEL