SMART WASTE MANAGEMENT SYSTEM FOR METROPOLITAN CITIES

Category: INTERNET OF THINGS

A PROJECT REPORT

Submitted by

SANTHOSHSHIVAN K

RAGHUL R

MYSTICA C

SHANMUGAKAMALESH M

FROM

UNIVERSITY COLLEGE OF ENGINEERING KANCHEEPURAM

In fulfillment of project in IBM-NALAYATHIRAN 2022

Team Id: PNT2022TMID40462

PROJECT GUIDES

Industry Mentor: Mr. Dinesh

Faculty Mentor: Mrs. R. Vijayarani

INDEX

1. INTRODUCTION

1.1 Project Overview:

Smart waste management System

Smart waste management solutions use sensors placed in waste receptacles to measure fill levels and to notify city collection services when bins are ready to be emptied. Over time, historical data collected by sensors can be used to identify fill patterns, optimize driver routes and schedules, and reduce operational costs. The cost of these sensors is steadily decreasing, making IoT waste bins more feasible to implement and more attractive to city leaders.

1.2 Purpose:

Smart bins are the newest type of waste management technology that helps cities reduce waste effectively. A smart waste management system allows sanitation workers and employees to obtain real-time data, which helps prevent the bins from overflowing, contributing to healthier and smarter cities.

- Efficient waste collection in a short span of time with less labor,
- To minimize the harmful effects of waste on the environment,
- A cleaner and more organized city plan,
- Reducing the irregularly placed and excessive number of containers to help to improve the environmental image, and reduce noise pollution, and traffic density during collection by collecting waste at longer intervals.
- In the present scenario, we see the garbage bins being overloaded and all the garbage spilling out resulting in pollution. The detection, monitoring, and management of waste are one of the primary problems of the present era.
- People throw garbage in the dustbin which is already overflowed. Sometimes due to unclean garbage bins, bad smell arises also toxic and unhygienic gases are produced which is a way to

support Air pollution and other harmful diseases which are easily spreadable. It is the very bad look of the city.

2. LITERATURE SURVEY

2.1 Existing Problem:

In the present scenario, we see the garbage bins being overloaded and all the garbage spills outs resulting in pollution. The detection, monitoring, and management of waste are one of the primary problems of the present era.

A big challenge in urban cities is solid waste management. The garbage collecting authority in traditional waste management systems doesn't know about the level of garbage in dustbins the dust bins get full of garbage, then it gets overflowed as well as spelled out from the dustbin leading to unhygienic conditions in cities.

People throw garbage on that dustbin which is already overflowed. Sometimes due to unclean garbage bins, bad smell arises also toxic & unhygienic gases are produced which is a way to support Air pollution and other harmful diseases which are easily spreadable. It is very a bad look at the city. The use of a traditional system results in an efficient and time and money-spending System.

Hence, we design a system based on an IoT application for collecting garbage from particular areas whose garbage bins are overflowing with prior concerns. This method is advanced in which garbage management is automated. This project Garbage Monitoring system using IoT is a very innovative system that will help to keep the cities clean.

2.2 References:

1. K.S. Adu-Manu, C. Tapparello, W. Heinzelman, F.A. Katsriku, J.-D. Abdulai

A Smart IoT System for Waste Management: Without well management, the waste containers may be overflowed or give off unpleasant smell, which affect the public health. (2018).

2.

IoT based Waste collection Management System for Smart cities: In the recent decades, Urbanization has been a pivotal issue tremendously. In the meantime, there is an expansion in waste creation.(2019)

3.

IoT based Smart waste Management System: "Indian prospective": In present scenario corporations in India doesn't get real time information about the dustbins.(2019)

4.

Waste Management Improvement in cities using IoT: Implementing waste management the major challenge is the management of waste in cost optimal. And the containers everyday which is a complicates. (2020)

2.3 Problem Statement Definition:

Design a system based on IoT application for collecting garbage from particular area whose garbage bins are overflowing with prior concern. This method is advanced in which garbage management is automated. This project Garbage Monitoring system using IOT is a very innovative system which will help to keep the cities clean.

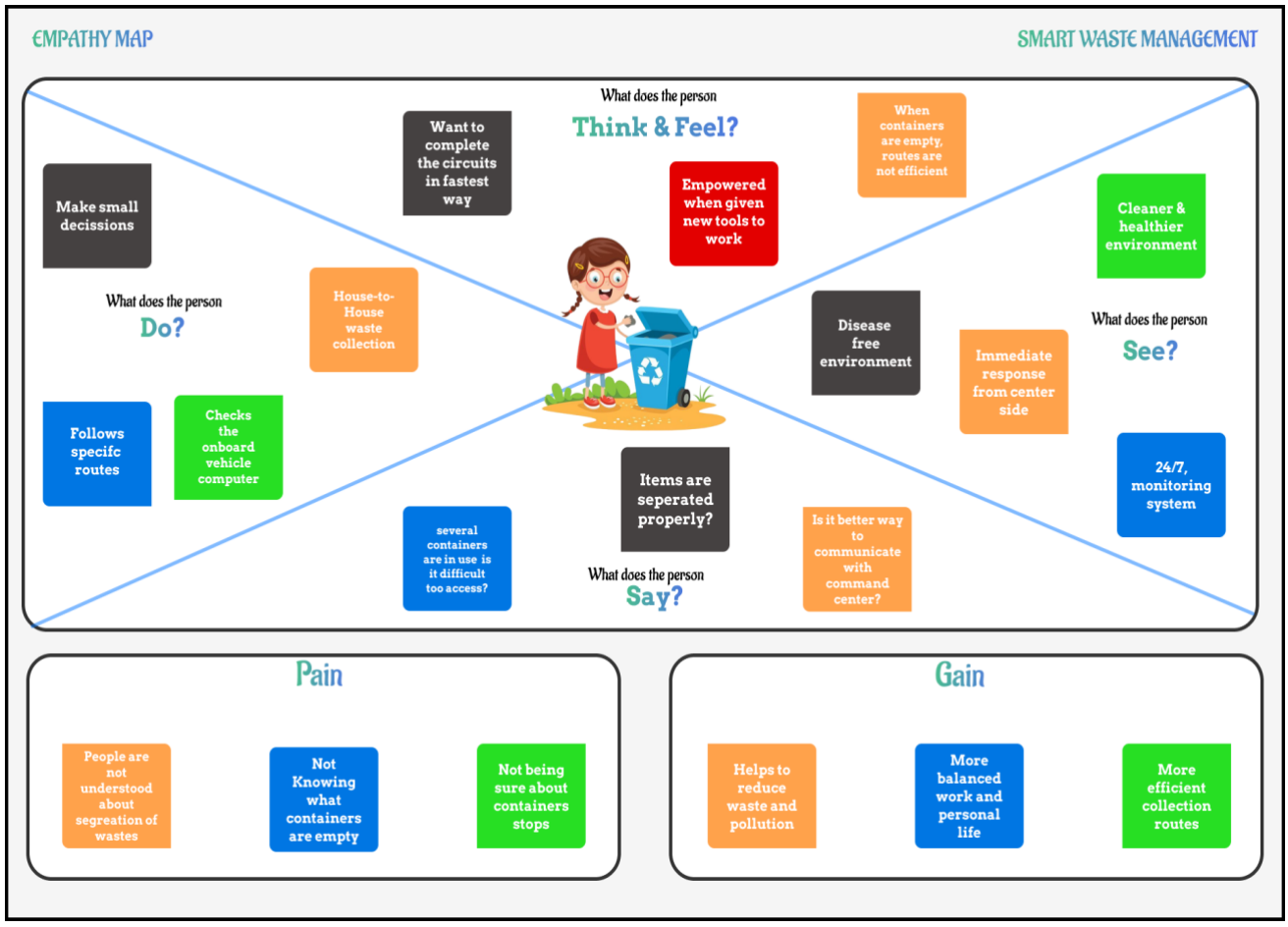
A growing population and economy, which means increased volumes of waste generated. This puts pressure on waste management facilities, which are already in short supply.

A policy and regulatory environment that does not actively promote the waste management hierarchy. This has limited the economic potential of the waste management sector, which has an estimated turnover of approximately R10 billion per annum[11]. Both waste collection and the recycling industry make meaningful contributions to job creation and GDP, and they can expand further.

IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas:

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes. It is a useful tool to helps teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.

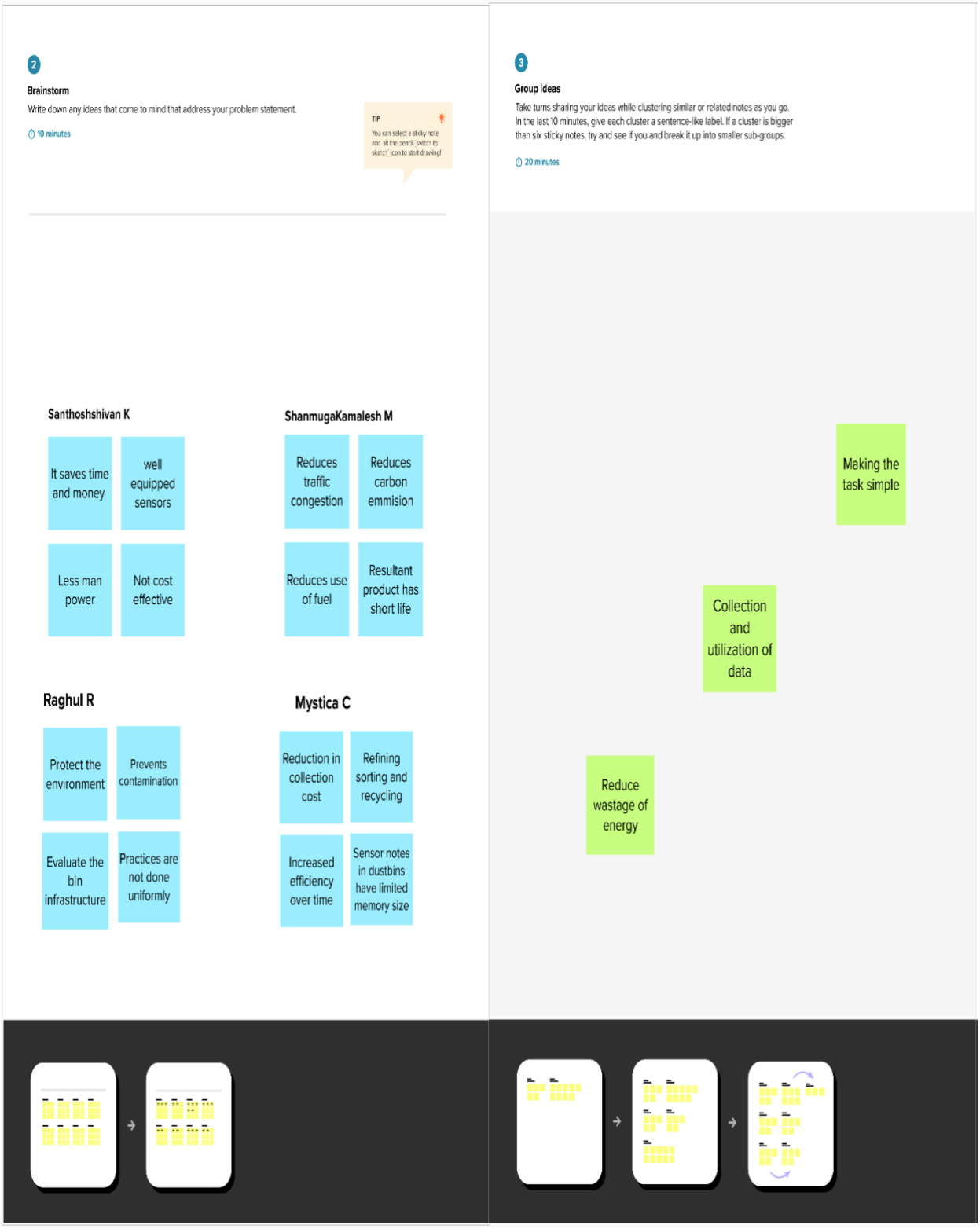


Reference: ClickHere.

3.2 Ideation & Brainstorming:

Ideation and Brainstorming Ideation is often closely related to the practice of brainstorming, a **specific technique that is utilized to generate new ideas**. A principal difference between ideation and brainstorming is that ideation is commonly more thought of as being an individual pursuit, while brainstorming is almost always a group activity.

Reference: ClickHere

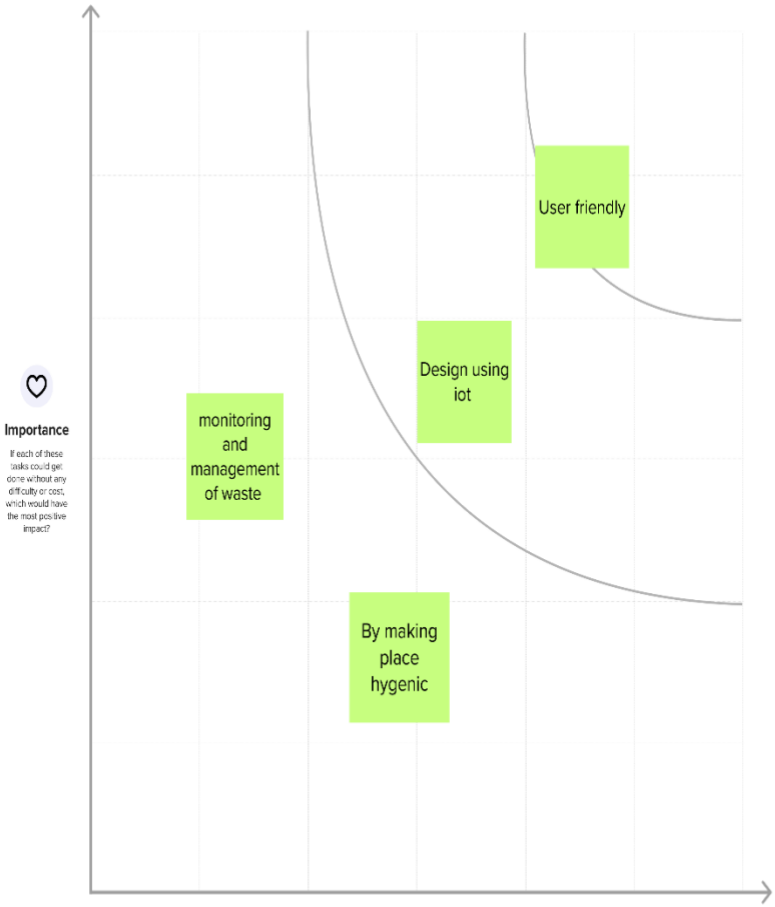

2
Brainstorm
Write down any ideas that come to mind that address your problem statement.
10 minutes
TIP
Santhoshshivan K
It saves time and money
well equipped sensors
Less man power
Not cost effective
ShanmugaKamalesh M
Reduces traffic congestion
Reduces carbon emmision
Reduces use of fuel
Resultant product has short life
Raghul R
Protect the environment
Prevents contamination
Evaluate the bin infrastructure
Practices are not done uniformly
Mystica C
Reduction in collection cost
Refining sorting and recycling
Increased efficiency over time
Sensor notes in dustbins have limited memory size
3
Group ideas
Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.
20 minutes
Making the task simple
Collection and utilization of data
Reduce wastage of energy

4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes



After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons

A **Share the mural**
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.

B **Export the mural**
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

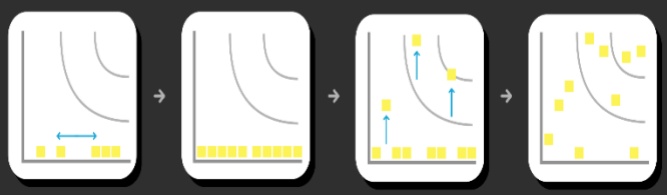
Keep moving forward

Strategy blueprint
Define the components of a new idea or strategy.
Open the template →

Customer experience journey map
Understand customer needs, motivations, and obstacles for an experience.
Open the template →

Strengths, weaknesses, opportunities & threats
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.
Open the template →

Share template feedback

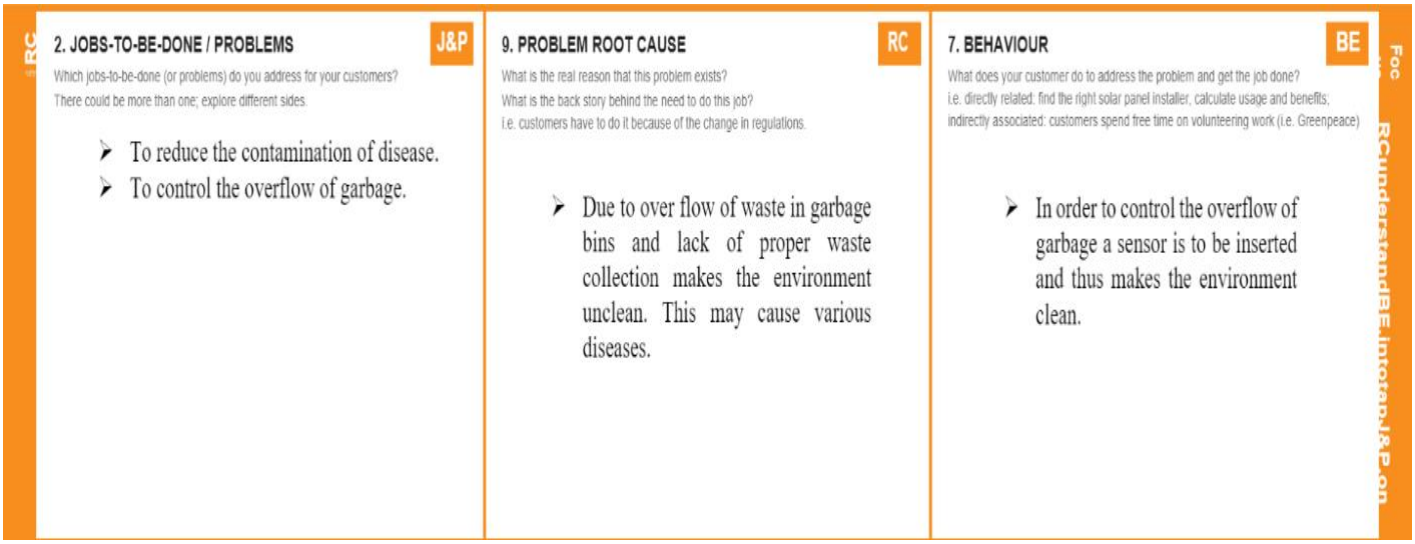


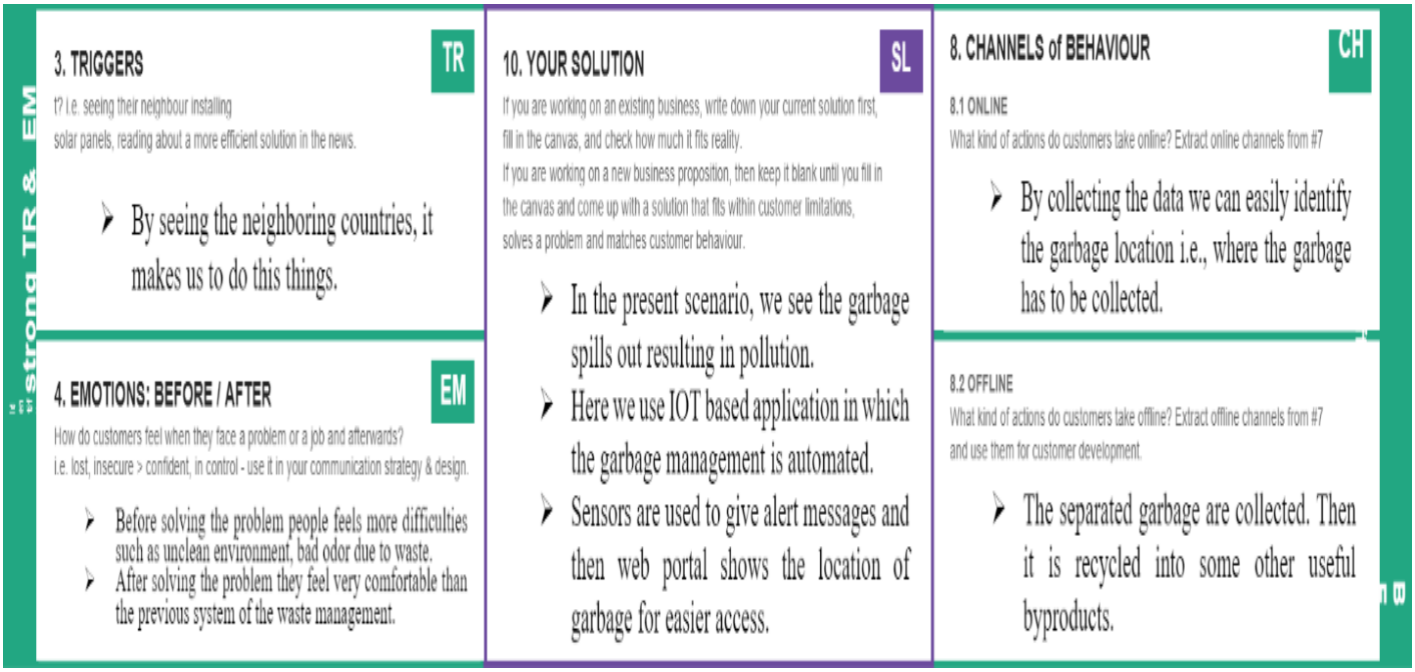
3.3 Proposed Solution:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	How can the trash bins in metropolitan cities can be easily notified and cleaned in an efficient manner?
2.	Idea / Solution description	The fill level and weight of garbage in Smart bins are observed using a web page and the information is collected using sensors. If the garbage in the smart bin gets filled then the sensors in the smart bins sends an information alert to GPS location of those smart bins to the respective person.
3.	Novelty / Uniqueness	❖ Smart waste bins classifies recyclables into separate compartments. ❖ This can lower waste management costs as much 80% and drastically improve employee efficiency.
4.	Social Impact / Customer Satisfaction	✓ It reduces CO2 emission since the usage of fuel is reduced. ✓ Prevents contamination due to overflowing of garbage. ✓ Maintain clean environment.

5.	Business Model (Revenue Model)	Offering software as a service model to the respective municipality and creating revenue from it.
6.	Scalability of the Solution	Finally the filled smart bins can be easily managed from small towns to large metropolitan cities.

3.4 PROBLEM SOLUTION FIT:





Reference: ClickHere.

4 REQUIREMENT ANALYSIS

4.1 Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	The municipality have to register by creating id and password through webpage by using G-mail.
FR-2	User Confirmation	User can login through their ID and Password and the status can be monitored.
FR-3	Smart Waste bins	If waste in the bins reaches 80%, then sensors in the bin gives an alert to the user and the user can monitor periodically so that we can collect some data.
FR-4	GPS	GPS is used to trace the actual location of the filled smart bin and helps to collect the filled smart bins from the particular location.
FR-5	Cloud	The data which are collected is stored under the IBM cloud.

4.2 Non-functional Requirements:

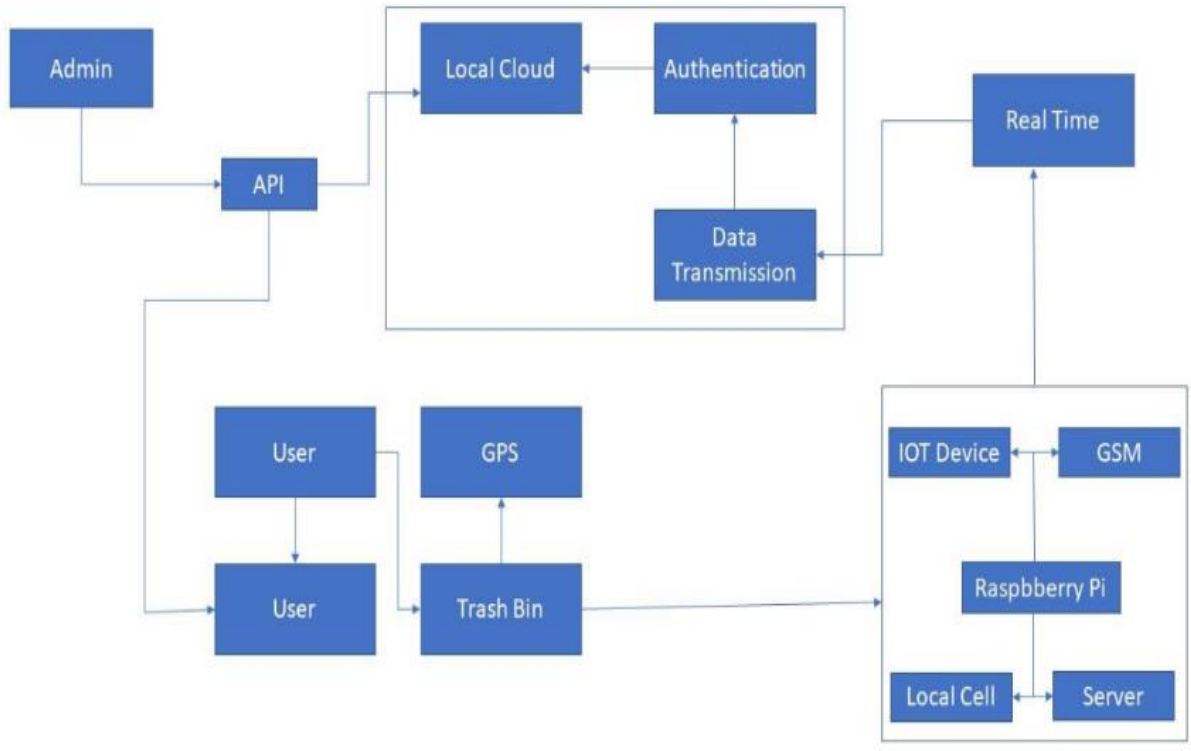
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The portal can beeasily accessible and monitored. This is an user-friendly process.
NFR-2	**Security**	Data are more secured and only the user can login to the webpage. No other person can access the data expect the user.
NFR-3	**Reliability**	Smart waste management provides better working conditions for drivers and waste collectors. They spend their time more efficiently by this technique.
NFR-4	**Performance**	User are provided with decision-making and optimization of waste collection routes and vehicle loads resulting in at least 30% of route reduction.
NFR-5	**Availability**	The system is user-friendly. Failure percentage is much less and 24*7 support is available.
NFR-6	**Scalability**	Finally the filled smart bins can be easily managed from small towns to large metropolitan cities.

5 PROJECT DESIGN

5.1 Data Flow Diagrams:

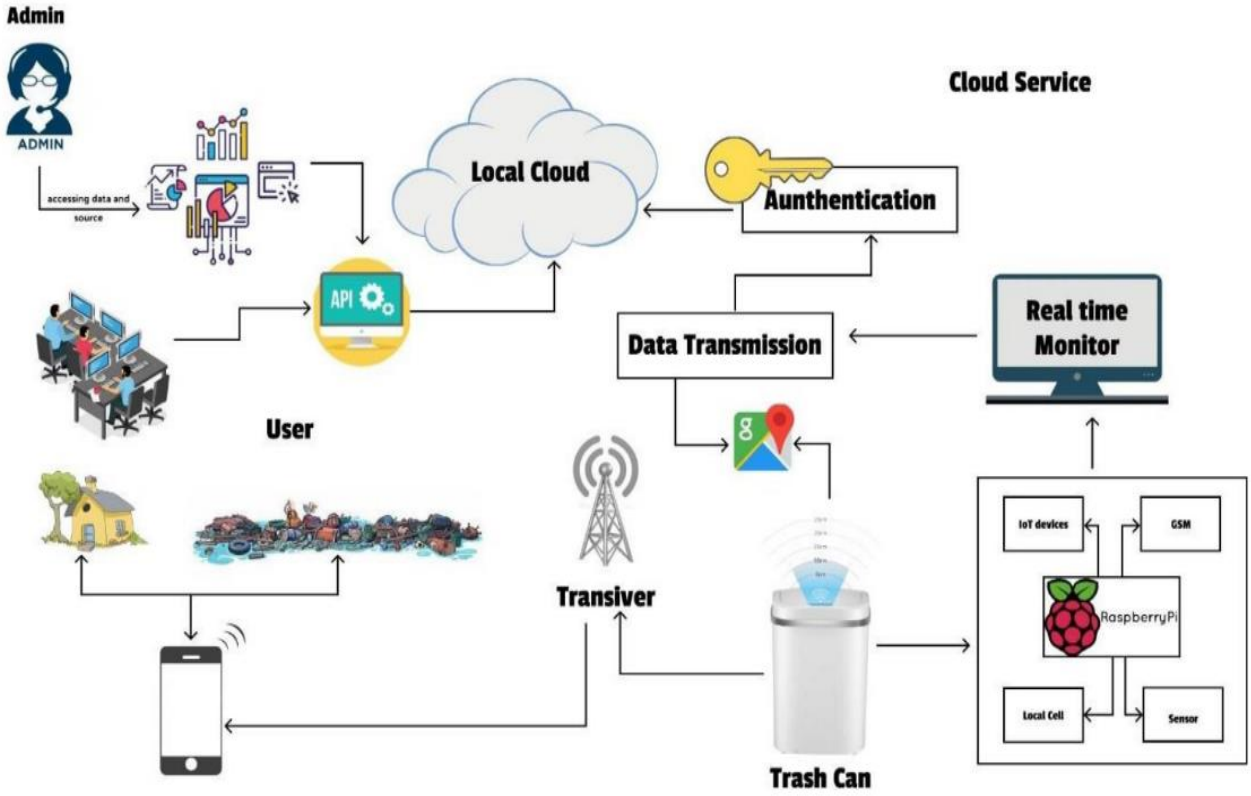
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



5.2 SOLUTION AND TECHNICAL ARCHITECTURE

1. Summary:

The smart bins are constructed based on the sensor application and raspberry pi . It can also act as a transceiver since it is connected to the mobile phone of the user . The overall process of the sensors and raspberry pi is monitored using real time monitor which can help data transmission.This is stored and formulated using cloud data . Through which the admin can access the data and then track the location from GPS .



2. Components & Technologies:

S.No	Component	Description	Technology
1.	Raspberry pi controller	The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega 328P microcontroller.	Arduino programming itself is done in python.
2.	Application Logic-1	Logic for Ultrasonic sensor data	Python
3.	GPRS	To track the location of the bin	Python
4.	IoT	To collect the data and alert the users	IBM Watson IoT Platform, Node Red.
5.	Cloud Database	Stores the collected data in cloud	Cloudant DB

3. Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open Source Microcontroller	Arduino UNO is used to make the IoT device.	Python
2.	Security	Encryption/Decryption used for security purpose.	Surface Mount Sensor
3.	Scalable Architecture	New features can be added	Node Red
4.	Availability	Web application can be accessed from anywhere	IBM Watson IoT Platform, NodeRed
5.	Performance	All truck drivers can access the application at sametime.	Cloudant DB, IBM Watson IoT Platform.

5.3 User Stories

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Admin (Web manage web server)	Login	USN-1	As a admin, I gave the user Id and Password for every worker and manage them.	I can manage web accountant/ dashboard	Medium	Sprint-1
Co-admin	Login	USN-2	As a Co-admin, I will manage the garbage level and monito the smart bins. If the garbage gets filled, the alert notification will be received and I'll post location and garbage id to the truck and will go and collect the trash	I can manage the smart bins and garbage level.	High	Sprint-2
Truck driver	Login	USN-3	As a truck driver, I'll follow the route send by Co-admin to reach the particular garbage through its id.	I can drive to reach the filled garbage in shortest route given	Medium	Sprint-3
Total Garbage	Login	USN-4	As a waste collector, I'll collect all the waste from the smart bins and load them into the truck and send them to landfill.	I can collect trash and load them into the truck and send off	Medium	Sprint-4
Municipality	Login	USN-5	As a municipality, I will check the process of waste collecting and sending them in a disciplined manner without any issues	I can manage all these process and supervise it	High	Sprint-5

6. PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING & SCHEDULING:

TITLE	DESCRIPTION	DATE
Literature Survey & Information Gathering	Literature survey on the selected project is done by gathering information about related details on technical papers and web browsing.	28 SEPTEMBER 2022
Prepare Empathy Map	Prepared Empathy Map Canvas to capture the user Pains & Gains list of problem statements.	24 SEPTEMBER 2022
Ideation	List the organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility and importance.	25 SEPTEMBER 2022
Proposed Solution	Prepared the proposed solution document, which includes the novelty, feasibility of idea business model, socia impact, scalability of solution, etc.	23 SEPTEMBER 2022
Problem Solution Fit	Prepared problem - solution fit document.	30 SEPTEMBER 2022
Solution Architecture	Prepare solution architecture document.	28 SEPTEMBER 2022

6.2 SPRINT DELIVERY SCHEDULE

Product Backlog, Sprint Schedule, and Estimation

Use the below template to create product backlog and sprint schedule.

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, we must register for an IBM cloud account , IoT platform ,RED node service and cloudant DB.	2	High	Mystica,Raghul
Sprint-2		USN-2	As a user, we develop a python script to publish random sensor data.	2	High	Santhoshshivan,Shanmugakamalesh
Sprint-3		USN-3	As a user send the sensor data to IBM Watson IoT platform,install palette for nodes available in Node red.	2	Medium	Santhoshshivan,Shanmugakamalesh Mystica,Raghul
Sprint-4	Dashboard	USN-4	Configure the Node-RED flow to receive data from the IBM IoT platform and a web UI should be created in Node-RED	1	High	Santhoshshivan,Shanmugakamalesh Mystica,Raghul

Project Tracker, Velocity & Burndown Charts

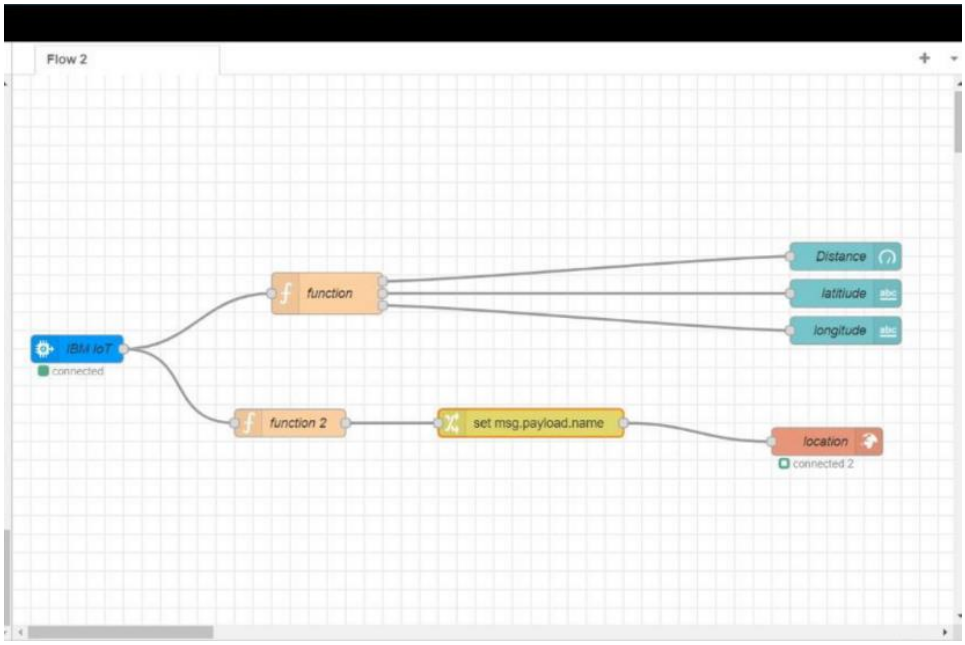
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed(as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	30	30 Oct 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	49	04 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	50	09 Nov 2022

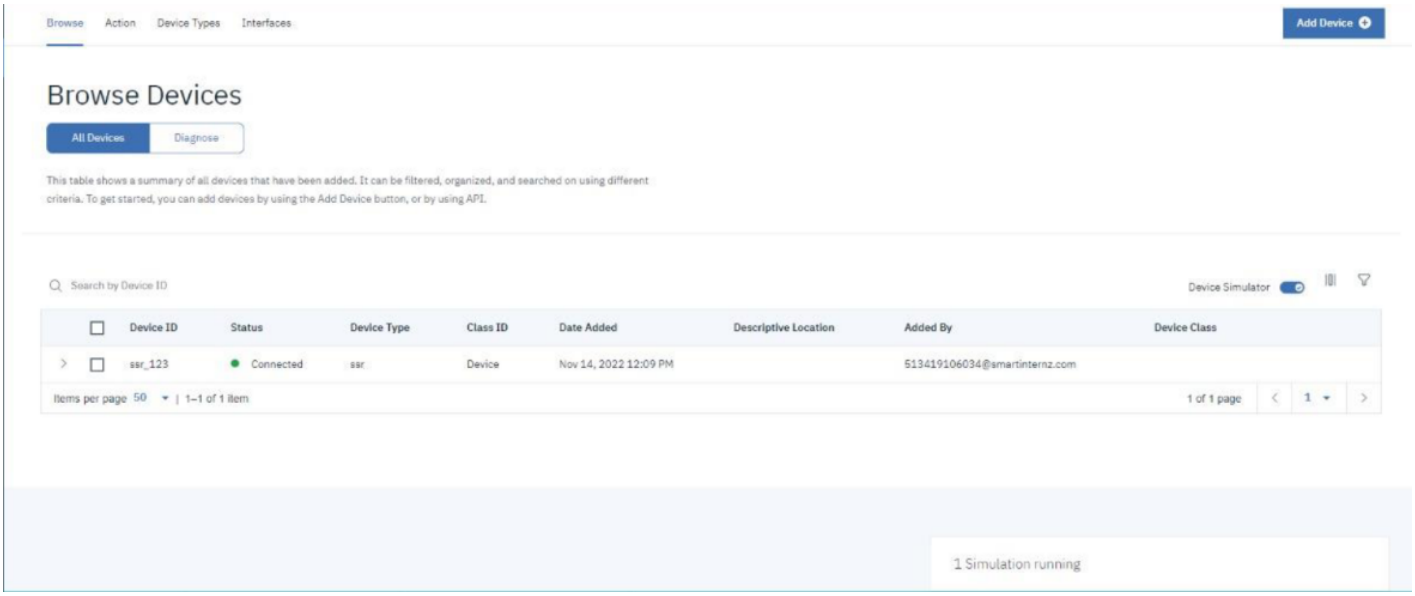
Velocity:

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

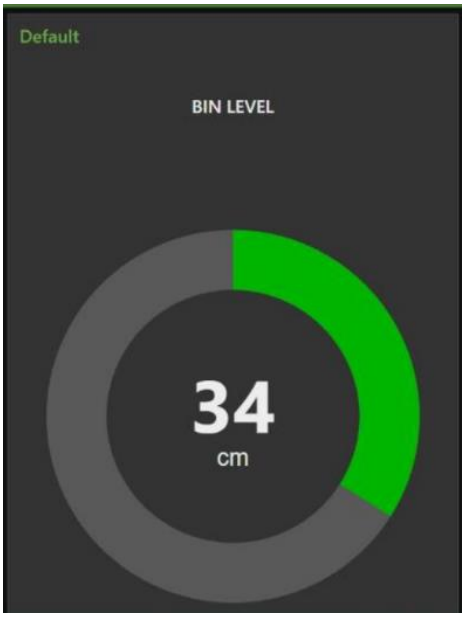
7. CODING AND SOLUTIONING

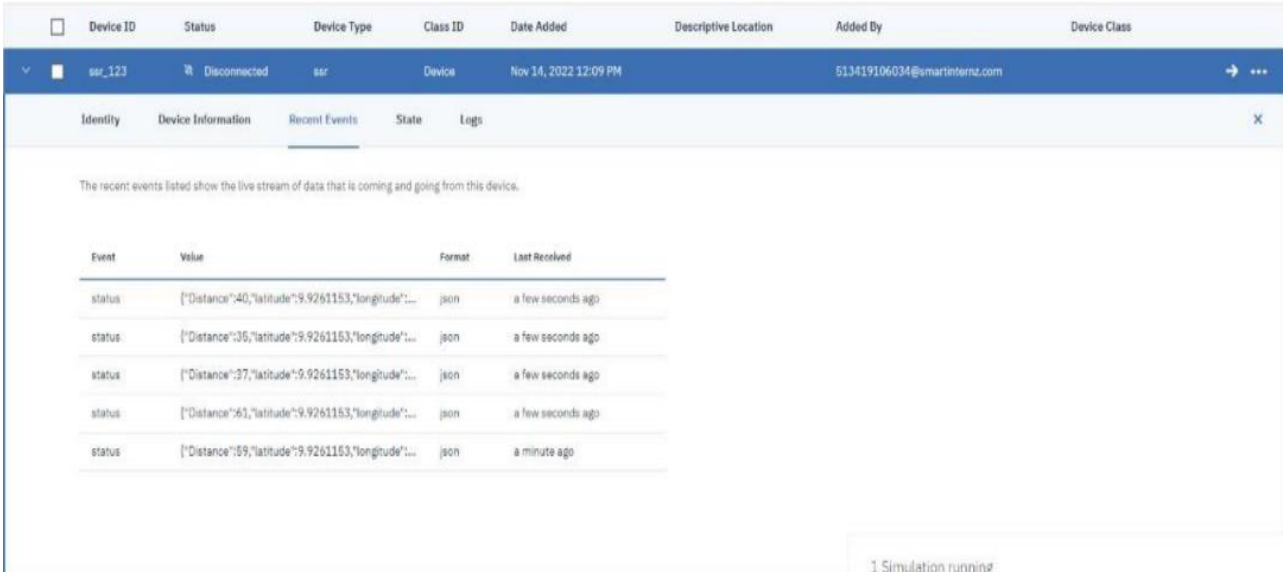
NODE RED SERVICE ASSOCIATED WITH IBM CLOUD:





Node red Dashboard:





8. ADVANTAGES AND DISADVANTAGES

ADVANTAGES:

- ✓ Meet the increasing demand for sustainable solutions
- ✓ Transportation Management System and reduce the use of fuel
- ✓ Intelligent Route Optimization.
- ✓ Customer Service System.
- ✓ Customer Self-Service App.
- ✓ Clean and healthy environment
- ✓ Automated Workflows.

DISADVANTAGES:

- ✓ This results into high initial cost due to expensive smart dustbins compare to other methods.
- ✓ Sensor nodes used in the dustbins have limited memory size.
- ✓ Wireless technologies used in the system such as zigbee and wifi have shorter range and lower data speed.

9. CONCLUSION

We presented an intelligent waste collection system. The system is based on IoT sensors. It is responsible for measuring the waste level in the smart bins. When the smart bins gets filled almost there will be information received by the admin, Since the admin can access the data and location of the bin. Later send this data (through Internet) to a server for storage and processing. This data helps to compute the optimized collection routes for the workers. In future, we would like to enhance the system for different kind of wastes, namely solid and liquid wastes.

10. FUTURE SCOPE

The advantage of this work is its contribution in making a Smart city. Among the many challenges that a city faces, waste management is of utmost importance. This is because, it is directly related to health of people living in the area. We are further extending this work to address problems of seggragating different kind of wastes (e.g., solid, liquid etc.), and identifying different vehicles for collecting it. The optimization algorithms may be devised accordingly depending on the requirements. In future, we would like to enhance the system for different kind of wastes, namely solid and liquid wastes.

11.APPENDIX

11.1 SOURCE CODE:

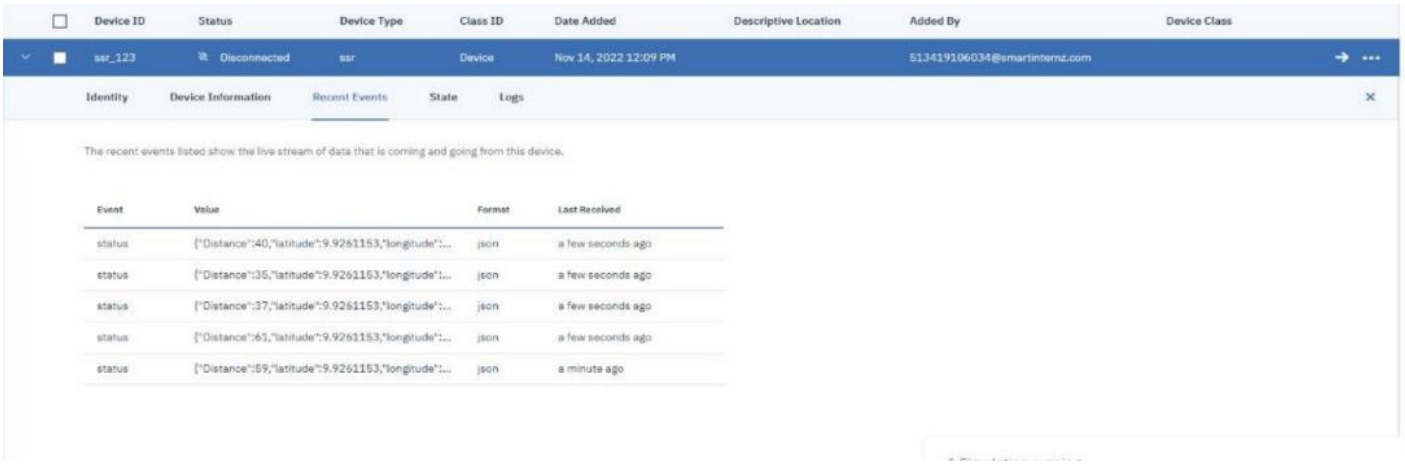
PYTHON CODE TO PUBLISH DATA

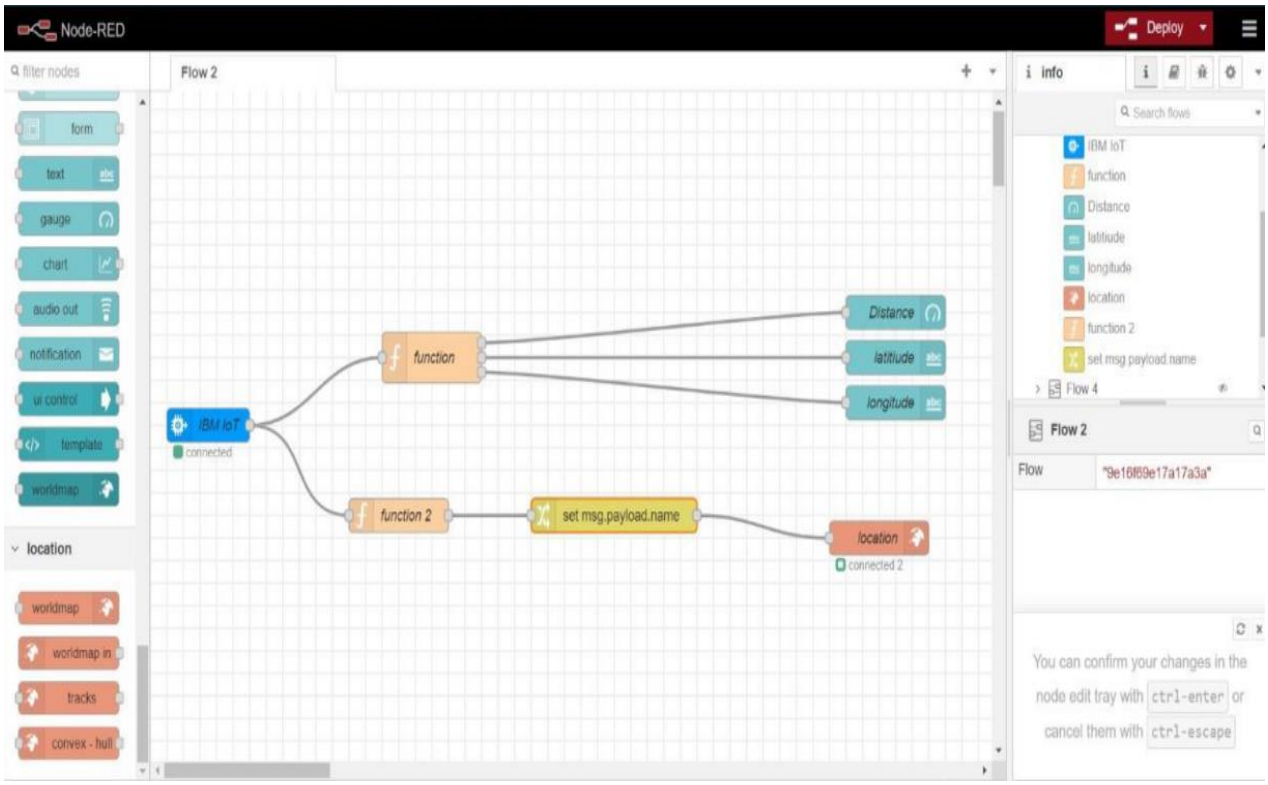
```
#IBM Watson IOT Platform
#pip install wiotp-sdk from geopy.geocoders import Nominatim
import wiotp.sdk.device
import time
import random
myConfig = {
        "identity": {
          "orgId": "n7xtmx",
          "typeId": "ssr",
          "deviceId":"ssr_123"
 },
 "auth": {
          "token": "12345678"
        }
 }
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:
           dist1=random.randint(0,100)
          loc = Nominatim(user_agent="GetLoc")
          getLoc1 = loc.geocode("madurai")
```

```
  lat=getLoc1.latitude
  log=getLoc1.longitude
  f="alert"
  if dist1>=80:
    a=print('The bin1 level is high')
    type(a)
    myData={'name':f,'Distance':dist1,'latitude':lat,'longitude':log}
else:
  a=print('The bin1 level is low')
  type(a)
  myData={'Distance':dist1,'latitude':lat,'longitude':log}
  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()
```

OUTPUT


Python 3.7.9 Shell
File Edit Shell Debug Options Window Help
========== RESTART: C:\Users\USER\Documents\ibm\ibm iot.py ==========
2022-11-18 15:48:23,339 wiotp.sdk.device.client.DeviceClient INFO Connected successfully: d:n7xtmx:ssr:ssr_123
The bin1 level is high
Published data Successfully: %s {'name': 'alert', 'Distance': 95, 'latitude': 9.9261153, 'longitude': 78.1140983}
The bin1 level is high
Published data Successfully: %s {'name': 'alert', 'Distance': 83, 'latitude': 9.9261153, 'longitude': 78.1140983}
The bin1 level is low
Published data Successfully: %s {'Distance': 71, 'latitude': 9.9261153, 'longitude': 78.1140983}
The bin1 level is low
Published data Successfully: %s {'Distance': 75, 'latitude': 9.9261153, 'longitude': 78.1140983}
The bin1 level is high
Published data Successfully: %s {'name': 'alert', 'Distance': 85, 'latitude': 9.9261153, 'longitude': 78.1140983}
The bin1 level is high
Published data Successfully: %s {'name': 'alert', 'Distance': 99, 'latitude': 9.9261153, 'longitude': 78.1140983}
The bin1 level is low
Published data Successfully: %s {'Distance': 60, 'latitude': 9.9261153, 'longitude': 78.1140983}
The bin1 level is high
Published data Successfully: %s {'name': 'alert', 'Distance': 97, 'latitude': 9.9261153, 'longitude': 78.1140983}
The bin1 level is low
Published data Successfully: %s {'Distance': 31, 'latitude': 9.9261153, 'longitude': 78.1140983}
The bin1 level is low
Published data Successfully: %s {'Distance': 6, 'latitude': 9.9261153, 'longitude': 78.1140983}
The bin1 level is low
Published data Successfully: %s {'Distance': 53, 'latitude': 9.9261153, 'longitude': 78.1140983}
The bin1 level is high
Published data Successfully: %s {'name': 'alert', 'Distance': 83, 'latitude': 9.9261153, 'longitude': 78.1140983}
The bin1 level is low
Published data Successfully: %s {'Distance': 37, 'latitude': 9.9261153, 'longitude': 78.1140983}
The bin1 level is low
Published data Successfully: %s {'Distance': 78, 'latitude': 9.9261153, 'longitude': 78.1140983}
The bin1 level is low
Published data Successfully: %s {'Distance': 45, 'latitude': 9.9261153, 'longitude': 78.1140983}
The bin1 level is low
Published data Successfully: %s {'Distance': 50, 'latitude': 9.9261153, 'longitude': 78.1140983}
The bin1 level is low
Published data Successfully: %s {'Distance': 11, 'latitude': 9.9261153, 'longitude': 78.1140983}
The bin1 level is low
Published data Successfully: %s {'Distance': 27, 'latitude': 9.9261153, 'longitude': 78.1140983}
The bin1 level is low
Published data Successfully: %s {'Distance': 3, 'latitude': 9.9261153, 'longitude': 78.1140983}
The bin1 level is high
Published data Successfully: %s {'name': 'alert', 'Distance': 95, 'latitude': 9.9261153, 'longitude': 78.1140983}
The bin1 level is low
Published data Successfully: %s {'Distance': 6, 'latitude': 9.9261153, 'longitude': 78.1140983}
The bin1 level is high
Published data Successfully: %s {'name': 'alert', 'Distance': 86, 'latitude': 9.9261153, 'longitude': 78.1140983}


Device ID
Status
Device Type
Class ID
Date Added
Descriptive Location
Added By
Device Class
ssr_123
Disconnected
ssr
Device
Nov 14, 2022 12:09 PM
513419106034@smartinternz.com
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
status
{"Distance":40,"latitude":9.9261153,"longitude":...
json
a few seconds ago
status
{"Distance":35,"latitude":9.9261153,"longitude":...
json
a few seconds ago
status
{"Distance":37,"latitude":9.9261153,"longitude":...
json
a few seconds ago
status
{"Distance":61,"latitude":9.9261153,"longitude":...
json
a few seconds ago
status
{"Distance":59,"latitude":9.9261153,"longitude":...
json
a minute ago



JSON CODE:


```
[
    {
        "id": "9e16f69e17a17a3a",
        "type": "tab",
        "label": "Flow 2",
        "disabled": false,
        "info": "",
        "env": []
    },
    {
        "id": "3a1b64499b64086e",
        "type": "ibmiot in",
        "z": "9e16f69e17a17a3a",
        "authentication": "apiKey",
        "apiKey": "d3d241d760be32e9",
        "inputType": "evt",
        "logicalInterface": "",
        "ruleId": "",
        "deviceId": "",
        "applicationId": "",
        "deviceType": "+",
        "eventType": "+",
        "commandType": "",
        "format": "json",
        "name": "IBM IoT",
        "service": "registered",
        "allDevices": "",
        "allApplications": "",
        "allDeviceTypes": true,
        "allLogicalInterfaces": "",
        "allEvents": true,
```

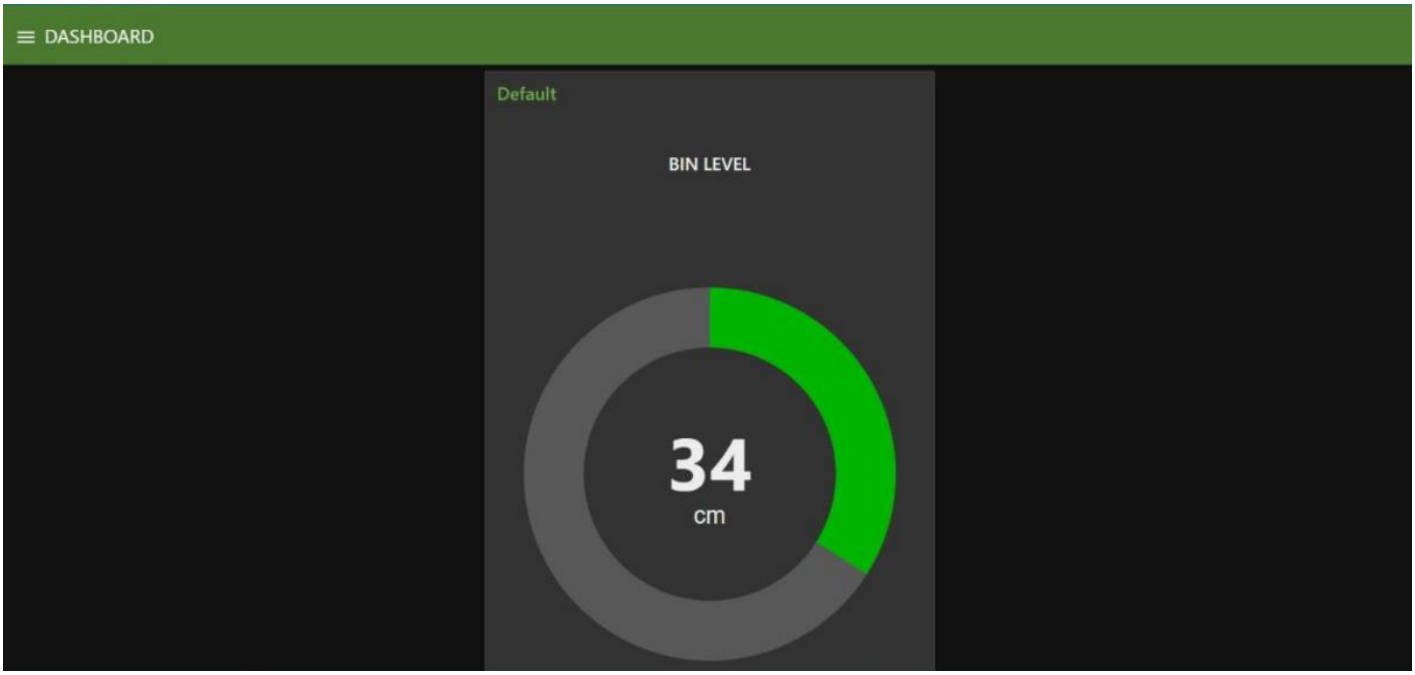
```
        "allCommands": "",
        "allFormats": "",
        "qos": 0,
        "x": 70,
        "y": 300,
        "wires": [
            [
                "864c73fbd8df5b8b",
                "b77a9b3e8f3fb252"
            ]
        ]
    },
    {
        "id": "864c73fbd8df5b8b",
        "type": "function",
        "z": "9e16f69e17a17a3a",
        "name": "function ",
        "func": "var msg1={};\nvar msg2={};\nvar msg3={};\nmsg1.payload = msg.payload.Distance;
\nmsg2.payload = msg.payload.latitude; \nmsg3.payload = msg.payload.longitude;\nreturn [msg1, msg2,
msg3];\n",
        "outputs": 3,
        "noerr": 0,
        "initialize": "",
        "finalize": "",
        "libs": [],
        "x": 340,
        "y": 240,
        "wires": [
            [
                "53694b0547a04328"
            ],
            [
                "d39c51e2d12e96f0"
            ],
            [
                "de6ce8c8f71af009"
            ]
        ]
    },
    {
        "id": "53694b0547a04328",
        "type": "ui_gauge",
        "z": "9e16f69e17a17a3a",
        "name": "Distance",
        "group": "e1f7ae7916f8d6e2",
        "order": 1,
        "width": "6",
        "height": "9",
        "gtype": "donut",
        "title": "BIN LEVEL",
        "label": "cm",
        "format": "{{value}}",
        "min": 0,
        "max": "100",
        "colors": [
```

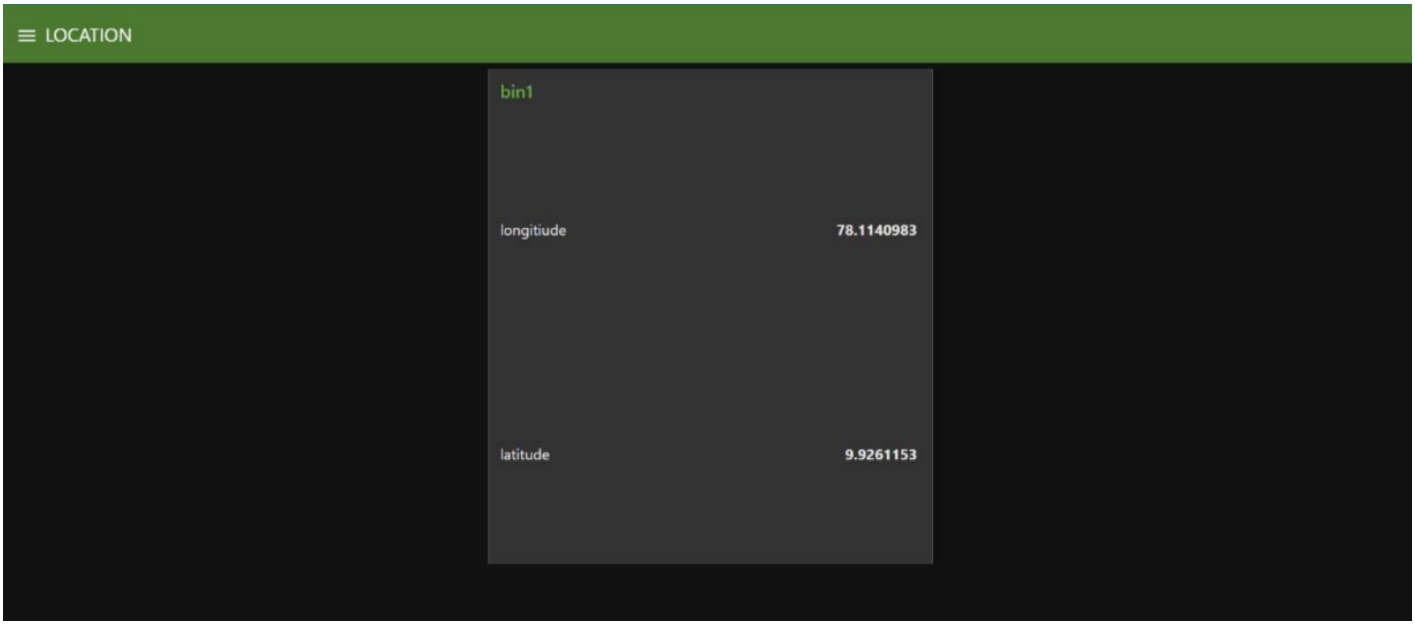
```
        "#00b500",
        "#e6e600",
        "#ca3838"
    ],
    "seg1": "50",
    "seg2": "80",
    "x": 900,
    "y": 200,
    "wires": []
},
{
    "id": "d39c51e2d12e96f0",
    "type": "ui_text",
    "z": "9e16f69e17a17a3a",
    "group": "f9ef24bc62beb59f",
    "order": 2,
    "width": "6",
    "height": "3",
    "name": "latitiude",
    "label": "latitude",
    "format": "{{msg.payload}}",
    "layout": "row-spread",
    "x": 900,
    "y": 240,
    "wires": []
},
{
    "id": "de6ce8c8f71af009",
    "type": "ui_text",
    "z": "9e16f69e17a17a3a",
    "group": "f9ef24bc62beb59f",
    "order": 1,
    "width": "6",
    "height": "3",
    "name": "longitude",
    "label": "longitiude",
    "format": "{{msg.payload}}",
    "layout": "row-spread",
    "x": 900,
    "y": 280,
    "wires": []
},
{
    "id": "46d8538fcca24a22",
    "type": "worldmap",
    "z": "9e16f69e17a17a3a",
    "name": "location",
    "lat": "",
    "lon": "",
    "zoom": "",
    "layer": "OSMC",
    "cluster": "",
    "maxage": "",
    "usermenu": "show",
    "layers": "show",
```

```
        "panit": "false",
        "panlock": "false",
        "zoomlock": "false",
        "hiderightclick": "false",
        "coords": "none",
        "showgrid": "false",
        "showruler": "false",
        "allowFileDrop": "true",
        "path": "/map",
        "overlist": "DR,CO,RA,DN,HM",
        "maplist": "OSMG,OSMC,EsriC,EsriS,EsriT,EsriDG,UKOS",
        "mapname": "GOOGLE MAPS",
        "mapurl": "https://www.google.com/maps",
        "mapopt": "",
        "mapwms": true,
        "x": 880,
        "y": 400,
        "wires": []
    },
    {
        "id": "b77a9b3e8f3fb252",
        "type": "function",
        "z": "9e16f69e17a17a3a",
        "name": "function 2",
        "func": "msg.payload = {  latitude: msg.payload.latitude,\n    longitude: msg.payload.longitude}\nreturn
msg;",
        "outputs": 1,
        "noerr": 0,
        "initialize": "",
        "finalize": "",
        "libs": [],
        "x": 300,
        "y": 380,
        "wires": [
            [
                "65823a0aa9c0e974"
            ]
        ]
    },
    {
        "id": "65823a0aa9c0e974",
        "type": "change",
        "z": "9e16f69e17a17a3a",
        "name": "",
        "rules": [
            {
                "t": "set",
                "p": "payload.name",
                "pt": "msg",
                "to": "worldmap",
                "tot": "str"
            }
        ],
        "action": "",
        "property": "",
```

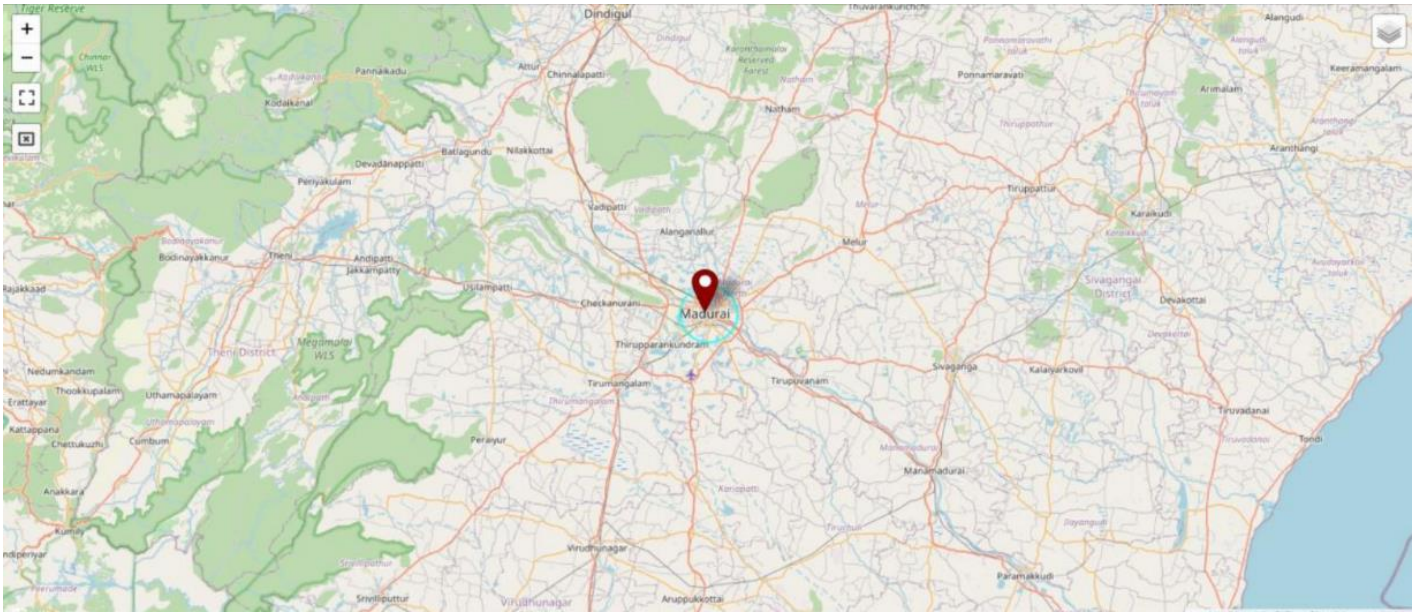
```
    "from": "",
    "to": "",
    "reg": false,
    "x": 560,
    "y": 380,
    "wires": [
      [
        "46d8538fcca24a22"
      ]
    ]
  },
  {
    "id": "d3d241d760be32e9",
    "type": "ibmiot",
    "name": "IOT",
    "keepalive": "60",
    "serverName": "",
    "cleansession": true,
    "appId": "",
    "shared": false
  },
  {
    "id": "e1f7ae7916f8d6e2",
    "type": "ui_group",
    "name": "Default",
    "tab": "004f558216ec7811",
    "order": 1,
    "disp": true,
    "width": "6",
    "collapse": false
  },
  {
    "id": "f9ef24bc62beb59f",
    "type": "ui_group",
    "name": "bin1",
    "tab": "e8096443afed3ad3",
    "order": 1,
    "disp": true,
    "width": "6",
    "collapse": false
  },
  {
    "id": "004f558216ec7811",
    "type": "ui_tab",
    "name": "DASHBOARD",
    "icon": "dashboard",
    "disabled": false,
    "hidden": false
  },
  {
    "id": "e8096443afed3ad3",
    "type": "ui_tab",
    "name": "LOCATION",
    "icon": "dashboard",
    "order": 2,
```

```
        "disabled": false,
        "hidden": false
    }
]
```





Location(detection):



11.2 GIT-HUB LINK: [GitHub-Repo]

PROJECT DEMO LINK:

1. **https://node-red-qltdp-2022-11-07.eu-gb.mybluemix.net/ui/#!/0?socketid=WzX3XVVK_oZjhjBAAAAl**

2. **https://possible-wheat-booth.glitch.me/**