Project report

Team ID	PNT2022TMID47513
Project Name	Smart waste management system in metropolitan cities

1. Introduction

1. Project overview
2. Purpose

2. Literature Survey

1. Existing problem
2. Reference
3. Problem Statement Definition

3. Ideation & Proposed Solution

1. Empathy map canvas
2. Ideation & Brainstorming
3. Proposed Solution
4. problem Solution fit

4. Requirement Analysis

1. Functional requirements
2. Non-Functional requirements

5. Project Design

1. Data Flow diagram
2. Solution & Technical Architecture
3. User Stories

6. Project Planning & Scheduling

1. Sprint Planning & Estimation
2. Sprint Delivery Schedule

7. Coding & Solution

1. Code 1

2. Code 2

3. Code 3

4. Code 4

9. Results

1. Performance

10. Pros & Cons

11. Conclusion

12. Future Scope

13. Appendix

13.1 Source code

13.2 GitHub & Project Demo link

Project Report

Team ID	PNT2022TMId47513
Project Name	Smart Waste Management system in metropolitan cities.

1.INTRODUCTION

1.1 Project Objectives

Indiscriminate disposal of solid waste is a major issue in urban centers of most developing countries and it poses a serious threat to healthy living of the citizens. Access to reliable data on the state of solid waste at different locations within the city will help both the local authorities and the citizens to effectively manage the menace. In this paper, an intelligent solid waste monitoring system is developed using Internet of Things (IoT) and cloud computing (IBM) technologies. The fill level of solid waste in each of the containers, which are strategically situated across the communities, is detected using ultrasonic sensors. A Wireless Fidelity (Wi-Fi) communication link is used to transmit the sensor data to an IBM cloud platform. Depending on the fill level, the system sends appropriate notification message (in form of tweet) to alert relevant authorities and concerned citizen(s) for necessary action. Also, the fill level is monitored by Sensor in real-time. The system performance shows that the proposed solution may be found useful for efficient waste management in smart and connected communities.

1.2 Purpose

Smart waste management is about using technology and data to create a more efficient waste industry. Based on IoT (Internet of Things) technology, smart waste management aims to optimize resource allocation, reduce running costs, and increase the sustainability of waste services. A waste management system is the strategy an organization uses to dispose, reduce, reuse, and prevent waste. Possible waste disposal methods are recycling, composting, incineration, landfills, bioremediation, waste to energy, and waste minimization. This is make possible to plan more efficient routes for trash collectors who empty the bins, but also lowers the chance of any bin being fill for over a week. A good level of coordination exists between the garbage collectors and the information supplied via technology. The sensor sent the alert message so that they can collect the garbage on time without littering the surrounding area. The fill patterns of specific containers cam be identified by historical data & managed accordingly in the long term. Thus, smart waste management provides us with the most optimal way of managing the waste in an efficient manner using advanced Technology.

2. Literature Survey:

2.1 Existing problem:

Solid waste management issue is the biggest challenge to the authorities of both small and large cities' in developing countries. This is mainly due to the increasing generation of such solid waste and the burden posed on the municipal budget. In addition to the high costs, the solid waste management is associated lack of understanding over different factors that affect the entire handling system. An analysis of literature and reported related to waste management in developing countries, showed that few articles supplied quantitative information. The objective of the mentioned studies was to determine the stakeholders' action/behavior that have a role in the solid waste management and to analyze different factors that affect the system. The studies carried out in 4 continents, in 22 developing countries and on more than thirty urban areas. A combination of variable methods that were used in this study was mentioned in details in order to encourage the stakeholders and to assess the factors influencing the performance of the solid waste management in the studied cities.

2.2 References

Title **:** Smart Waste analysis

Author **:** M. Mohammad Aazam

Description :

It provides the idea of sensors based waste bins, capable of notifying waste level status. An automatic waste bin and make use of cloud computing paradigm to evolve a more robust and effective smart waste management mechanism. Waste management is linked to different stakeholders, including recyclers, importers and exporters, food industry, healthcare, research, environment protection and related organizations, and tourism industry Mohammad Aazam et al proposed Cloud SWAM, in which each bin is equipped with sensors to notify its waste level.

Title **:** Waste Management System Using IoT Based Machine Learning

Author **:** T. Anh Khoa, C.H. Phuc, P.D.Lam

Description :

In this work, an optimal algorithm combining graph theory and LR has been described, with the possibility of assessing the probability of a trash bin

being fully based on the number of classes in the university, is algorithm presents many advantages, as compared with the old waste collection methods.

Title : IoT-Based Smart Garbage System for Efficient Food Waste Management
Author : I. Hong,S. Park, B. Lee, J. Lee, D. Jeong
Description :

An IoT-based SGS for replacing existing RFID-based garbage collection systems. To provide differentiation from passive collection bins and other types of RFID-based food garbage collection systems, we also proposed components required in external and public environments and designed the SGS based on these components. The basic system structure of a SGB is a centralized structure in which information gathered in each bin is transferred to the server; we also designed a HSGB for improving the battery efficiency of each SGB.

Title : Smart Solid Waste Management

Author : Mohd Helmy Abd Wahab

Description :

At the time of trash disposal, the material to be recycled could be identified using RFID technology.

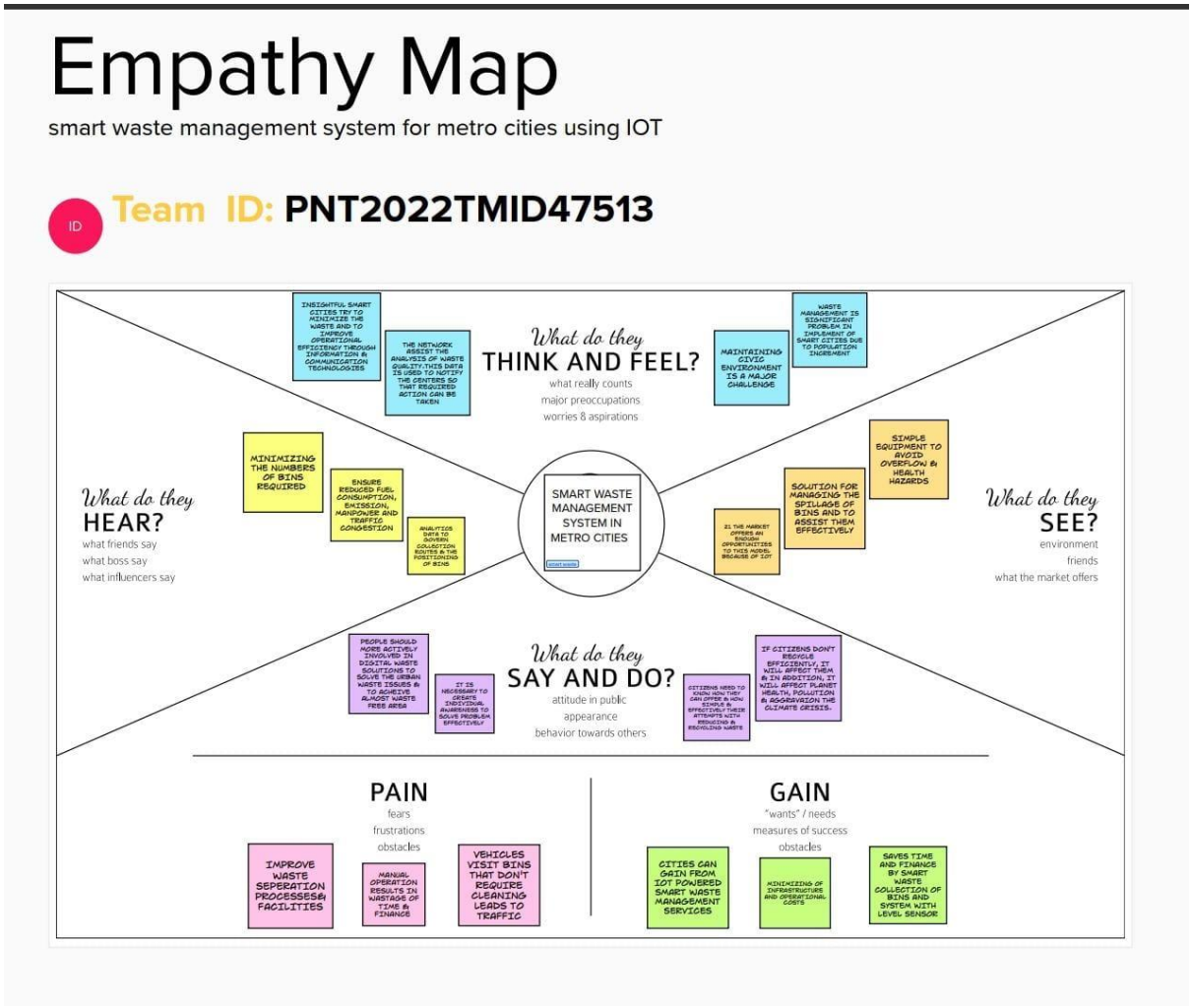
2.3 Problem Statement definition

Problem Statement	I'm (Customer)	I'm trying	But	Because	What makes me feel
PS 1	Municipality corporation authority	When the trash bins are filled & it made aware of where the full bins are located.	Don't have the facilities at the moment.	There is no tool available to determine the level of bins.	Frustrated

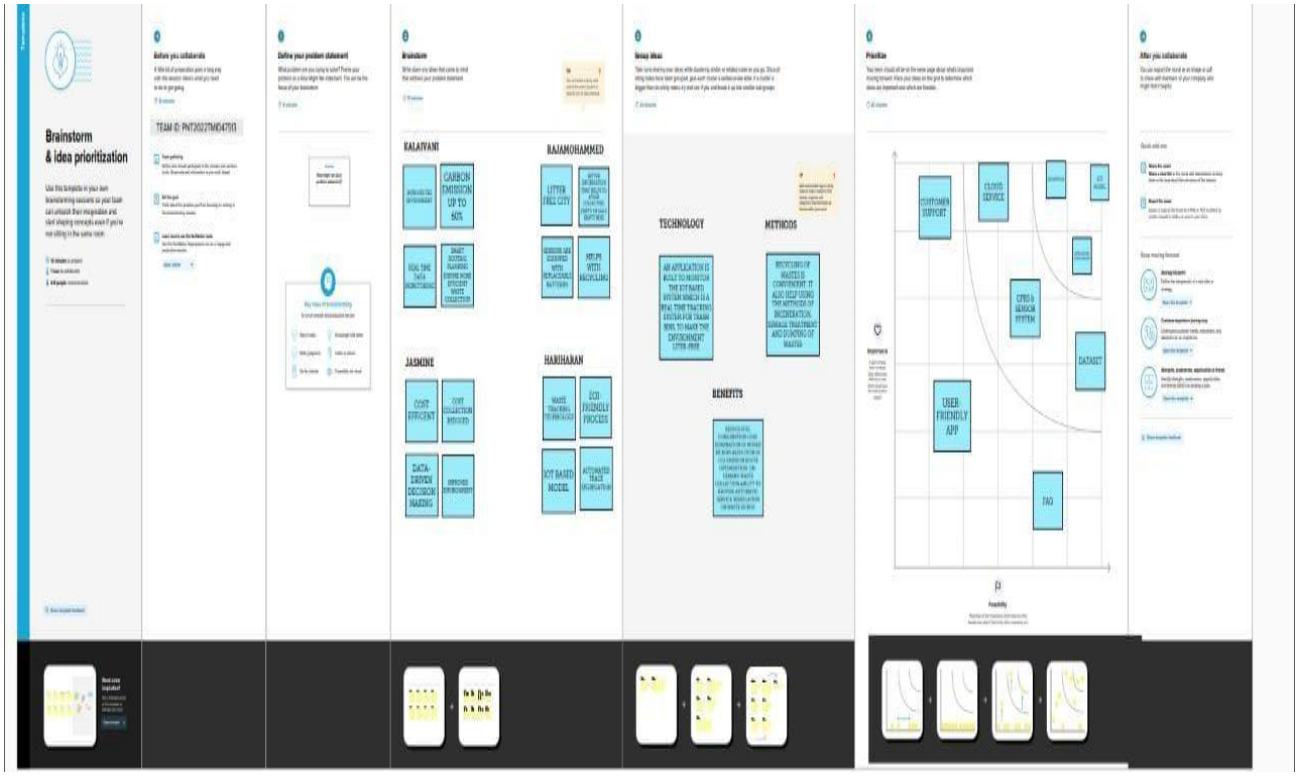
PS 2	Individual Working for a private limited corporation.	Get rid of the example of a surplus of waste.	The trash bins are always filled.	I occupy a metropolitan where there is activity is invariably crowd.	Worried

3. IDEATION & PROPOSED SOLUTION

3.1Empathy map Canvas



3.2 Ideation & Brainstorming



3.3 Proposed Solution

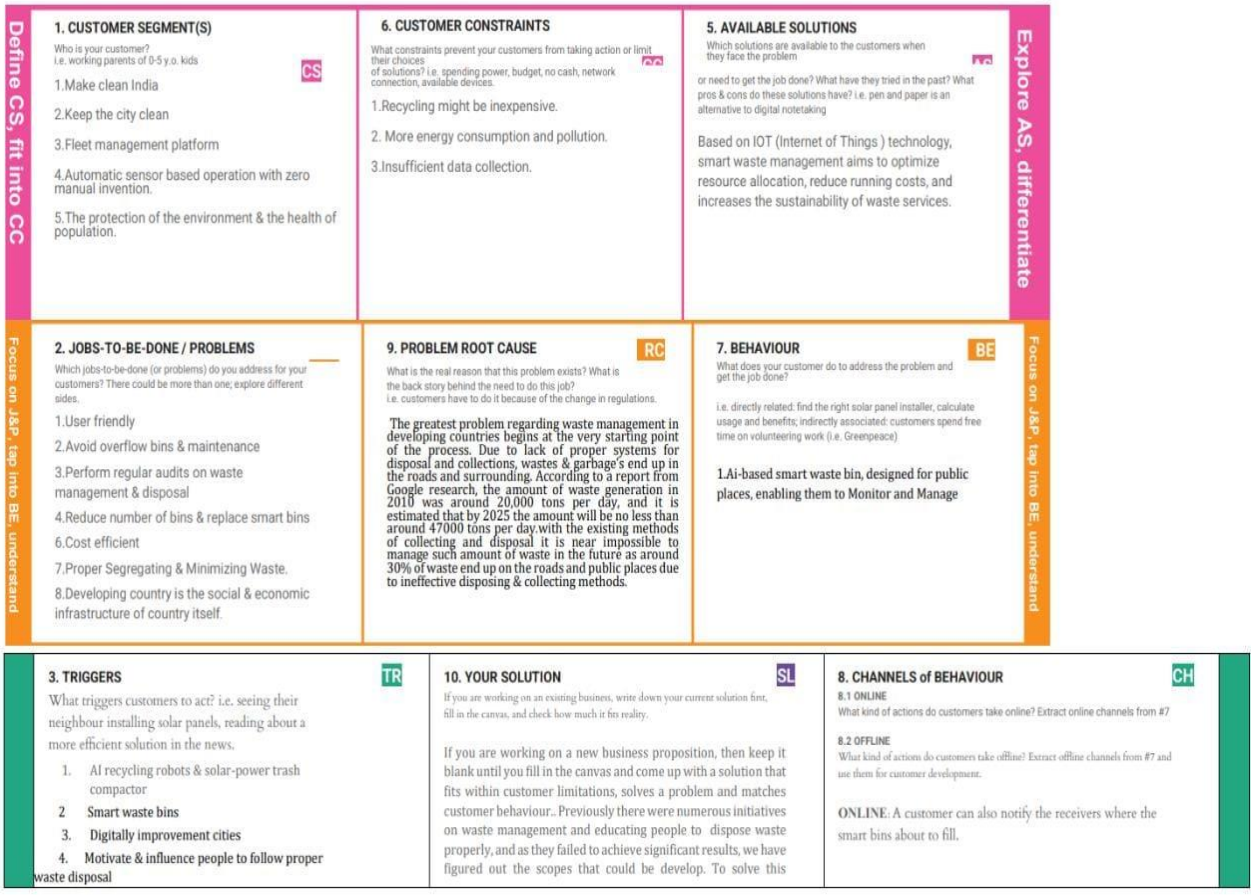
S. No	Parameter	Description
01.	Problem Statement (Problem to be solved)	This project is based on management of waste in metropolitan cities, where the garbage collection system is not optimized. An inefficient waste management creating serious environmental issues like bacterial infections, climate changes, pollution. Here smart waste management system is used for proper disposal of waste by using IOT.
02	Idea / Solution description	The key objectives are, 1.The proposed system would monitor the waste and manage the overall

		collection using Internet Of Things (IOT) 2. The proposed system monitors the waste bins, whenever the waste bins gets filled, it notifies to the receiver. 3. In proposed system, the received signal indicates the waste bin status and controlling system.
03	Novelty / Uniqueness	We are going to establish SWM in our college. But in practical the hard thing is to operate cleaner. So here our team planned to build a wrist band to them, that is indicated by light blinking when the bins are filled.
04	Social Impact / Customer Satisfaction	A reduction in the number of waste collections needed by up to 80%, resulting in less manpower, emissions, fuel use and traffic congestion. A reduction in the number of waste bins needed. Analytics data to manage collection routes and the placement of bins more effectively. Major satisfaction of waste managementconservation of natural resources, reduction of air, water & land pollution, support for community development
05	Business model	Waste Management generates revenue through the provision of various waste management and disposal services and recycling solutions residential, commercial, Industrial and Municipal clients. 2.Corporate and other comprising the Company's other activities, including its development and operation of landfill gas-toenergy facilities in India, and its recycling brokerage services, as well as various corporate functions.
06	Scalability of the Solution	To make an city smart, moreover many discussion are undertaken all

		around the world to solve this issue. The proposed system uses sensor and communication technologies where waste data is collected from the smart bin, in real time, and then transmitted to an online platform where citizens can access and check the availability of the compartments scattered around a city.

3.4 Problem Solution Fit

Project Title: Smart waste Management for Metropolitan cities **Project Design Phase-I** **Team ID:** PNT2022TMID47513



4.1 Functional Requirements

Following are the functional requirements of the proposed solution.

FR NO	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task
1	Detailed bin inventory	Bins which are seen on the maps by GPS location, and it is visited at any time by street view. Bins are visible in maps by different color circle. We can see the garbage bin details in the dashboard -capacity, recyclable or nonrecyclable waste, waste measurement, GPS location and pick recognition.
2	Bin Monitoring	Waste which are filled in bins are monitored by sensors. Based on the previous data, the tool predicts when will the bin fill. Smart sensor recognize each and every action takesplace. Hence it will check the last collected data. With the real time data & predictions, we can eliminate the overflowing of bins.
3	Expensive bins	It helps us to identify bins that drive up collection costs. The tool calculate a rating of each bins in terms of collection cost.
4	Eliminates unefficient picks	1.The sensor recognize picks. 2.By the data filled on the bin, pick recognition, we can show how full the bins you collect are. 3.Eliminates the collection of empty bins.
5	Adjust bin distribution	1.Initially we have to ensure the most optimal distribution of bins. 2.Identifies area with either dense or sparse bin distribution. 3.Based on previous data, we can adjust bin capacity or location.
6	Waste collection routes	Based on current bin fill-levels and predictions of reaching full capacity, we have ready to respond and schedule. We have to compare planned and executed

		routes to identify any inconsistencies.

4.2 Non-Functional requirements

Following are the non-functional requirements of the proposed solution

NFR No	Non-Functional Requirements	Description
NFR-1	Usability	The device verifies that the usability is a special and important to analyse user requirements which will the design quality. In the design process with user experience as the core, the analysis of users product usability can indeed help designers better understand users potential needs in waste management, behaviour and experience.
NFR-2	Security	1.Use reusable and recyclable bottles 2.Avoid non-recyclable plastic container. 3.Use reusable bag
NFR-3	Realibility	This project (Smart waste management system) is all about creating better work experience for waste collectors and drivers. Waste collector will spend their time more efficiently instead of driving the same collection routes and servicing empty bin.
NFR-4	Performance	By using the various IoT networks, the sensors send the data to smart waste management software system, a cloud platform, for data-driven daily operations, and available waste. User are provided with data-driven decision making, and optimization of waste collection route reduction by at least 35%

NFR-5	Availability	By developing resilient hardware and software we empower the cities and countries to manage waste smarter
NFR-6	Scalability	Using the smart bins reduce the number of bins inside cities and urban areas because we able to monitor the garbage any time more cost effect and scalability when we move to smarter

5.PROJECT DESIGN

5.1 Data Flow Diagram

A Data flow diagram(DFD) is a traditional visual representation of the information flows within the system. A neat and clear DFD can depict the right amount of the system requirement graphically.

It shows how data enters and leaves the system, what changes the information, and where data is stored.

A smart waste management platform uses analytics to translate the data gather in your bins into actionable insights to help you improve your waste services.

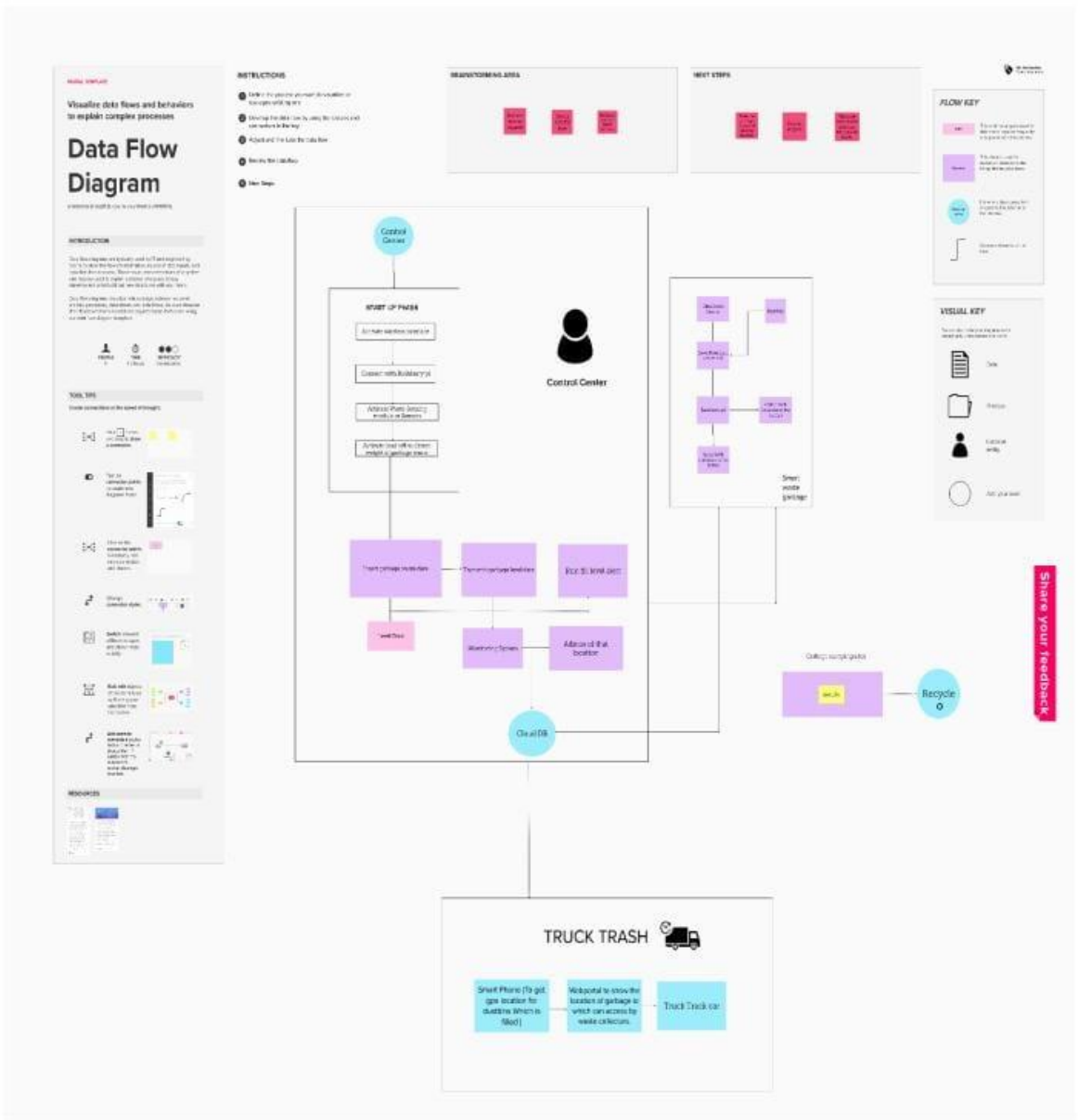
We can receive the data metric such as:

- The first test conducted is the situation where the garbage bin is empty or its garbage bin is empty or its garbage level
- Then the bin is filled with more garbage until its level is more surpassed threshold value, Which is set to 80% then the first warning SMS id being depicted.
- The first notification sent by the system, once the waste level of 85%full, the second notification send by the system. Then garbage need to be collected.
- Location prone to overflow.
- The number of bins needed to avoid overflowing waste.
- The number of collection service that could be saved.
- The amount of fuel that could be saved.

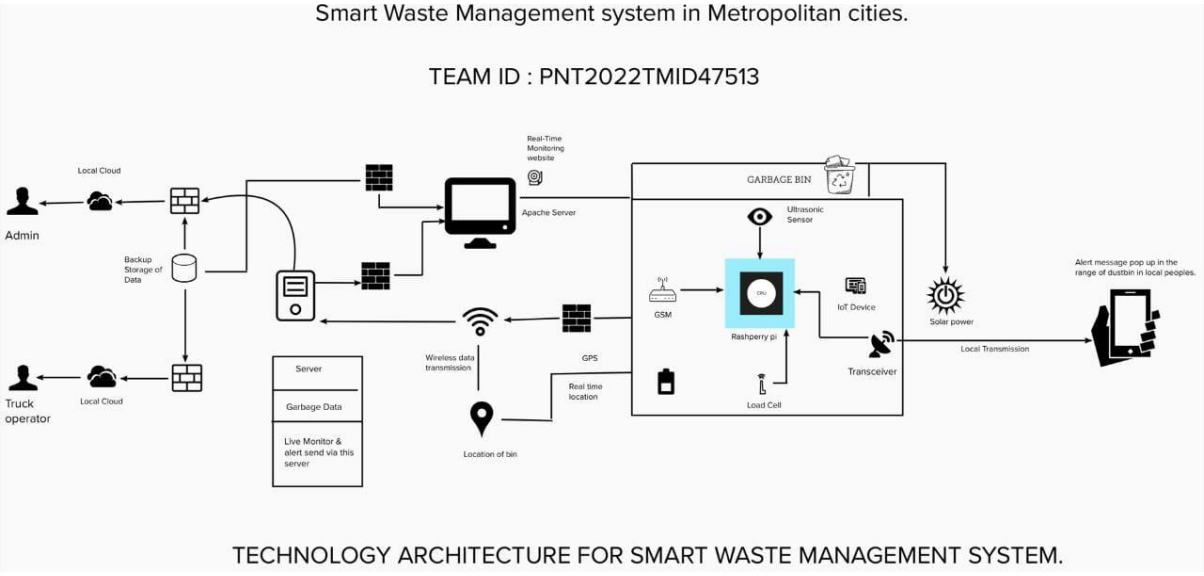
5.2 Data flow diagram.

Smart waste Management system for Metropolitan Cities

Team ID:PNT2022TMID47513



5.2 Solution & Technical Architecture



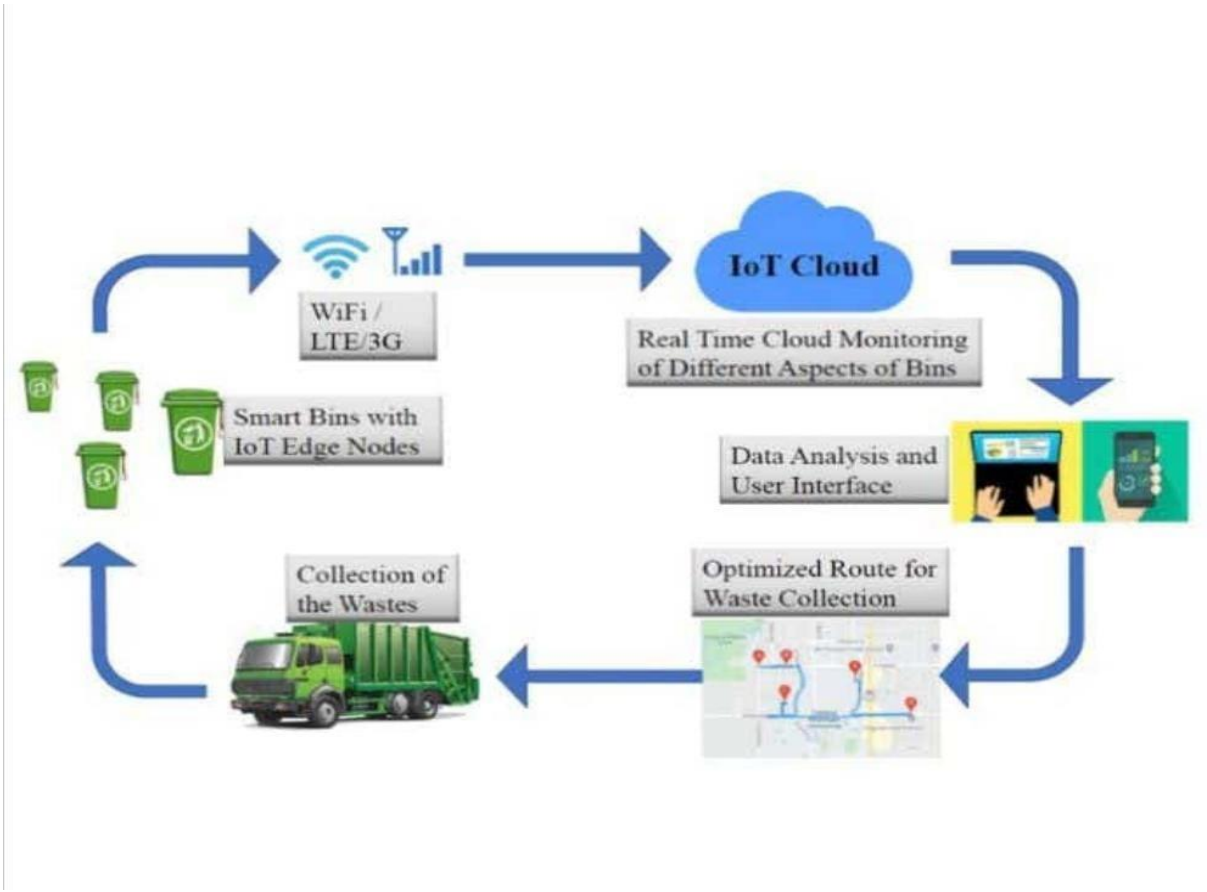


Table 1 : Components & Technology:

S.no	Components	Description	Technology
1.	User Interface	Mobile Application	HTML, CSS, JavaScript.
2.	Application Logic	Logic for a process in the application	JavaScript
3.	Database	Data type, Configuration etc.	Firebase, IBM cloud
4.	Cloud Database	Database service on cloud	IBM cloud.
5.	File storage	File storage requirements	Local filesystem & IBM cloud.
6.	Infrastructure	Application Deployment on cloud local server configuration	Local & Cloud Foundry

Table 1 : Application Characteristics:

S.No	Characteristics	Description	Technology
1	Open-Source frameworks	Github	Internal hosting service.
2	Security Implementation	Application Security: Verification Code	Network Automation.
3	Scalable Architecture	It provide the room for expansion more database of smart bins added additionally can be updated.	Cloud Storage
4	Availability	As the system control is connected to web server it is available 24*7 & can be accessed whenever needed.	Server, Appleixe, reple.
5	Performance	Performance is high, it uses 5mb caches.	Wireless Sensor Network.

5.3 User Stories

User type	Functional Requirement	User Story Number	User story / Task	Acceptance criteria	Priority	Release
Admin	Login	USN-1	As an Admin, I gave user id & password for every workers & manage them.	I can manage web account / dashboard.	Medium	Sprint-1
Co-Admin	Login	USN-2	As a Co admin, I'll manage garbage level monitor. If garbage get filling alert I'll post location & garbage id to trash truck	I can manage garbage monitoring.	High	Sprint-1
Truck Driver	Login	USN-3	As Truck Driver, I'll follow the route send by Co Admin to reach the filled garbage.	I can drive to reach the garbage filled route in shortest route given.	Medium	Sprint-2
Local Garbage Collector	Login	USN-4	As a Waste Collector, I'll collect all the trash from garbage & load into garbage truck &send them to landfill.	I can collect trash & pulled to truck & send off.	Medium	Sprint-3
Municipality officer	Login	USN-5	As a Municipality,	I can manage all	High	Sprint-4

			I'll check the process are happening in discipline manner without any issues.	these process going good.		

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Title	Description	Date
Literature Survey & Information Gathering	Literature survey on the selected project & gathering information by referring the, technical papers, research publications etc..	**4 October 2022**
Prepare Empathy Map	Prepare Empathy map Canvas to capture the users pains & Gains, Prepare list of problem and statements.	**5 October 2022**
Ideation	List the by organizing brainstorm session & prioritize the top 3 ideas based on the feasibility & importance.	**5 October 2022**
Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc..	**5 October 2022**
Proposed Solution Fit	Prepare problem – solution fit document.	**6 October 2022**
Solution Architecture	Prepare solution architecture document.	**6 October 2022**

Customer Journey	Prepare the Customer journey maps to understand the user interactions & experiences with the application (entry to exit).	26 October 2022
Functional requirements	Prepare the functional requirement document.	12 October 2022
Data flow Diagram	Draw the data flow diagram and submit for review.	15 October 2022
Technology Architecture	Prepare the technology architecture diagram.	16 October 2022
Prepare Milestone & Activity List	Prepare the milestone & activity list of the project.	27 October 2022
Project Development Delivery of Sprint-1, 2, 3 & 4.	Develop & submit the development code by testing it.	IN PROCESS

Project Planning (Product Backlog, Sprint Planning, Stories, Story points)

Sprint	**Functional Requirements**	**User Story Number**	**User Story/Task**	**Story Points**	**Priority**	**Team Members**
Sprint-1	Login	USN-1	As a Administrator, I need to give user id & passcode for ever workers over there in municipality.	10	High	Kalaivani
Sprint-1	Login	USN-2	As a Co-Admin, I'll Control the Waste level by monitoring them Via real time web	10	High	Raja Mohammed

			portal. Once the filling happens, I'll notify trash truck with location of bin with bin ID.			
Sprint-2	Dashboard	USN-3	As a Truck Driver, I 'll follow Co - Admin 's Instruction to reach the filling bin in short roots and save time	20	High	Jasmine
Sprint-3	Dashboard	USN-4	As a Local Garbage collector, I'll gather all the waste from the garbage, load it onto a garbage truck, and deliver it to landfills.	20	Low	Hariharan
Sprint-4	Dashboard	USN-5	As a municipality officer, I 'll make sure everything is proceeding as planning and without any problems.	20	Medium	Kalaivani

6.2 Sprint Delivery Plan

Sprint	**Total Story Points**	**Duration**	**Sprint Start Date**	**Sprint end date**	**Story Point Completed**	**Sprint release date**
Sprint-1	20	6 days	24 oct 2022	29 oct 2022	20	29 oct 2022
Sprint-2	20	6 days	31 oct 2022	05 Nov 2022	20	05 Nov 2022

Sprint-3	20	6 days	7 Nov 2022	12 oct 2022	20	12 oct 2022
Sprint-4	20	6 days	14 Nov 2022	19 Nov 2022	20	19 oct 2022

7. CODING & SOLUTIONING

7.1 Code 1 (Bin-1)

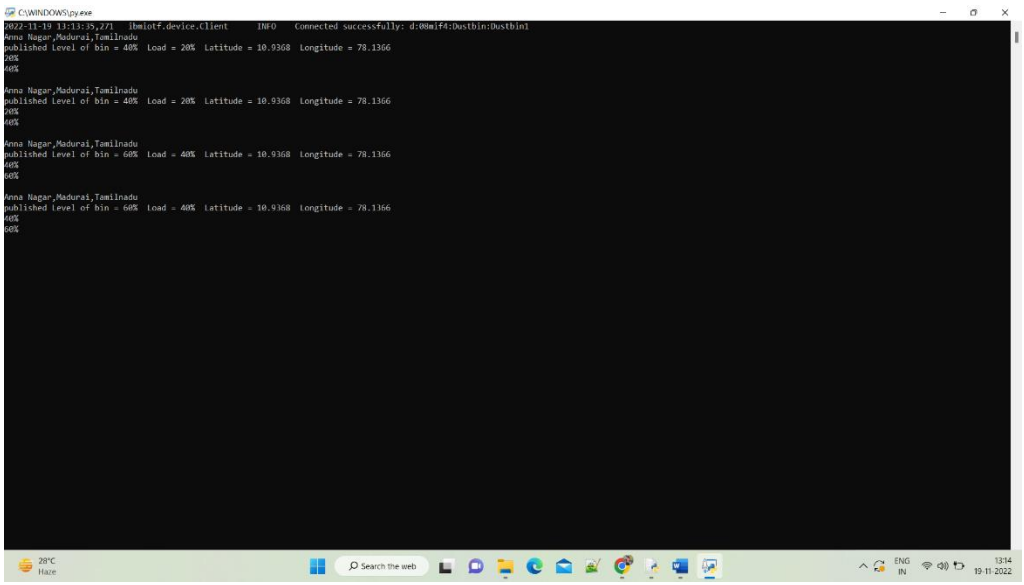
```
import time
import random
import sys
import requests
import json
import ibmiotf.application
import ibmiotf.device



# watson device details


organization = "08mif4"
devicType =  "Dustbin"
deviceId = "Dustbin1"
authMethod= "token"
authToken= "123456789"


#generate random values for random variables (Distance and load)




def myCommandCallback(cmd):
    global a
    print("command recieved:%s" %cmd.data['command'])
    control=cmd.data['command']
```

```
    print(control)


try:
        deviceOptions={"org": organization, "type": devicType,"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 "Distance" with value integer value into the cloud as a type of event for every 10 seconds
deviceCli.connect()


while True:


    Distance= random.randint(1,75)
    Loadcell= random.randint(0,20)
    data= {'dist':Distance,'load':Loadcell}
    if Loadcell<5 and Loadcell>0:
        load="20%"
    elif Loadcell<10 and Loadcell>5:
        load="40%"
    elif Loadcell<15 and Loadcell>10:
        load="60%"
    elif Loadcell<18 and Loadcell>15:
        load="80%"
    elif Loadcell<20 and Loadcell>18:
        load="90%"
    else:
        load="100%"


    if Distance<7 and Distance>1:
```

```
        level="90%"
    elif Distance<15 and Distance>7:
        level="80%"
    elif Distance<30 and Distance>15:
        level="60%"
    elif Distance<45 and Distance>30:
        level="40%"
    elif Distance<60 and Distance>45:
        level="20%"
    elif Distance<75 and Distance>60:
        level="10%"
    else:
        level="0%"

    if level=="90%" or load=="90%":
         warn="Alert:"Dustbin is almost filled"
    else:
         warn="

    def myOnPublishCallback(latitude=10.9368,longitude=78.1366):
        print("Anna Nagar,Madurai,Tamilnadu")
        print("published Level of bin = %s " %level,"Load = %s " %load, "Latitude = %s "
%latitude,"Longitude = %s " %longitude)
        print(load)
        print(level)
        print(warn)

    time.sleep(10)

    success=deviceCli.publishEvent ("IoTSensor","json",warn,qos=0,on_publish=
myOnPublishCallback)
```

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



    if not success:
        print("not connected to ibmiot")
    time.sleep(20)



    deviceCli.commandCallback=myCommandCallback
#disconnect the device
deviceCli.disconnect()
```

OUTPUT:



7.2 Code 2 (Bin-2)

```
import time
import random
import sys
```

```
import requests
import json
import ibmiotf.application
import ibmiotf.device


# watson device details

organization = "08mif4"
devicType =  "DustbinA"
deviceId = "Dustbin2"
authMethod= "token"
authToken= "123456789"

#generate random values for random variables (Distance and load)



def myCommandCallback(cmd):
    global a
    print("command recieved:%s" %cmd.data['command'])
    control=cmd.data['command']
    print(control)


try:
        deviceOptions={"org": organization, "type": devicType,"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 "Distance" with value integer value into the cloud as a type of event for every 10 seconds
deviceCli.connect()

while True:

    Distance= random.randint(1,75)
    Loadcell= random.randint(0,20)
    data= {'dist':Distance,'load':Loadcell}
    if Loadcell<5 and Loadcell>0:
        load="20%"
    elif Loadcell<10 and Loadcell>5:
        load="40%"
    elif Loadcell<15 and Loadcell>10:
        load="60%"
    elif Loadcell<18 and Loadcell>15:
        load="80%"
    elif Loadcell<20 and Loadcell>18:
        load="90%"
    else:
        load="100%"

    if Distance<7 and Distance>1:
        level="90%"
    elif Distance<15 and Distance>7:
        level="80%"
    elif Distance<30 and Distance>15:
        level="60%"
    elif Distance<45 and Distance>30:
        level="40%"
    elif Distance<60 and Distance>45:
        level="20%"
    elif Distance<75 and Distance>60:
```

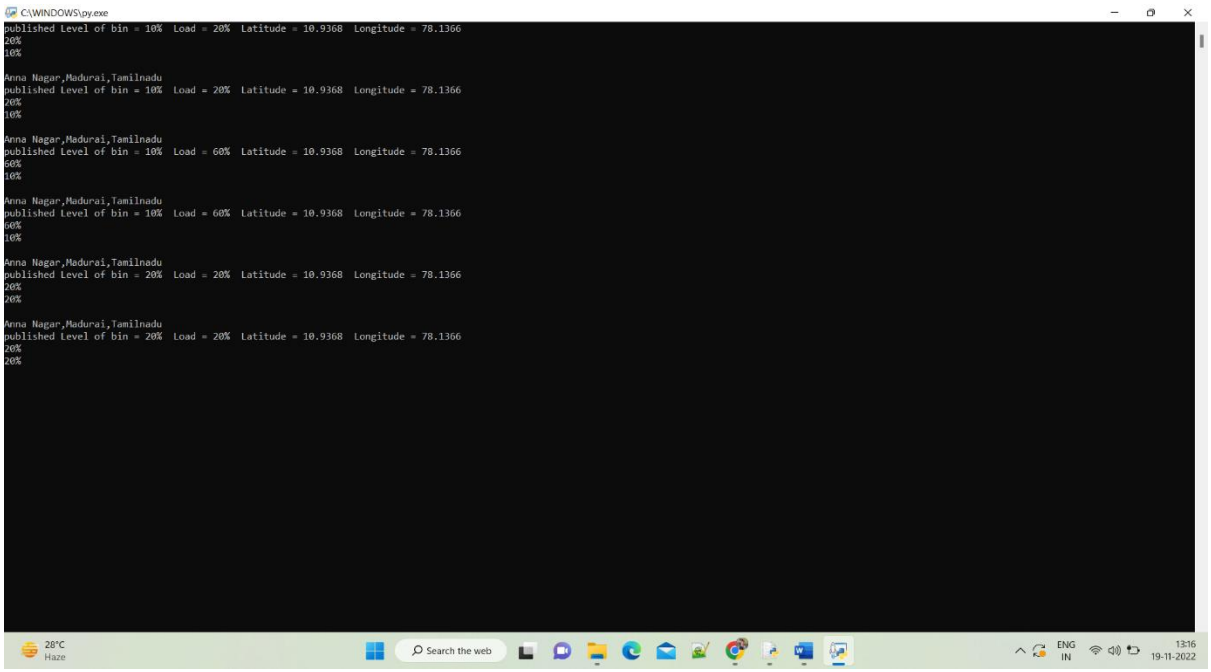
```
        level="10%"
    else:
        level="0%"

    if level=="90%" or load=="90%":
         warn="Alert:"Dustbin is almost filled"
    else:
         warn="

    def myOnPublishCallback(latitude=10.9368,longitude=78.1366):
        print("Anna Nagar,Madurai,Tamilnadu")
        print("published Level of bin = %s " %level,"Load = %s " %load, "Latitude = %s " %latitude,"Longitude = %s " %longitude)
        print(load)
        print(level)
        print(warn)

    time.sleep(10)

    success=deviceCli.publishEvent ("IoTSensor","json",warn,qos=0,on_publish= myOnPublishCallback)

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

    if not success:
        print("not connected to ibmiot")
    time.sleep(20)
```

deviceCli.commandCallback=myCommandCallback

#disconnect the device

deviceCli.disconnect()

OUTPUT 2:



7.3 Code 3 (Bin-3)

import time

import random

import sys

import requests

import json

import ibmiotf.application

import ibmiotf.device

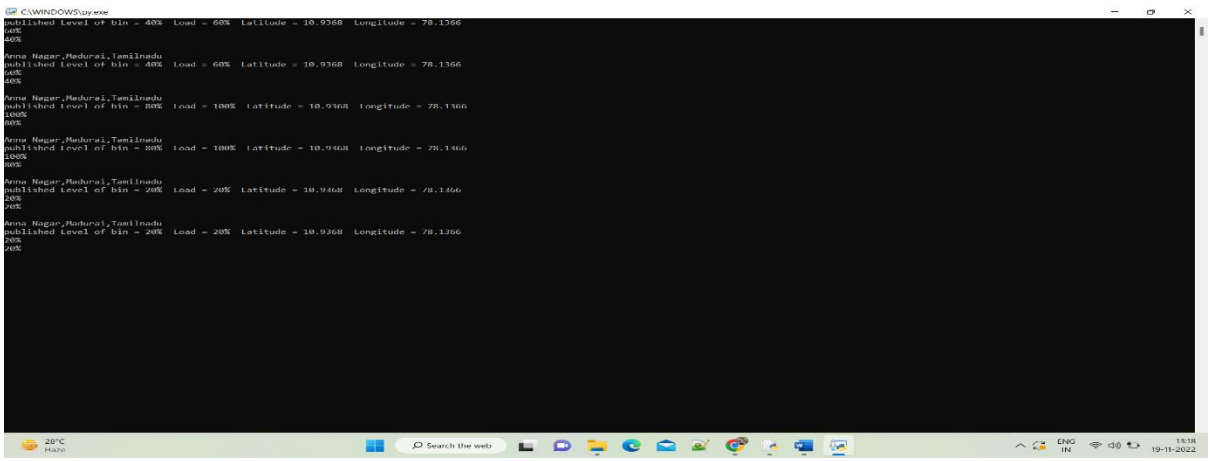
watson device details

```
organization = "08mif4"
devicType =  "DustbinB"
deviceId = "Dustbin2"
authMethod= "token"
authToken= "123456789"

#generate random values for random variables (Distance and load)



def myCommandCallback(cmd):
    global a
    print("command recieved:%s" %cmd.data['command'])
    control=cmd.data['command']
    print(control)


try:
        deviceOptions={"org": organization, "type": devicType,"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 "Distance" with value integer value into the cloud as a type of event for every 10 seconds
deviceCli.connect()

while True:


    Distance= random.randint(1,75)
    Loadcell= random.randint(0,20)
    data= {'dist':Distance,'load':Loadcell}
```

```
if Loadcell<5 and Loadcell>0:
    load="20%"
elif Loadcell<10 and Loadcell>5:
    load="40%"
elif Loadcell<15 and Loadcell>10:
    load="60%"
elif Loadcell<18 and Loadcell>15:
    load="80%"
elif Loadcell<20 and Loadcell>18:
    load="90%"
else:
    load="100%"

if Distance<7 and Distance>1:
    level="90%"
elif Distance<15 and Distance>7:
    level="80%"
elif Distance<30 and Distance>15:
    level="60%"
elif Distance<45 and Distance>30:
    level="40%"
elif Distance<60 and Distance>45:
    level="20%"
elif Distance<75 and Distance>60:
    level="10%"
else:
    level="0%"

if level=="90%" or load=="90%":
      warn="Alert:"Dustbin is almost filled"
else:
      warn="
```

```
    def myOnPublishCallback(latitude=10.9368,longitude=78.1366):
        print("Anna Nagar,Madurai,Tamilnadu")
        print("published Level of bin = %s " %level,"Load = %s " %load, "Latitude = %s " %latitude,"Longitude = %s " %longitude)
        print(load)
        print(level)
        print(warn)

    time.sleep(10)

    success=deviceCli.publishEvent ("IoTSensor","json",warn,qos=0,on_publish=myOnPublishCallback)

    success=deviceCli.publishEvent ("IoTSensor","json",data,qos=0,on_publish=myOnPublishCallback)
   if not success:
        print("not connected to ibmiot")
    time.sleep(20)
  deviceCli.commandCallback=myCommandCallback
#disconnect the device
deviceCli.disconnect()
```

OUTPUT:



7.4 Code 4 (Bin-4)

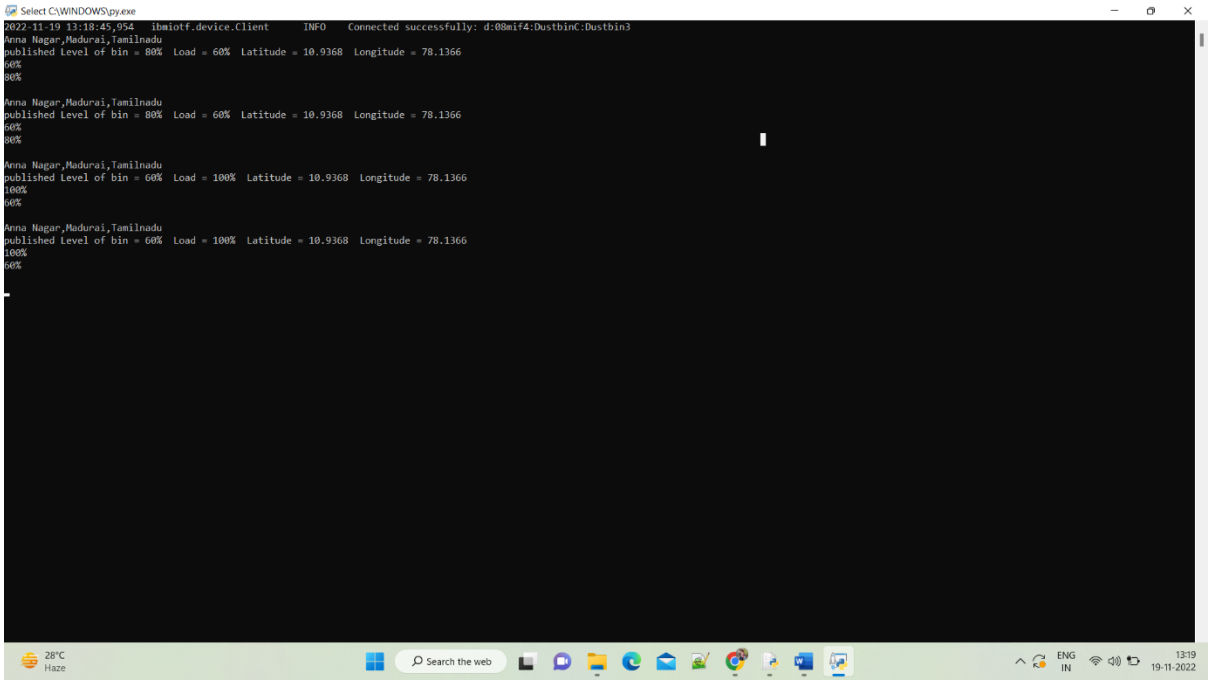
```
import time
import random
import sys
import requests
import json
import ibmiotf.application
import ibmiotf.device



# watson device details


organization = "08mif4"
devicType =  "DustbinC"
deviceId = "Dustbin3"
authMethod= "token"
authToken= "123456789"

#generate random values for random variables (Distance and load)




def myCommandCallback(cmd):
    global a
    print("command recieved:%s" %cmd.data['command'])
    control=cmd.data['command']
    print(control)


try:
        deviceOptions={"org": organization, "type": devicType,"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 "Distance" with value integer value into the cloud as a type of event for every 10 seconds
deviceCli.connect()

while True:

    Distance= random.randint(1,75)
    Loadcell= random.randint(0,20)
    data= {'dist':Distance,'load':Loadcell}
    if Loadcell<5 and Loadcell>0:
        load="20%"
    elif Loadcell<10 and Loadcell>5:
        load="40%"
    elif Loadcell<15 and Loadcell>10:
        load="60%"
    elif Loadcell<18 and Loadcell>15:
        load="80%"
    elif Loadcell<20 and Loadcell>18:
        load="90%"
    else:
        load="100%"

    if Distance<7 and Distance>1:
        level="90%"
    elif Distance<15 and Distance>7:
        level="80%"
    elif Distance<30 and Distance>15:
        level="60%"
    elif Distance<45 and Distance>30:
```

```
        level="40%"
    elif Distance<60 and Distance>45:
        level="20%"
    elif Distance<75 and Distance>60:
        level="10%"
    else:
        level="0%"

    if level=="90%" or load=="90%":
          warn="Alert:"Dustbin is almost filled"
    else:
          warn="

    def myOnPublishCallback(latitude=10.9368,longitude=78.1366):
        print("Anna Nagar,Madurai,Tamilnadu")
        print("published Level of bin = %s " %level,"Load = %s " %load, "Latitude = %s " %latitude,"Longitude = %s " %longitude)
        print(load)
        print(level)
        print(warn)

    time.sleep(10)
   success=deviceCli.publishEvent ("IoTSensor","json",warn,qos=0,on_publish= myOnPublishCallback)

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

 if not success:
        print("not connected to ibmiot")
    time.sleep(20)
deviceCli.commandCallback=myCommandCallback
#disconnect the device
```

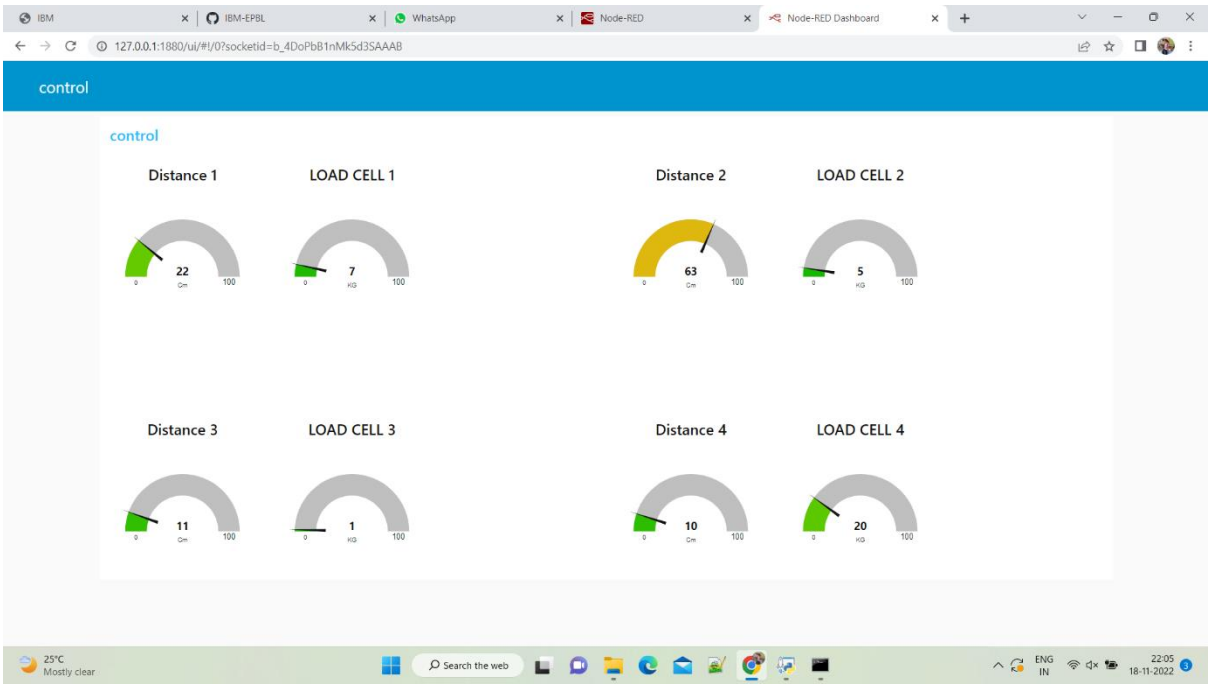
deviceCli.disconnect()

OUTPUT:



8. RESULT

8.1 Performance of Load Cells according to bin.



8.PROS & CONS

8.1 Pros

- Conservation of natural resources
- reduction of air, water and land pollution
- Support for community development
- No Missed Pickups.
- Reduced Overflows.
- Waste Generation Analysis.
- CO2 Emission Reduction.

8.2 Cons

- Increasing cost of the dustbin.
- Reliable Internet Connection is Crucial
- Compatibility Problems Between Devices
- Smart Home Tech is Not Suitable for All Houses.
- Setup and Configuration

9. CONCLUSION

It is clear that improper waste management practices have a significant impact on the natural environment and sustainable development in the study area. Thus, awareness about SWM impact on sound environmental development or/and sustainable development in seemingly low. Therefore, it is important that the SWM should be developed from the primary level. Waste storage and primary disposal are the dominant means of managing waste. Thus, it has caused significant challenges in the study area. Therefore, waste separation from the household level, proper storage, more efficient waste collection systems, and sustainable recovery and disposal practices are identified as needed processes in the study area. Considering the nature and components of waste generated by households and business places, the waste reduction, reuse, recycling and composting processes would be more suitable in managing the challenge. These management options should be integrated in a sustainable framework. Adequate consideration should be given to monitoring processes. Public education and properly planned waste management programs also need to be introduced into the current waste management system. Especially awareness programmes must be conducted in order to improve the knowledge about the importance of SWM for sound environmental development in the area.

10. FUTURE SCOPE

The future of Waste management starts and proceeds with technological adjustments. Like every other industry, to proceed, the waste management industry needs to become digitized and data-driven to advance its work field. The future is smart and competitive! Especially for businesses, they are required to be one step ahead of their competitors. When smart waste management solutions are applied over time, the data is collected. These data in hand sensors can be used to identify fill patterns, optimize driver routes and schedules, and reduce operational costs. These sensors' cost is steadily decreasing, making smart bins more feasible to implement and more attractive to companies or city leaders.

11. APPENDIX

11.1 Source code

```
import time
import random
import sys
import requests
import json
import ibmiotf.application
import ibmiotf.device


# watson device detailsorganization = "08mif4"
devicType =  "DustbinC"
deviceId = "Dustbin3"
authMethod= "token"
authToken= "123456789"


#generate random values for random variables (Distance and load)
def myCommandCallback(cmd):
    global a
    print("command recieved:%s" %cmd.data['command'])
```

```
    control=cmd.data['command']
    print(control)


try:
        deviceOptions={"org": organization, "type": devicType,"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 "Distance" with value integer value into the cloud as a type of event for every 10 seconds
deviceCli.connect()


while True:


    Distance= random.randint(1,75)
    Loadcell= random.randint(0,20)
    data= {'dist':Distance,'load':Loadcell}
    if Loadcell<5 and Loadcell>0:
        load="20%"
    elif Loadcell<10 and Loadcell>5:
        load="40%"
    elif Loadcell<15 and Loadcell>10:
        load="60%"
    elif Loadcell<18 and Loadcell>15:
        load="80%"
    elif Loadcell<20 and Loadcell>18:
        load="90%"
    else:
        load="100%"
```

```
    if Distance<7 and Distance>1:
        level="90%"
    elif Distance<15 and Distance>7:
        level="80%"
    elif Distance<30 and Distance>15:
        level="60%"
    elif Distance<45 and Distance>30:
        level="40%"
    elif Distance<60 and Distance>45:
        level="20%"
    elif Distance<75 and Distance>60:
        level="10%"
    else:
        level="0%"

    if level=="90%" or load=="90%":
         warn="Alert:"Dustbin is almost filled"
    else:
         warn="
  def myOnPublishCallback(latitude=10.9368,longitude=78.1366):
        print("Anna Nagar,Madurai,Tamilnadu")
        print("published Level of bin = %s " %level,"Load = %s " %load, "Latitude = %s "
%latitude,"Longitude = %s " %longitude)
        print(load)
        print(level)
        print(warn)
        time.sleep(10)


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

```
  if not success:
        print("not connected to ibmiot")
    time.sleep(20)

deviceCli.commandCallback=myCommandCallback
#disconnect the device
deviceCli.disconnect()
```

JSON CODE FOR NODE-RED


```
[
    {
        "id": "06e35f5f7e5cbbda",
        "type": "tab",
        "label": "Flow 1",
        "disabled": false,
        "info": "",
        "env": []
    },
    {
        "id": "6ed1189c17ed0439",
        "type": "tab",
        "label": "Flow 1",
        "disabled": false,
        "info": "",
        "env": []
    },
    {
        "id": "edb4621c238eb45a",
        "type": "tab",
        "label": "Flow 2",
        "disabled": false,
        "info": "",
```

```
    "env": []
},
{
    "id": "517648d26c93720d",
    "type": "ui_spacer",
    "z": "6ed1189c17ed0439",
    "name": "spacer",
    "group": "f3b64a4198b3c46c",
    "order": 3,
    "width": 4,
    "height": 1
},
{
    "id": "e3086afde0717b7f",
    "type": "ui_spacer",
    "z": "6ed1189c17ed0439",
    "name": "spacer",
    "group": "f3b64a4198b3c46c",
    "order": 6,
    "width": 4,
    "height": 1
},
{
    "id": "329502bdd11bd52a",
    "type": "ui_spacer",
    "z": "6ed1189c17ed0439",
    "name": "spacer",
    "group": "f3b64a4198b3c46c",
    "order": 7,
    "width": 4,
    "height": 1
},
```

```
{
    "id": "7321a3216f223378",
    "type": "ui_spacer",
    "z": "6ed1189c17ed0439",
    "name": "spacer",
    "group": "f3b64a4198b3c46c",
    "order": 8,
    "width": 4,
    "height": 1
},
{
    "id": "f830dc54889698c4",
    "type": "ui_spacer",
    "z": "6ed1189c17ed0439",
    "name": "spacer",
    "group": "f3b64a4198b3c46c",
    "order": 9,
    "width": 4,
    "height": 1
},
{
    "id": "3e4d0a1c6525c3dd",
    "type": "ui_spacer",
    "z": "6ed1189c17ed0439",
    "name": "spacer",
    "group": "f3b64a4198b3c46c",
    "order": 10,
    "width": 4,
    "height": 1
},
{
    "id": "f5fa3cc71404c6c2",
```

```
    "type": "ui_spacer",
    "z": "6ed1189c17ed0439",
    "name": "spacer",
    "group": "f3b64a4198b3c46c",
    "order": 11,
    "width": 4,
    "height": 1
},
{
    "id": "eacd5aaaa83c8e0a",
    "type": "ui_spacer",
    "z": "6ed1189c17ed0439",
    "name": "spacer",
    "group": "f3b64a4198b3c46c",
    "order": 12,
    "width": 4,
    "height": 1
},
{
    "id": "e9c145684f7c3c5b",
    "type": "ui_spacer",
    "z": "6ed1189c17ed0439",
    "name": "spacer",
    "group": "f3b64a4198b3c46c",
    "order": 13,
    "width": 24,
    "height": 1
},
{
    "id": "4bf3afc9ad5605e6",
    "type": "ui_spacer",
    "z": "6ed1189c17ed0439",
```

```
    "name": "spacer",
    "group": "f3b64a4198b3c46c",
    "order": 14,
    "width": 24,
    "height": 1
},
{
    "id": "84a1cdcb1acbd2ce",
    "type": "ui_spacer",
    "z": "6ed1189c17ed0439",
    "name": "spacer",
    "group": "f3b64a4198b3c46c",
    "order": 17,
    "width": 4,
    "height": 1
},
{
    "id": "8e3c9b05b4f4659f",
    "type": "ui_spacer",
    "z": "6ed1189c17ed0439",
    "name": "spacer",
    "group": "f3b64a4198b3c46c",
    "order": 20,
    "width": 4,
    "height": 1
},
{
    "id": "d3debe038a3f028a",
    "type": "ui_spacer",
    "z": "6ed1189c17ed0439",
    "name": "spacer",
    "group": "f3b64a4198b3c46c",
```

```
    "order": 21,
    "width": 4,
    "height": 1
},
{
    "id": "cdf1f0030c949ca8",
    "type": "ui_spacer",
    "z": "6ed1189c17ed0439",
    "name": "spacer",
    "group": "f3b64a4198b3c46c",
    "order": 22,
    "width": 4,
    "height": 1
},
{
    "id": "28a4dbc012c368f4",
    "type": "ui_spacer",
    "z": "6ed1189c17ed0439",
    "name": "spacer",
    "group": "f3b64a4198b3c46c",
    "order": 23,
    "width": 4,
    "height": 1
},
{
    "id": "bbd70391a5bc8257",
    "type": "ui_spacer",
    "z": "6ed1189c17ed0439",
    "name": "spacer",
    "group": "f3b64a4198b3c46c",
    "order": 24,
    "width": 4,
```

```
    "height": 1
  },
  {
    "id": "b94286c6f5fc0e17",
    "type": "ui_spacer",
    "z": "6ed1189c17ed0439",
    "name": "spacer",
    "group": "f3b64a4198b3c46c",
    "order": 25,
    "width": 4,
    "height": 1
  },
  {
    "id": "4dcd3b1b88e0508e",
    "type": "ui_spacer",
    "z": "6ed1189c17ed0439",
    "name": "spacer",
    "group": "f3b64a4198b3c46c",
    "order": 26,
    "width": 4,
    "height": 1
  },
  {
    "id": "bf9996433728395e",
    "type": "ibmiot",
    "name": "Kalai",
    "keepalive": "60",
    "serverName": "",
    "cleansession": true,
    "appId": "",
    "shared": false
  },
```

```
    {
        "id": "f3b64a4198b3c46c",
        "type": "ui_group",
        "name": "control",
        "tab": "2b19469befff9adb",
        "order": 2,
        "disp": true,
        "width": "24",
        "collapse": false,
        "className": ""
    },
    {
        "id": "2b19469befff9adb",
        "type": "ui_tab",
        "name": "control",
        "icon": "control",
        "disabled": false,
        "hidden": false
    },
    {
        "id": "81b33744381f342d",
        "type": "ui_base",
        "theme": {
            "name": "theme-light",
            "lightTheme": {
                "default": "#0094CE",
                "baseColor": "#0094CE",
                "baseFont": "-apple-system,BlinkMacSystemFont,Segoe UI,Roboto,Oxygen-
Sans,Ubuntu,Cantarell,Helvetica Neue,sans-serif",
                "edited": false
            },
            "darkTheme": {
                "default": "#097479",
```

```
            "baseColor": "#097479",
            "baseFont": "-apple-system,BlinkMacSystemFont,Segoe UI,Roboto,Oxygen-Sans,Ubuntu,Cantarell,Helvetica Neue,sans-serif",
            "edited": false
        },
        "customTheme": {
            "name": "Untitled Theme 1",
            "default": "#4B7930",
            "baseColor": "#4B7930",
            "baseFont": "-apple-system,BlinkMacSystemFont,Segoe UI,Roboto,Oxygen-Sans,Ubuntu,Cantarell,Helvetica Neue,sans-serif"
        },
        "themeState": {
            "base-color": {
                "default": "#0094CE",
                "value": "#0094CE",
                "edited": false
            },
            "page-titlebar-backgroundColor": {
                "value": "#0094CE",
                "edited": false
            },
            "page-backgroundColor": {
                "value": "#fafafa",
                "edited": false
            },
            "page-sidebar-backgroundColor": {
                "value": "#ffffff",
                "edited": false
            },
            "group-textColor": {
                "value": "#1bbfff",
                "edited": false
```

```
        },
        "group-borderColor": {
          "value": "#ffffff",
          "edited": false
        },
        "group-backgroundColor": {
          "value": "#ffffff",
          "edited": false
        },
        "widget-textColor": {
          "value": "#111111",
          "edited": false
        },
        "widget-backgroundColor": {
          "value": "#0094ce",
          "edited": false
        },
        "widget-borderColor": {
          "value": "#ffffff",
          "edited": false
        },
        "base-font": {
          "value": "-apple-system,BlinkMacSystemFont,Segoe UI,Roboto,Oxygen-Sans,Ubuntu,Cantarell,Helvetica Neue,sans-serif"
        }
      },
      "angularTheme": {
        "primary": "indigo",
        "accents": "blue",
        "warn": "red",
        "background": "grey",
        "palette": "light"
      }
```

```
    },
    "site": {
        "name": "Node-RED Dashboard",
        "hideToolbar": "false",
        "allowSwipe": "false",
        "lockMenu": "false",
        "allowTempTheme": "true",
        "dateFormat": "DD/MM/YYYY",
        "sizes": {
            "sx": 48,
            "sy": 48,
            "gx": 6,
            "gy": 6,
            "cx": 6,
            "cy": 6,
            "px": 0,
            "py": 0
        }
    }
},
{
    "id": "ce1790a002f55f3a",
    "type": "ibmiot in",
    "z": "6ed1189c17ed0439",
    "authentication": "apiKey",
    "apiKey": "bf9996433728395e",
    "inputType": "evt",
    "logicalInterface": "",
    "ruleId": "",
    "deviceId": "Dustbin1",
    "applicationId": "",
    "deviceType": "Dustbin",
```

```
    "eventType": "+",
    "commandType": "",
    "format": "json",
    "name": "IBM IoT",
    "service": "registered",
    "allDevices": "",
    "allApplications": "",
    "allDeviceTypes": false,
    "allLogicalInterfaces": "",
    "allEvents": true,
    "allCommands": "",
    "allFormats": "",
    "qos": 0,
    "x": 250,
    "y": 180,
    "wires": [
      [
        "b678812da97d9d1a",
        "f720c62cad238799",
        "35b263513ea4f373"
      ]
    ]
  },
  {
    "id": "b678812da97d9d1a",
    "type": "debug",
    "z": "6ed1189c17ed0439",
    "name": "msg.payload",
    "active": true,
    "tosidebar": true,
    "console": false,
    "tostatus": false,
```

```
    "complete": "payload",
    "targetType": "msg",
    "statusVal": "",
    "statusType": "auto",
    "x": 610,
    "y": 180,
    "wires": []
},
{
    "id": "f720c62cad238799",
    "type": "function",
    "z": "6ed1189c17ed0439",
    "name": "Distance 1",
    "func": "msg.payload = msg.payload.dist\nglobal.set('d',msg.payload)\nreturn msg;",
    "outputs": 1,
    "noerr": 0,
    "initialize": "",
    "finalize": "",
    "libs": [],
    "x": 430,
    "y": 220,
    "wires": [
        [
            "5dcbaf252dc78b06",
            "b678812da97d9d1a"
        ]
    ]
},
{
    "id": "35b263513ea4f373",
    "type": "function",
    "z": "6ed1189c17ed0439",
```

```
    "name": "LOAD cell 1",
    "func": "msg.payload =msg. payload.load\nglobal.set('l', msg.payload)\nreturn msg;",
    "outputs": 1,
    "noerr": 0,
    "initialize": "",
    "finalize": "",
    "libs": [],
    "x": 430,
    "y": 300,
    "wires": [
        [
            "b7ac8ba401c6cab8"
        ]
    ]
},
{
    "id": "5dcbaf252dc78b06",
    "type": "ui_gauge",
    "z": "6ed1189c17ed0439",
    "name": "",
    "group": "f3b64a4198b3c46c",
    "order": 1,
    "width": 4,
    "height": 4,
    "gtype": "gage",
    "title": "Distance 1",
    "label": "Cm",
    "format": "{{value}}",
    "min": 0,
    "max": "100",
    "colors": [
        "#00b500",
```

```
        "#e6e600",
        "#ca3838"
    ],
    "seg1": "",
    "seg2": "",
    "className": "",
    "x": 710,
    "y": 240,
    "wires": []
},
{
    "id": "b7ac8ba401c6cab8",
    "type": "ui_gauge",
    "z": "6ed1189c17ed0439",
    "name": "",
    "group": "f3b64a4198b3c46c",
    "order": 2,
    "width": 4,
    "height": 4,
    "gtype": "gage",
    "title": "LOAD CELL 1",
    "label": "KG",
    "format": "{{value}}",
    "min": 0,
    "max": "100",
    "colors": [
        "#00b500",
        "#e6e600",
        "#ca3838"
    ],
    "seg1": "",
    "seg2": "",
```

```
    "className": "",
    "x": 720,
    "y": 300,
    "wires": []
},
{
    "id": "5de18859cabb1a5d",
    "type": "http in",
    "z": "6ed1189c17ed0439",
    "name": "",
    "url": "/sensor",
    "method": "get",
    "upload": false,
    "swaggerDoc": "",
    "x": 210,
    "y": 420,
    "wires": [
        [
            "80650c336af78c61"
        ]
    ]
},
{
    "id": "5ab7d1be9c4e2831",
    "type": "http response",
    "z": "6ed1189c17ed0439",
    "name": "",
    "statusCode": "",
    "headers": {},
    "x": 710,
    "y": 400,
    "wires": []
```

```
},
{
    "id": "80650c336af78c61",
    "type": "function",
    "z": "6ed1189c17ed0439",
    "name": "function 1",
    "func": "msg.payload = { \"dist\": global.get('d'), \"load\": global.get('l')}\nreturn msg;",
    "outputs": 1,
    "noerr": 0,
    "initialize": "",
    "finalize": "",
    "libs": [],
    "x": 460,
    "y": 420,
    "wires": [
        [
            "5ab7d1be9c4e2831"
        ]
    ]
},
{
    "id": "e0022c1a3e189dea",
    "type": "ibmiot in",
    "z": "6ed1189c17ed0439",
    "authentication": "apiKey",
    "apiKey": "bf9996433728395e",
    "inputType": "evt",
    "logicalInterface": "",
    "ruleId": "",
    "deviceId": "Dustbin2",
    "applicationId": "",
    "deviceType": "DustbinA",
```

```
    "eventType": "+",
    "commandType": "",
    "format": "json",
    "name": "IBM IoT",
    "service": "registered",
    "allDevices": "",
    "allApplications": "",
    "allDeviceTypes": false,
    "allLogicalInterfaces": "",
    "allEvents": true,
    "allCommands": "",
    "allFormats": "",
    "qos": 0,
    "x": 250,
    "y": 500,
    "wires": [
      [
        "2a22e946c6d5f734",
        "233a55d8b0e40a46",
        "a5ed197df7ced05a"
      ]
    ]
  },
  {
    "id": "233a55d8b0e40a46",
    "type": "function",
    "z": "6ed1189c17ed0439",
    "name": "Distance 2",
    "func": "msg.payload = msg.payload.dist\nglobal.set('d',msg.payload)\nreturn msg;",
    "outputs": 1,
    "noerr": 0,
    "initialize": "",
```

```
    "finalize": "",
    "libs": [],
    "x": 450,
    "y": 540,
    "wires": [
      [
        "2a22e946c6d5f734",
        "9b44a1863803e38a"
      ]
    ]
  },
  {
    "id": "a5ed197df7ced05a",
    "type": "function",
    "z": "6ed1189c17ed0439",
    "name": "LOAD cell 2",
    "func": "msg.payload =msg. payload.load\nglobal.set('l', msg.payload)\nreturn msg;",
    "outputs": 1,
    "noerr": 0,
    "initialize": "",
    "finalize": "",
    "libs": [],
    "x": 450,
    "y": 600,
    "wires": [
      [
        "40ccb32035a0f55f"
      ]
    ]
  },
  {
    "id": "2a22e946c6d5f734",
```

```
    "type": "debug",
    "z": "6ed1189c17ed0439",
    "name": "msg.payload",
    "active": true,
    "tosidebar": true,
    "console": false,
    "tostatus": false,
    "complete": "payload",
    "targetType": "msg",
    "statusVal": "",
    "statusType": "auto",
    "x": 650,
    "y": 480,
    "wires": []
},
{
    "id": "9b44a1863803e38a",
    "type": "ui_gauge",
    "z": "6ed1189c17ed0439",
    "name": "",
    "group": "f3b64a4198b3c46c",
    "order": 4,
    "width": 4,
    "height": 4,
    "gtype": "gage",
    "title": "Distance 2",
    "label": "Cm",
    "format": "{{value}}",
    "min": 0,
    "max": "100",
    "colors": [
        "#00b500",
```

```
        "#e6e600",
        "#ca3838"
    ],
    "seg1": "",
    "seg2": "",
    "className": "",
    "x": 710,
    "y": 540,
    "wires": []
},
{
    "id": "40ccb32035a0f55f",
    "type": "ui_gauge",
    "z": "6ed1189c17ed0439",
    "name": "",
    "group": "f3b64a4198b3c46c",
    "order": 5,
    "width": 4,
    "height": 4,
    "gtype": "gage",
    "title": "LOAD CELL 2",
    "label": "KG",
    "format": "{{value}}",
    "min": 0,
    "max": "100",
    "colors": [
        "#00b500",
        "#e6e600",
        "#ca3838"
    ],
    "seg1": "",
    "seg2": "",
```

```
    "className": "",
    "x": 720,
    "y": 580,
    "wires": []
},
{
    "id": "60298a7291818343",
    "type": "http in",
    "z": "6ed1189c17ed0439",
    "name": "",
    "url": "/sensor",
    "method": "get",
    "upload": false,
    "swaggerDoc": "",
    "x": 190,
    "y": 660,
    "wires": [
        [
            "616151913ceb65e2"
        ]
    ]
},
{
    "id": "616151913ceb65e2",
    "type": "function",
    "z": "6ed1189c17ed0439",
    "name": "function 2",
    "func": "msg.payload = { \"dist\": global.get('d'), \"load\": global.get('l')}\nreturn msg;",
    "outputs": 1,
    "noerr": 0,
    "initialize": "",
    "finalize": "",
```

```
    "libs": [],
    "x": 420,
    "y": 660,
    "wires": [
      [
        "332391e22b2af8e8"
      ]
    ]
  },
  {
    "id": "332391e22b2af8e8",
    "type": "http response",
    "z": "6ed1189c17ed0439",
    "name": "",
    "statusCode": "",
    "headers": {},
    "x": 670,
    "y": 660,
    "wires": []
  },
  {
    "id": "4d33e05e616db2bb",
    "type": "ibmiot in",
    "z": "6ed1189c17ed0439",
    "authentication": "apiKey",
    "apiKey": "bf9996433728395e",
    "inputType": "evt",
    "logicalInterface": "",
    "ruleId": "",
    "deviceId": "Dustbin2",
    "applicationId": "",
    "deviceType": "DustbinB",
```

```
    "eventType": "+",
    "commandType": "",
    "format": "json",
    "name": "IBM IoT",
    "service": "registered",
    "allDevices": "",
    "allApplications": "",
    "allDeviceTypes": false,
    "allLogicalInterfaces": "",
    "allEvents": true,
    "allCommands": "",
    "allFormats": "",
    "qos": 0,
    "x": 250,
    "y": 760,
    "wires": [
      [
        "c7ddb56ba52e82df",
        "fbf611802e58a9d1",
        "231892da1f5ab0fb"
      ]
    ]
  },
  {
    "id": "1c11c86fbb36f097",
    "type": "http in",
    "z": "6ed1189c17ed0439",
    "name": "",
    "url": "/sensor",
    "method": "get",
    "upload": false,
    "swaggerDoc": "",
```

```
    "x": 190,
    "y": 900,
    "wires": [
      [
        "cdea8fe7bd7a2f5e"
      ]
    ]
  },
  {
    "id": "c49cd92e337f886b",
    "type": "debug",
    "z": "6ed1189c17ed0439",
    "name": "msg.payload",
    "active": true,
    "tosidebar": true,
    "console": false,
    "tostatus": false,
    "complete": "payload",
    "targetType": "msg",
    "statusVal": "",
    "statusType": "auto",
    "x": 750,
    "y": 940,
    "wires": []
  },
  {
    "id": "c7ddb56ba52e82df",
    "type": "debug",
    "z": "6ed1189c17ed0439",
    "name": "msg.payload",
    "active": true,
    "tosidebar": true,
```

```
    "console": false,
    "tostatus": false,
    "complete": "payload",
    "targetType": "msg",
    "statusVal": "",
    "statusType": "auto",
    "x": 690,
    "y": 760,
    "wires": []
},
{
    "id": "71be31afc89560dd",
    "type": "function",
    "z": "6ed1189c17ed0439",
    "name": "Distance 4",
    "func": "msg.payload = msg.payload.dist\nglobal.set('d',msg.payload)\nreturn msg;",
    "outputs": 1,
    "noerr": 0,
    "initialize": "",
    "finalize": "",
    "libs": [],
    "x": 470,
    "y": 980,
    "wires": [
        [
            "c49cd92e337f886b",
            "b88ea394cc4571c3"
        ]
    ]
},
{
    "id": "fbf611802e58a9d1",
```

```
    "type": "function",
    "z": "6ed1189c17ed0439",
    "name": "Distance 3",
    "func": "msg.payload = msg.payload.dist\nglobal.set('d',msg.payload)\nreturn msg;",
    "outputs": 1,
    "noerr": 0,
    "initialize": "",
    "finalize": "",
    "libs": [],
    "x": 470,
    "y": 780,
    "wires": [
      [
        "c7ddb56ba52e82df",
        "240d2e6c8f487fd8"
      ]
    ]
  },
  {
    "id": "231892da1f5ab0fb",
    "type": "function",
    "z": "6ed1189c17ed0439",
    "name": "LOAD cell 3",
    "func": "msg.payload =msg. payload.load\nglobal.set('l', msg.payload)\nreturn msg;",
    "outputs": 1,
    "noerr": 0,
    "initialize": "",
    "finalize": "",
    "libs": [],
    "x": 470,
    "y": 820,
    "wires": [
```

```
        [
            "e18c17929284e061"
        ]
    ]
},
{
    "id": "a0cbff62cdd2e77c",
    "type": "function",
    "z": "6ed1189c17ed0439",
    "name": "LOAD cell 4",
    "func": "msg.payload =msg. payload.load\nglobal.set('l', msg.payload)\nreturn msg;",
    "outputs": 1,
    "noerr": 0,
    "initialize": "",
    "finalize": "",
    "libs": [],
    "x": 470,
    "y": 1020,
    "wires": [
        [
            "c7a15e2a5bf9c2da"
        ]
    ]
},
{
    "id": "cdea8fe7bd7a2f5e",
    "type": "function",
    "z": "6ed1189c17ed0439",
    "name": "function 3",
    "func": "msg.payload = { \"dist\": global.get('d'), \"load\": global.get('l')}\nreturn msg;",
    "outputs": 1,
    "noerr": 0,
```

```
    "initialize": "",
    "finalize": "",
    "libs": [],
    "x": 420,
    "y": 900,
    "wires": [
        [
            "9bfb685be1503933"
        ]
    ]
},
{
    "id": "9bfb685be1503933",
    "type": "http response",
    "z": "6ed1189c17ed0439",
    "name": "",
    "statusCode": "",
    "headers": {},
    "x": 690,
    "y": 900,
    "wires": []
},
{
    "id": "240d2e6c8f487fd8",
    "type": "ui_gauge",
    "z": "6ed1189c17ed0439",
    "name": "",
    "group": "f3b64a4198b3c46c",
    "order": 15,
    "width": 4,
    "height": 4,
    "gtype": "gage",
```

```
    "title": "Distance 3",
    "label": "Cm",
    "format": "{{value}}",
    "min": 0,
    "max": "100",
    "colors": [
      "#00b500",
      "#e6e600",
      "#ca3838"
    ],
    "seg1": "",
    "seg2": "",
    "className": "",
    "x": 830,
    "y": 800,
    "wires": []
},
{
    "id": "b88ea394cc4571c3",
    "type": "ui_gauge",
    "z": "6ed1189c17ed0439",
    "name": "",
    "group": "f3b64a4198b3c46c",
    "order": 18,
    "width": 4,
    "height": 4,
    "gtype": "gage",
    "title": "Distance 4",
    "label": "Cm",
    "format": "{{value}}",
    "min": 0,
    "max": "100"
```

```
        "colors": [
            "#00b500",
            "#e6e600",
"#ca3838"
        ],
        "seg1": "",
        "seg2": "",
        "className": "",
        "x": 750,
        "y": 1000,
        "wires": []
    },
    {
        "id": "e18c17929284e061",
        "type": "ui_gauge",
        "z": "6ed1189c17ed0439",
        "name": "",
        "group": "f3b64a4198b3c46c",
        "order": 16,
        "width": 4,
        "height": 4,
        "gtype": "gage",
        "title": "LOAD CELL 3",
        "label": "KG",
        "format": "{{value}}",
        "min": 0,
        "max": "100",
        "colors": [
            "#00b500",
            "#e6e600",
            "#ca3838"
        ],
```

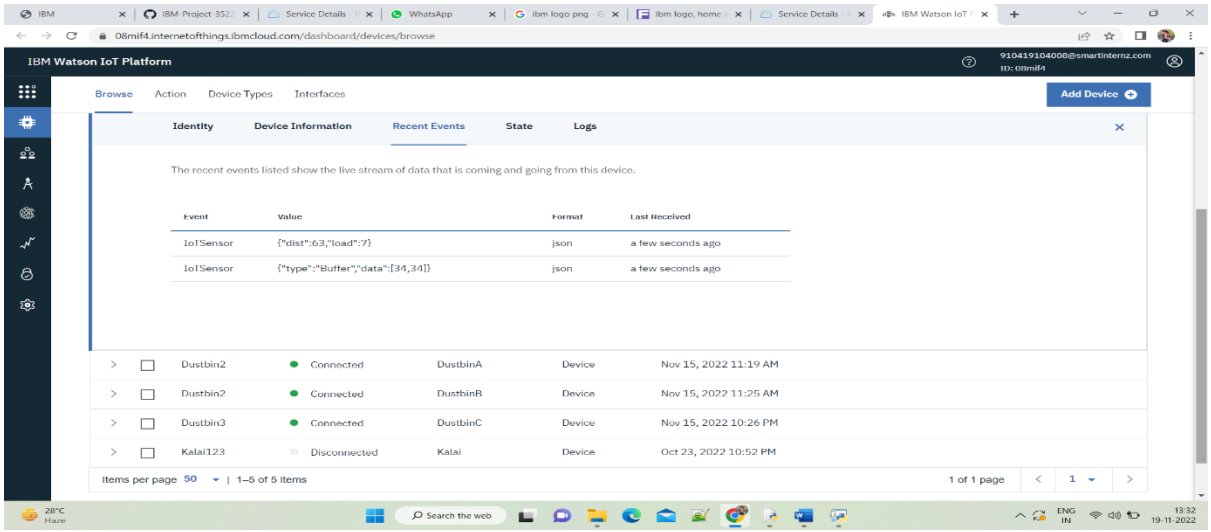
```
    "seg1": "",
    "seg2": "",
    "className": "",
    "x": 840,
    "y": 840,
    "wires": []
},
{
    "id": "147146a0342debce",
    "type": "ibmiot in",
    "z": "6ed1189c17ed0439",
    "authentication": "apiKey",
    "apiKey": "bf9996433728395e",
    "inputType": "evt",
    "logicalInterface": "",
    "ruleId": "",
    "deviceId": "Dustbin3",
    "applicationId": "",
    "deviceType": "DustbinC",
    "eventType": "+",
    "commandType": "",
    "format": "json",
    "name": "IBM IoT",
    "service": "registered",
    "allDevices": "",
    "allApplications": "",
    "allDeviceTypes": false,
    "allLogicalInterfaces": "",
    "allEvents": true,
    "allCommands": "",
    "allFormats": "",
    "qos": 0,
```

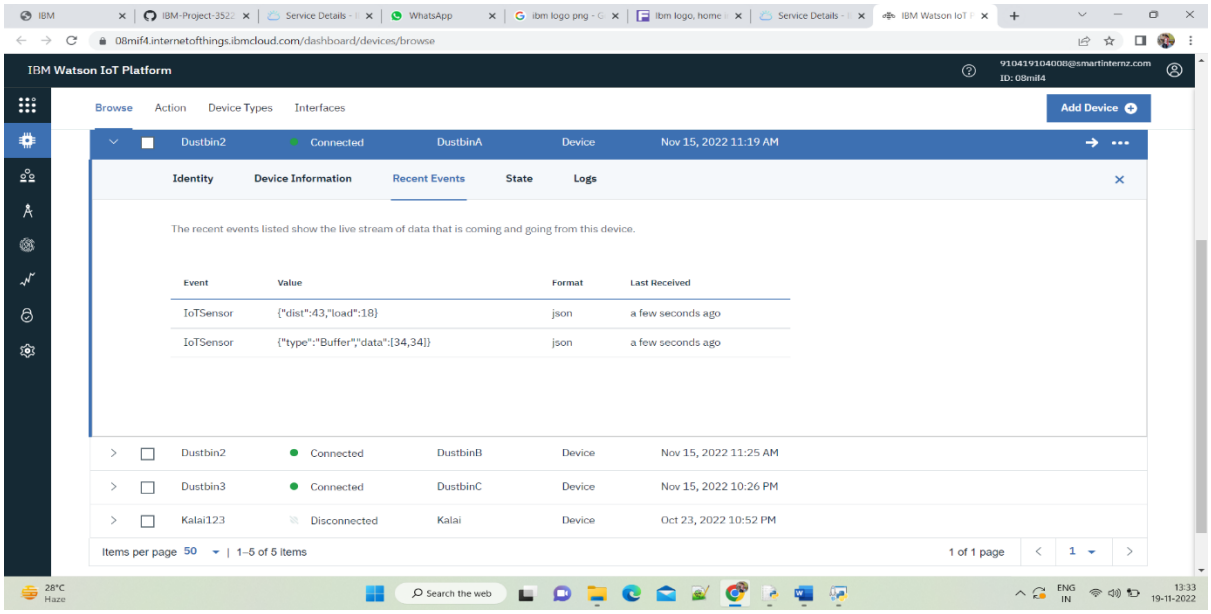
```
    "x": 230,
    "y": 1000,
    "wires": [
        [
            "71be31afc89560dd",
            "a0cbff62cdd2e77c",
            "c49cd92e337f886b"
        ]
    ]
},
{
    "id": "c7a15e2a5bf9c2da",
    "type": "ui_gauge",
    "z": "6ed1189c17ed0439",
    "name": "",
    "group": "f3b64a4198b3c46c",
    "order": 19,
    "width": 4,
    "height": 4,
    "gtype": "gage",
    "title": "LOAD CELL 4",
    "label": "KG",
    "format": "{{value}}",
    "min": 0,
    "max": "100",
    "colors": [
        "#00b500",
        "#e6e600",
        "#ca3838"
    ],
    "seg1": "",
    "seg2": "",
```

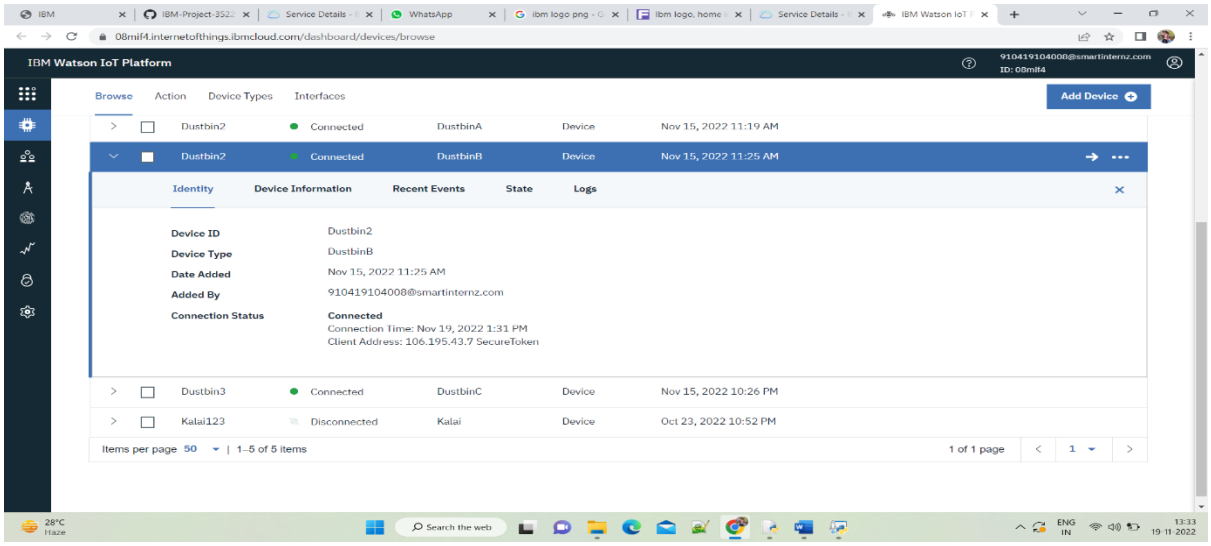
```
    "className": "",
    "x": 760,
    "y": 1040,
    "wires": []
},
{
    "id": "3cec67f2e3359287",
    "type": "http in",
    "z": "6ed1189c17ed0439",
    "name": "",
    "url": "/sensor",
    "method": "get",
    "upload": false,
    "swaggerDoc": "",
    "x": 230,
    "y": 1080,
    "wires": [
        [
            "c08f7bb853942b70"
        ]
    ]
},
{
    "id": "c08f7bb853942b70",
    "type": "function",
    "z": "6ed1189c17ed0439",
    "name": "function 4",
    "func": "msg.payload = { \"dist\": global.get('d'), \"load\": global.get('l')}\nreturn msg;",
    "outputs": 1,
    "noerr": 0,
    "initialize": "",
    "finalize": "",
```

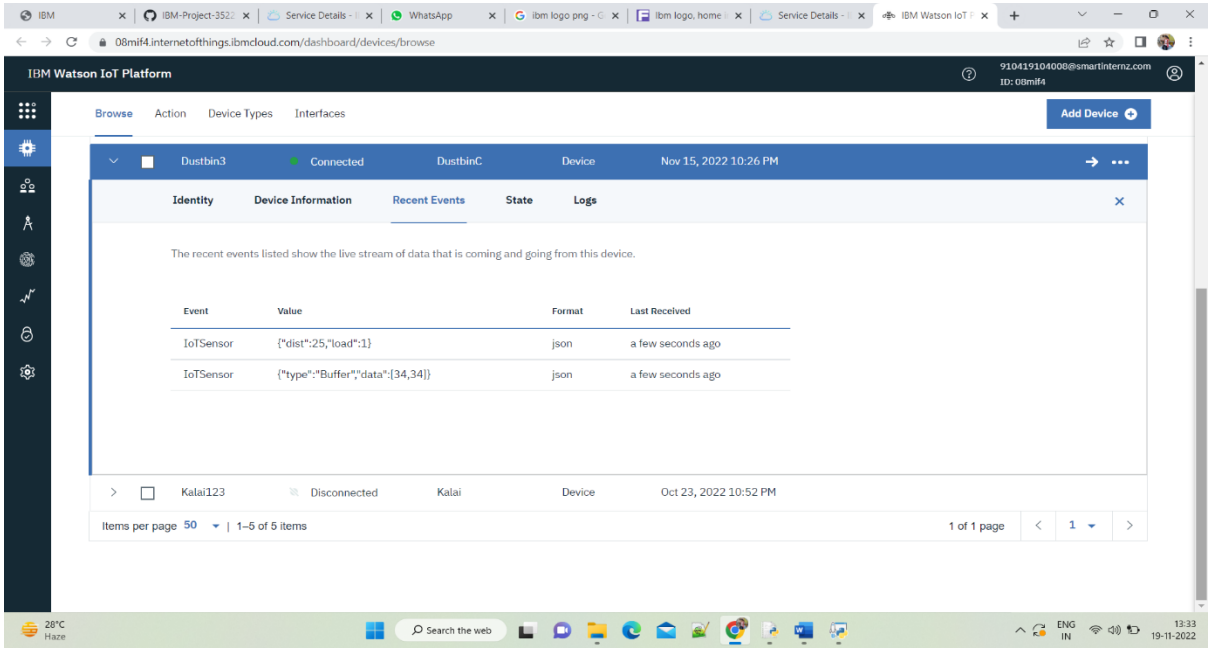
```
        "libs": [],
        "x": 460,
        "y": 1080,
        "wires": [
            [
                "07773f295c5a783c"
            ]
        ]
    },
    {
        "id": "07773f295c5a783c",
        "type": "http response",
        "z": "6ed1189c17ed0439",
        "name": "",
        "statusCode": "",
        "headers": {},
        "x": 670,
        "y": 1080,
        "wires": []
    }
]
```


OUTPUT IMAGE FOR SIMULATOR :

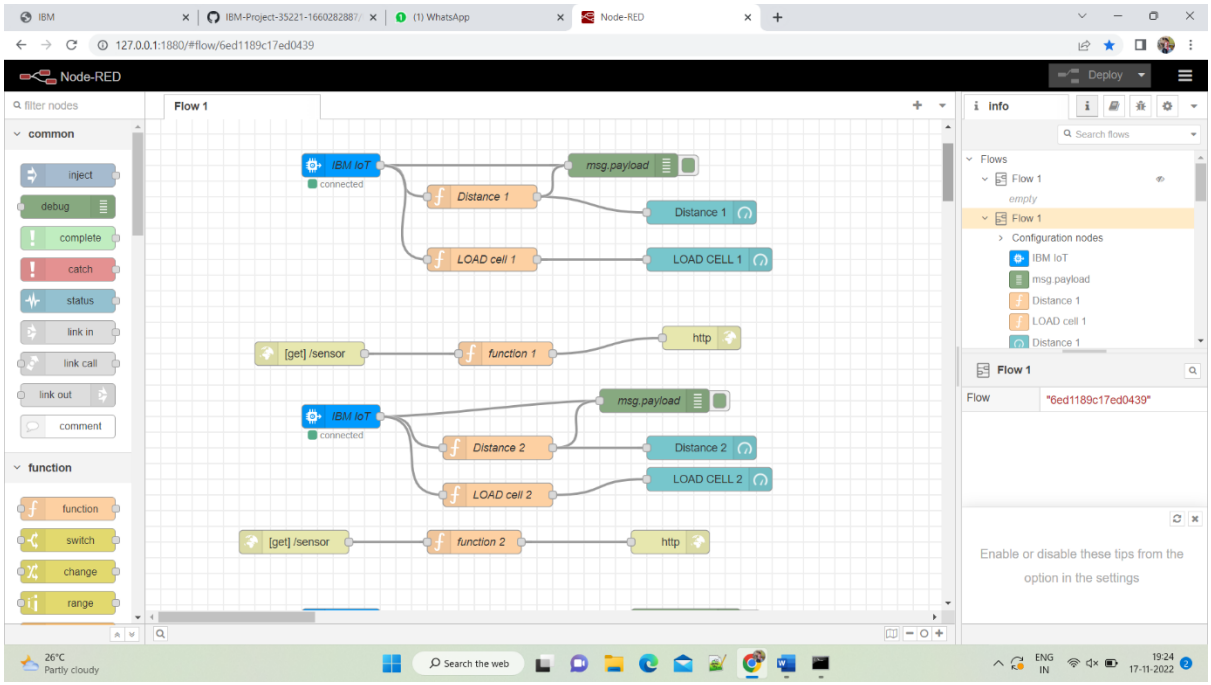


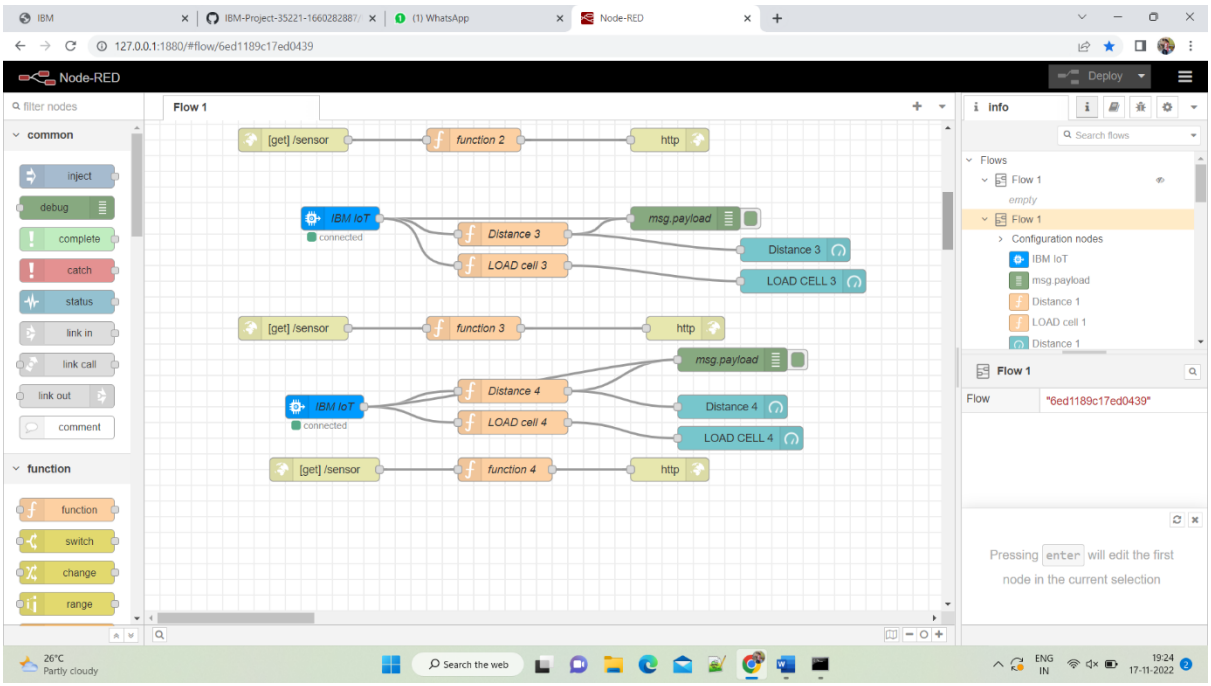

IBM Watson IoT Platform
Browse
Action
Device Types
Interfaces
Add Device
Dustbin2 Connected DustbinA Device Nov 15, 2022 11:19 AM
Identity
Device Information
Recent Events
State
Logs
The recent events listed show the live stream of data that is coming and going from this device.
Event
Value
Format
Last Received
IoTSensor
{"dist":43,"load":18}
json
a few seconds ago
IoTSensor
{"type":"Buffer","data":[34,34]}
json
a few seconds ago
Dustbin2 Connected DustbinB Device Nov 15, 2022 11:25 AM
Dustbin3 Connected DustbinC Device Nov 15, 2022 10:26 PM
Kalai123 Disconnected Kalai Device Oct 23, 2022 10:52 PM
Items per page 50
1–5 of 5 items
1 of 1 page


IBM Watson IoT Platform
Browse
Action
Device Types
Interfaces
Add Device
Dustbin2 Connected DustbinA Device Nov 15, 2022 11:19 AM
Dustbin2 Connected DustbinB Device Nov 15, 2022 11:25 AM
Identity
Device Information
Recent Events
State
Logs
Device ID
Dustbin2
Device Type
DustbinB
Date Added
Nov 15, 2022 11:25 AM
Added By
910419104008@smartinternz.com
Connection Status
Connected
Connection Time: Nov 19, 2022 1:31 PM
Client Address: 106.195.43.7 SecureToken
Dustbin3 Connected DustbinC Device Nov 15, 2022 10:26 PM
Kalai123 Disconnected Kalai Device Oct 23, 2022 10:52 PM
Items per page 50
1–5 of 5 items
1 of 1 page

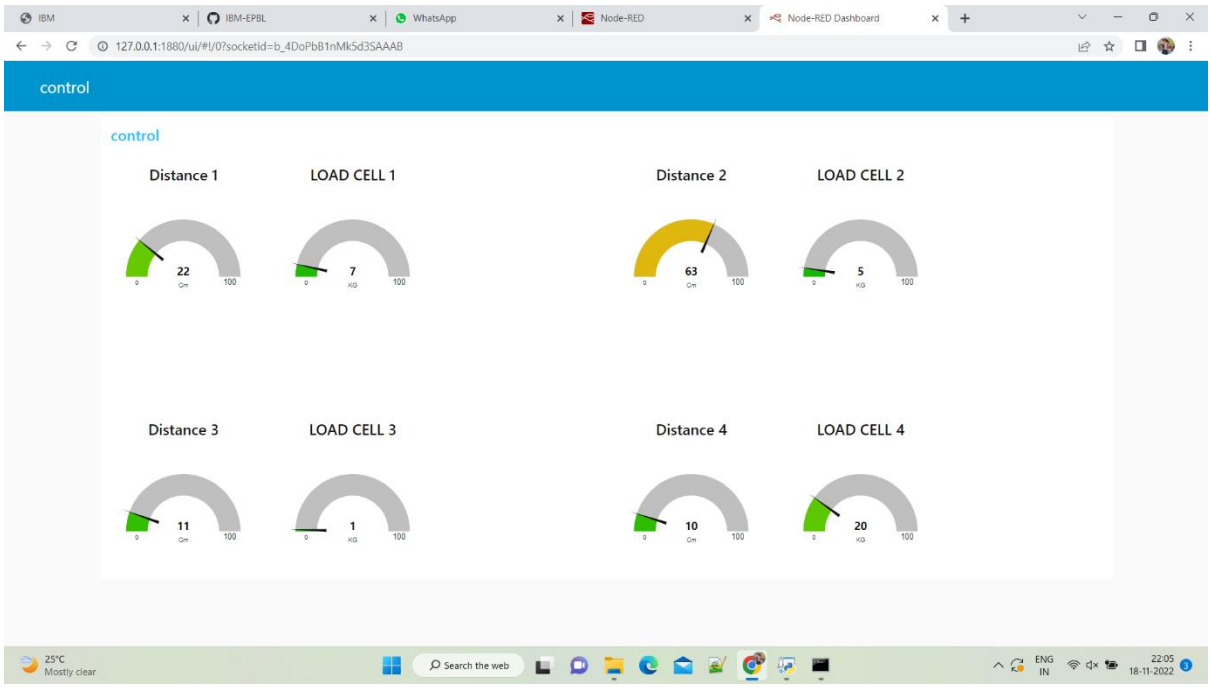

IBM Watson IoT Platform
Browse
Action
Device Types
Interfaces
Add Device
Dustbin3 Connected DustbinC Device Nov 15, 2022 10:26 PM
Identity
Device Information
Recent Events
State
Logs
The recent events listed show the live stream of data that is coming and going from this device.
Event
Value
Format
Last Received
IoTSensor
{"dist":25,"load":1}
json
a few seconds ago
IoTSensor
{"type":"Buffer","data":[34,34]}
json
a few seconds ago
Kalai123 Disconnected Kalai Device Oct 23, 2022 10:52 PM
Items per page 50
1–5 of 5 items
1 of 1 page

NODE-RED CONNECTIONS:





LOAD CELL:



13.3 LINKS

GitHub Link:

https://github.com/IBM-EPBL/IBM-Project-3714-1658592650

Demo video Link:

https://youtu.be/zagtadAZDG0