SMART WASTE MANAGEMENT SYSTEM FOR METROPOLITAN CITIES

TEAM ID:PNT2022TMID47365

TEAM MEMBER: JEGATHEESWARAN S

TEAM MEMBER: GNANA EMILSON A

TEAM MEMBER: MANOJKUMAR R

TEAM MEMBER: MUGESH V

PROJECT REPORT

1. INTRODUCTION

1.1 Project Overview

IoT is bringing revolution to almost every aspect of our lives by changing how we do things. The use of Smart IoT devices is on the rise with all the industries heavily investing in IoT. The main aims of investing in IoT are to improve operations efficiency, improve product quality, and reduce the costs of production.

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**

2. LITERATURE SURVEY

2.1 Existing problem

The waste management in metropolitan cities has serious environmental impacts like **water pollution, methane emissions, and soil degradation**. The average density of Indian municipal waste at the point

of collection varies from 400 to 600 kg per cubic meter. At the landfill site, however, the density is much higher because of compaction and putrefaction. Waste incineration (including Waste to Energy) and other thermal processes are **local sources of air pollution**, constituting additional health risk factors to city dwellers, who often already have to cope with serious air contamination issues. Ecosystems vary widely from location to location. However, one of the most outsize consequences of our global waste problem manifests itself in relation to our marine life and waterways. Simply put, it affects **the people who depend on the ocean for their livelihoods**.

Challenges:

1) No segregation at source
2) Incorrect/inadequate segregation techniques
3)Slow adoption of in-house composting
4)Lack of monitoring in housing societies
5)Tips for achieving 100% waste segregation
6) Problems faced by the government with respect to housing societies
7)Key approaches for housing societies to manage waste better

2.2 References

[1] K N Fallavi; V Ravi Kumar; B M Chaithra (2017) Smart waste management using Internet of Things: A survey , *IEEExplore*

[2] Inna Sosunova and Jari Porras (Member, IEEE) 18 July 2022. IoT-Enabled Smart Waste Management Systems for Smart Cities: A Systematic Review. *Ieee Access.*

[3] Saurabh Pargaien; Amrita Verma Pargaien; Dikendra Verma; Vatsala Sah; Neeraj Pandey(2021). Smart Waste Collection Monitoring System using IoT. *IEEExplore*

2.3 Problem Statement Definition

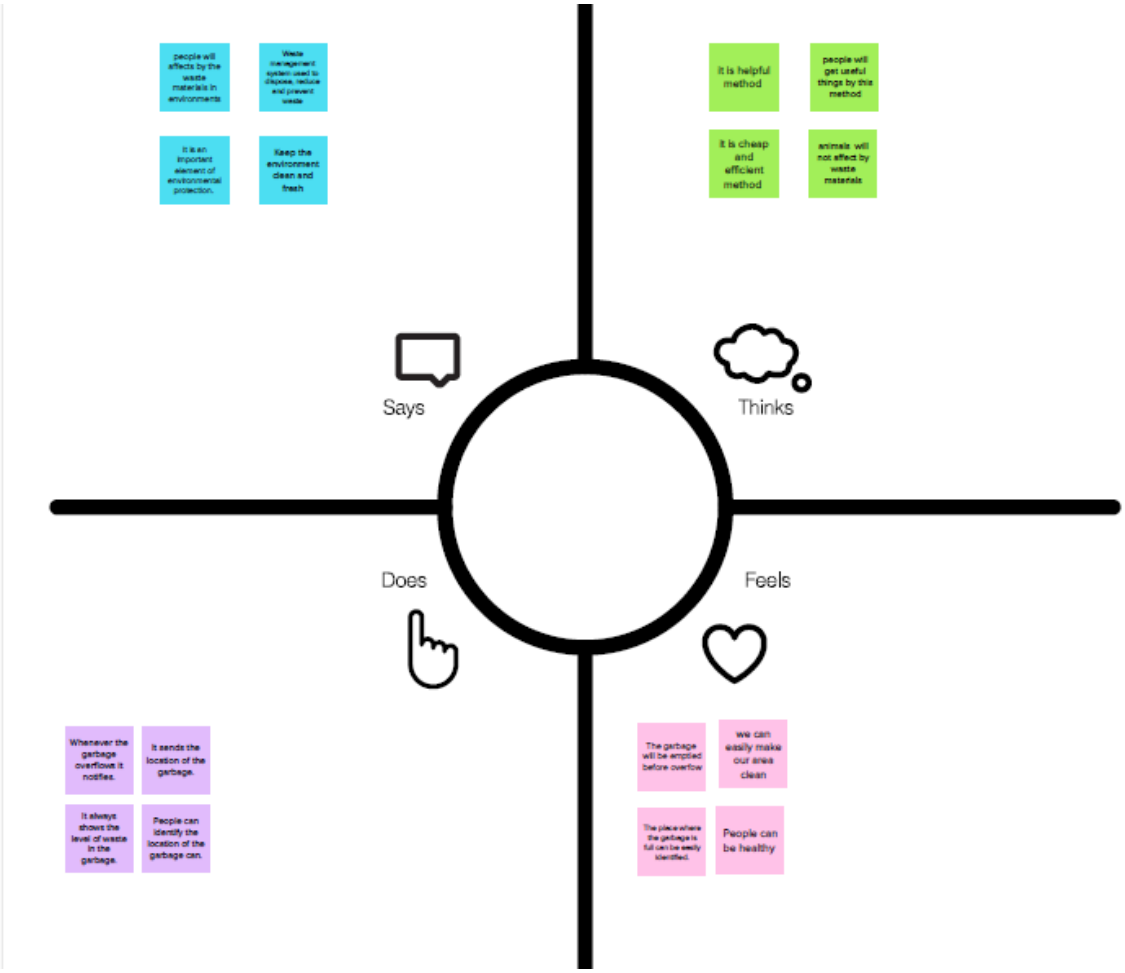
A big challenge in the urban cities is solid waste management. The garbage collecting authority in traditional waste management system doesn't know about the level of garbage in dustbin, if the dust bins gets full by garbage then it gets overflowed as well as spelled out from the dustbin leading to unhygienic condition in cities. People throw garbage on that dustbin which is already overflowed. Sometimes due to unclean garbage bins bad smell arises also toxic and unhygienic gases are produced which is way to support to the air pollution and to some harmful diseases which are easily spreadable. It is very bad look of the city. Use of traditional system result in inefficient and time and money spending system.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy map Canvas

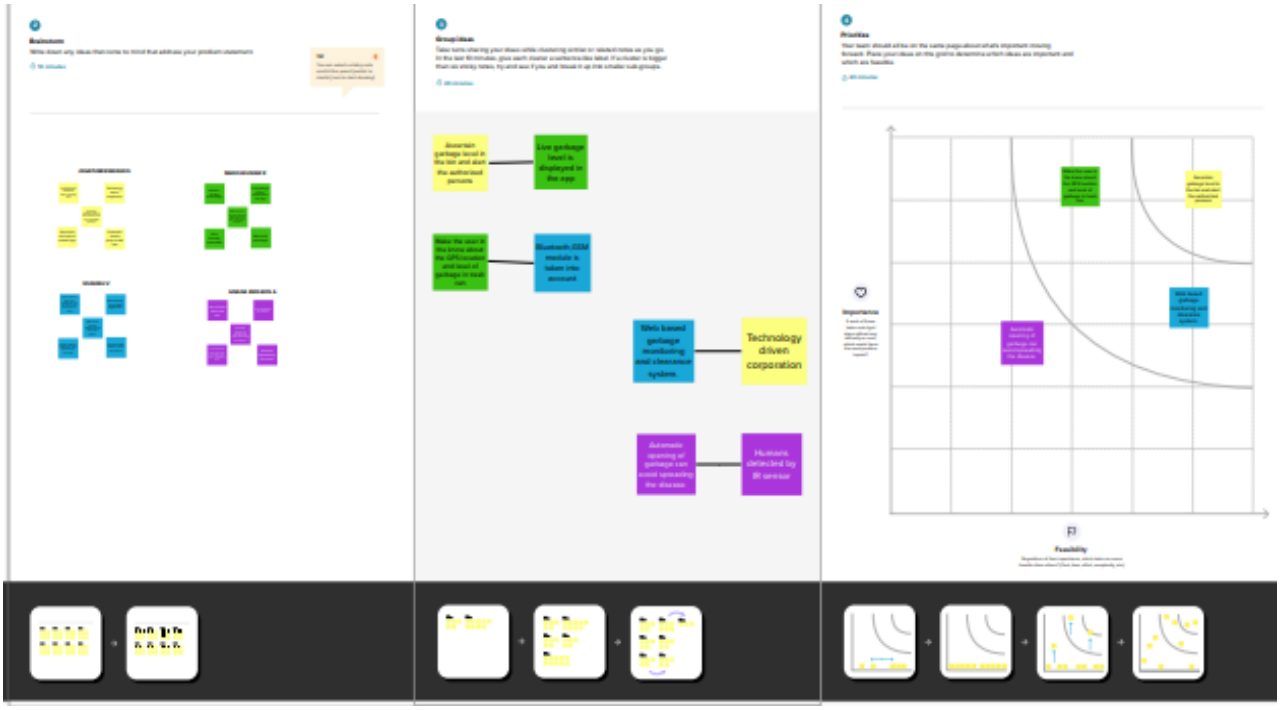
An empathy map is **a collaborative tool teams can use to gain a deeper insight into their customers**. Much like a user persona, an empathy map can represent a group of users, such as a customer segment. The empathy

map was originally created by Dave Gray and has gained much popularity within the agile community.



3.2)Ideation and Brainstorming

Ideation is the process where you generate ideas and solutions through sessions such as sketching, prototyping, brainwriting, worst possible idea and a wealth of other ideation techniques.



3.3) Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement(Problem to be solved)	Normally we have seen that there are many bins which are overflowed in metropolitan cities. Overflowing of trashes pollutes the city streets and it is one of the reasons for epidemics.
2.	Idea/Solution description	By use of the sensors we can detect the level of garbage and sending the information about garbage level to municipality.
3.	Novelty/Uniqueness	Intelligence data collection. Sensor installed on IoT devices is able to collect large volume of useful data.

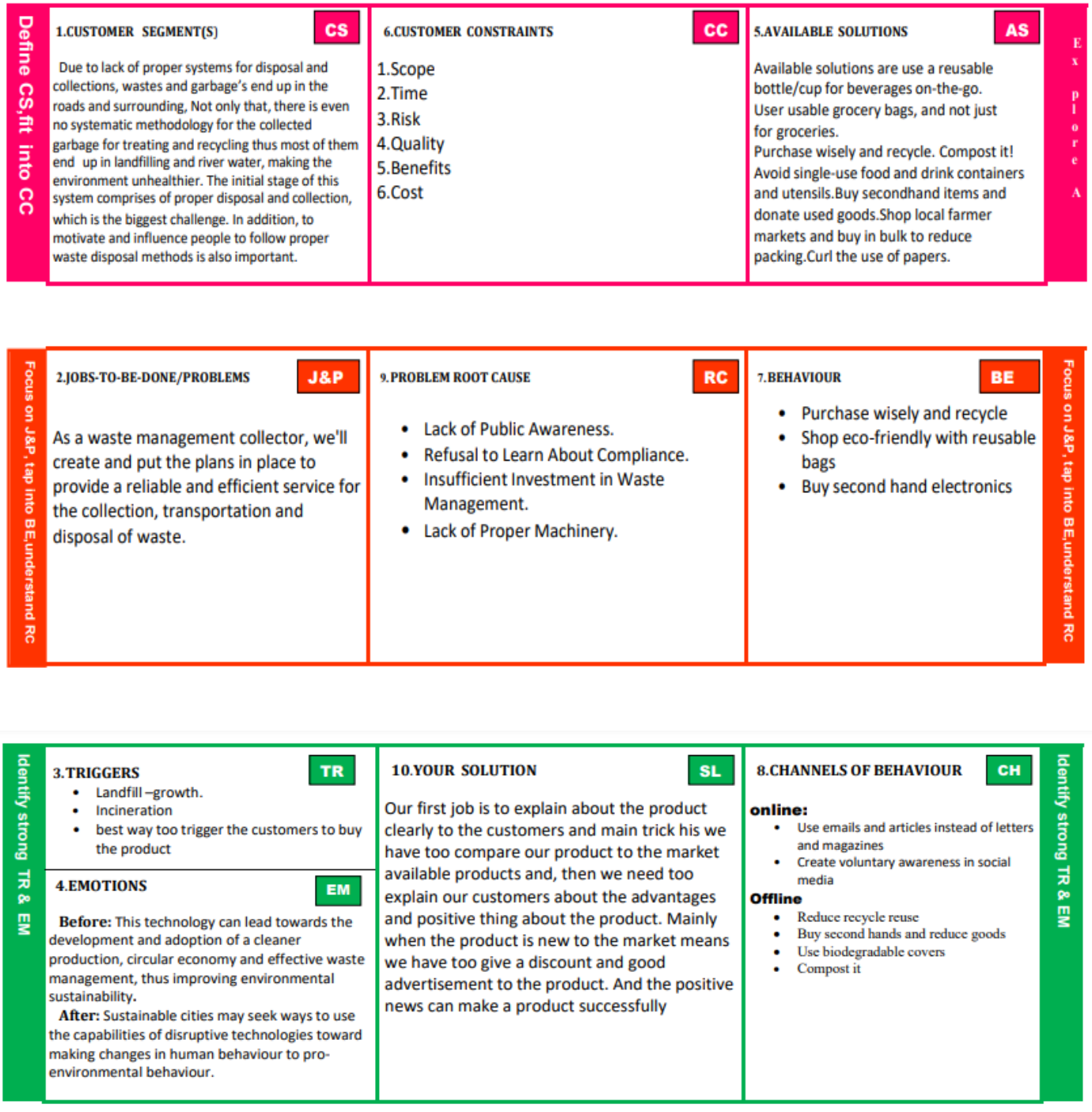
4.	Social Impact/Customer Satisfaction	Makes the lives of the trash collectors comfortable. The city will be clean. The epidemic will be curbed.
5.	Business Model(Revenue Model)	Increased life time of the model. Works even in hazardous environment(medical waste, e-waste, industrial waste i.e. chemicals)
6.	Scalability of the Solution	Works stably even in the busiest metropolitan cities (<100 Million people).

3.4) Problem Statement

Smart waste management is characterized by the usage of technology in order to be more efficient when it comes to managing waste. This makes it possible to plan more efficient routes for the trash collectors who empty the bins, but also lowers the chance of any bin being full for over a week. When placing garbage into the container, a sensor measures its capacity. Then the compactor compresses the waste and again takes a measurement. Finally, the smart bin sends a notification by mail or SMS when the container is ready to be emptied. Smart Bins help to create a cleaner, safer, more hygienic environment and enhanced operational efficiency while reducing management costs, resources, and road-side emissions. The Smart Bin is ideal for busy locations such as campuses, theme parks, airports, railway stations, and shopping malls. Waste management is an important element of environmental protection. Its purpose is to provide hygienic, efficient and economic solid waste storage, collection, transportation and treatment or disposal of waste without polluting the atmosphere, soil or water system. Poor waste management contributes to climate change and air pollution, and directly affects many

ecosystems and species. Landfills, considered the last resort in the waste hierarchy, release methane, a very powerful greenhouse gas linked to climate change.

3.5) **Problem Solution Fit**



4.) REQUIREMENT ANALYSIS

4.1) Functional Requirement:

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	Registration through Form Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Bin Invention	The proposed model provides real time monitoring to thegarbage bins placed in various location. You can see every monitored bin and stand, and you canuse google street view at any time to visit them.
FR-4	Bin Monitoring	The Garbage bins are monitored by smart sensors. the application also forecasts when the bin will be filled based on the past data and capacity of the bin. The sensor will know when the bin was last emptied. So, you can eliminate overflowing bins and cease collecting the empty ones,
FR-5	Notification	The percentage of garbage level will be detected through sensors. When the garbage level is increased above 75%, it givesnotification to the security team. After receiving the notification, the garbage collector collects the garbage.
FR-6	Optimize the route to collect	Waste collectors will use their time effectively by collecting the wastes which requires service rather than travelling the same routes.
FR-7	Database	Information about the location and status of bins will be stored in the database.
FR-8	Feedback	It helps the developer to improve the apps.

4.2) Non Functional Requirement:

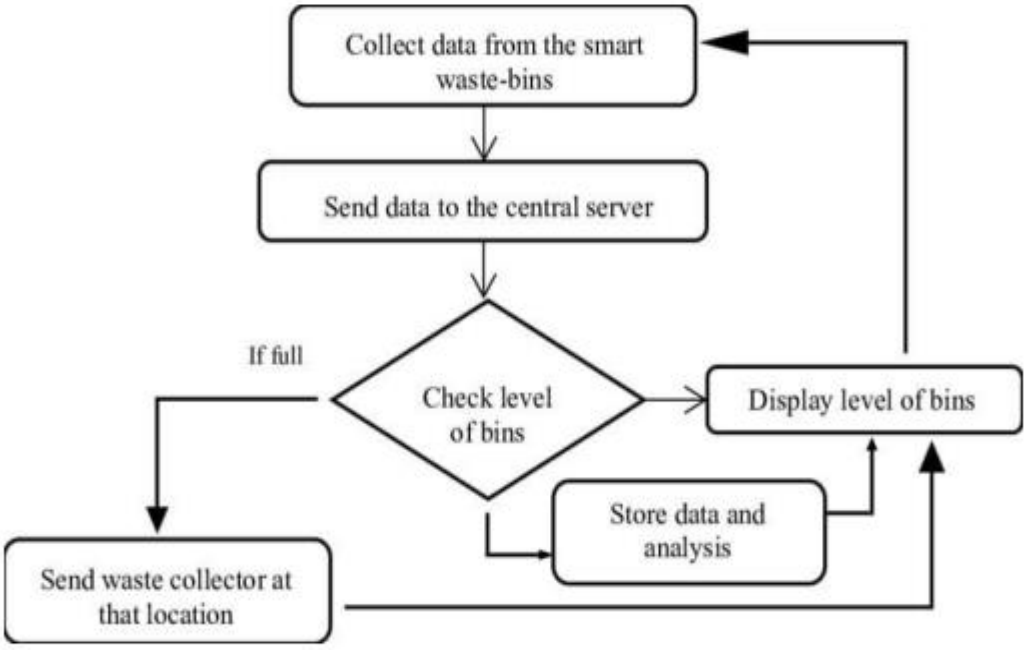
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	IoT device verifies that usability is a special and important perspective to analyze user requirements, which can further improve 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, behavior and experience.
NFR-2	**Security**	Use reusable bottles Use reusable grocery bags Purchase wisely and recycle Avoid single use food and drink containers
NFR-3	**Reliability**	Smart waste management is also about creating better working conditions for waste collectors and drivers. Instead of driving the same collection routes and servicing empty bins, waste collectors will spend their time more efficiently, taking care of bins that need servicing.
NFR-4	**Performance**	The Smart Sensors use ultrasound technology to measure the fill levels (along with other data) in bins several times a day. Using a variety of IoT networks (NB-IoT, GPRS), the sensors send the data to Sensoneo's Smart Waste Management Software System, a powerful cloud-based platform, for data-driven daily operations, available also as a waste management app. Customers are hence provided data-driven decision making, and optimization of waste collection routes, frequencies, and vehicle loads resulting in route reduction by at least 30%.
NFR-5	**Availability**	By developing & deploying resilient hardware and beautiful software we empower cities, businesses, and countries to manage waste smarter.
NFR-6	**Scalability**	Using smart bins may reduce the number of bins inside the cities because we monitor the garbage 24/7 more efficient.

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



A clever waste management platform uses analytics to translate the statistics accumulate for your bins into actionable insights to help you improve your wasteofferings. You can receive information on metric inclusive of:

The first check conducted is the situation in which the rubbish bin is empty or its garbage level could be very low.

Then, the bin is packed with greater rubbish till it is level has handed the first thresholdprice, which is ready to eighty% then the first caution SMS is being dispatched, as depicted.

The first notification SMS sent with the aid of the machine, once the waste reaches the level of eighty-five% full.

The 2d notification SMS dispatched with the aid of the machine, indicating that bin is atleast ninety-five% complete and the rubbish needs to be collected at once.

Location vulnerable to overflow.

The variety of packing containers needed to keep away from overflowing waste.

The quantity of collection offerings that could be saved.

The amount of gas that might be saved.

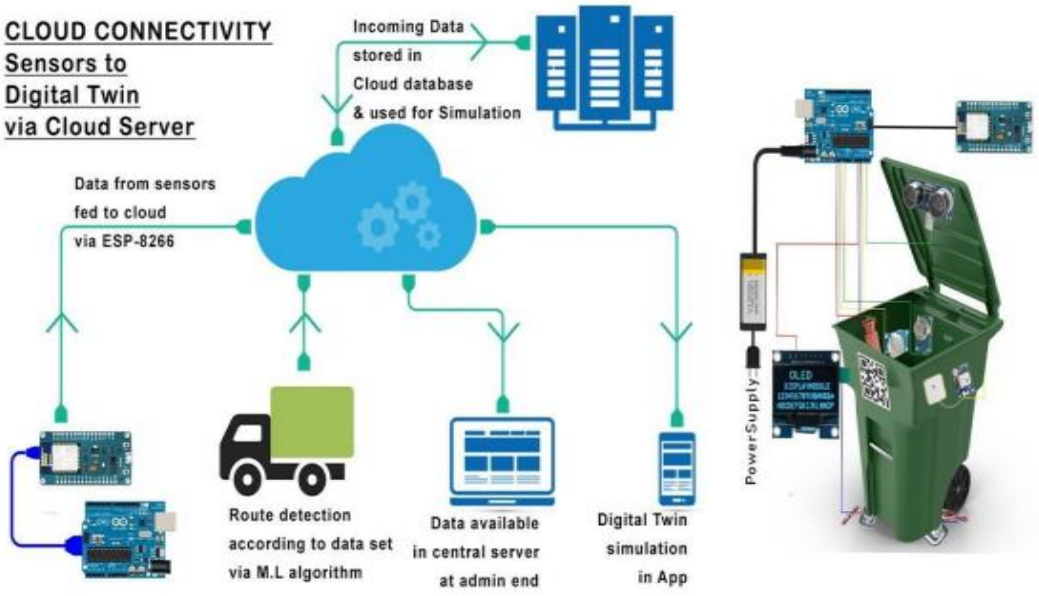
The driving distance that might be saved.

5.2) **Solution and Technical Architecture**

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Discover the finest technological solution to address current company issues..
- Describe the software's design, features, functionality, and other elements to the project's stakeholders..
- Specify the features, stages of development, and requirements for the solution.
- Offer guidelines for how the solution is created, managed, and delivered.

Solution Architecture Diagram



5.3) User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Admin (who manage web server	Login	USN-1	As an admin, I gave user id and password for every worker and 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 gets filling alert, I will post location and garbage I'd to trash truck.	I can manage garbage monitoring.	High	Sprint-2
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-3
Total garbage collector	Login	USN-4	As a Waste Collector, I'll collect all the trash from garbage and load into garbage truck and send them to landfill.	I can collect trash and pulled to truck and send off.	Medium	Sprint-4
Municipality	Login	USN-5	As a municipality, I'll check the process are happening in discipline manner without any issues.	I can manage all these processes going well.	High	Sprint-5

6.) PROJECT PLANNING AND SCHEDULING

6.1) Sprint Planning & Estimation

Product Backlog, Sprint Schedule and Estimation.

Title	Description	Date
Literature Survey & Information Gathering	Literature survey on the selected project & gathering information by referring the, technical papers, research publications etc..	19 OCTOBER 2022
Prepare Empathy map	Prepare Empathy map Canvas to capture the user's pains & Gains, Prepare list of problem and statements.	10 SEPTEMBER 2022
Ideation	List the by organizing brainstorm session & prioritize the top 3 ideas based on the feasibility & importance.	10 SEPTEMBER 2022
Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc..	24 OCTOBER 2022
Proposed Solution Fit	Prepare problem – solution fit document.	26 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 Requirement	Prepare the functional requirement document.	26 OCTOBER 2022
Data Flow Diagram	Draw the data flow diagram and submit for review.	26 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 PROGESS

6.2) **Sprint Delivery Schedule**

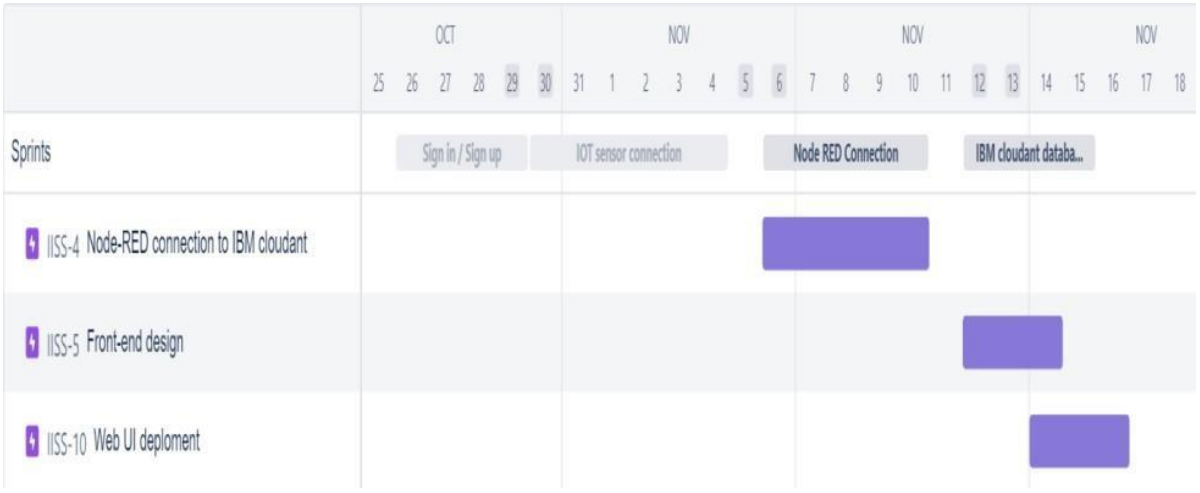
Project tracker, Velocity and Burndown Chart.

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	5 Days	26 Oct 2022	28 Oct 2022	20	28 Oct 2022
Sprint-2	20	5 Days	2 Nov 2022	06 Nov 2022	20	06 Nov 2022
Sprint-3	20	5 Days	07 Nov 2022	15 Nov 2022	20	15 Nov 2022
Sprint-4	20	5 Days	13 Nov 2022	19 Nov 2022	20	19 Nov 2022

Velocity:
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

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

6.3 JIRA REPORT



7. CODING AND SOLUTIONING

7.1 CODING

```
import random
import ibmiotf.application
import ibmiotf.device
import time
from time import sleep
import sys

#IBM Watson Device Credentials.
organization = "sl1jtd"
deviceType = "abcde"
deviceId = "08"
authMethod = "token"
authToken = "830119106008"

def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status = cmd.data['command']
try:
    deviceOptions = {"org": organization, "type": deviceType, "id":
deviceId, "auth-method": authMethod, "auth-token": authToken}
    deviceCli = ibmiotf.device.Client(deviceOptions)
except Exception as e:
    print("Caught exception connecting device: %s" % str(e))
    sys.exit()

deviceCli.connect()
while True:
 distance= random.randint(10,70)
```

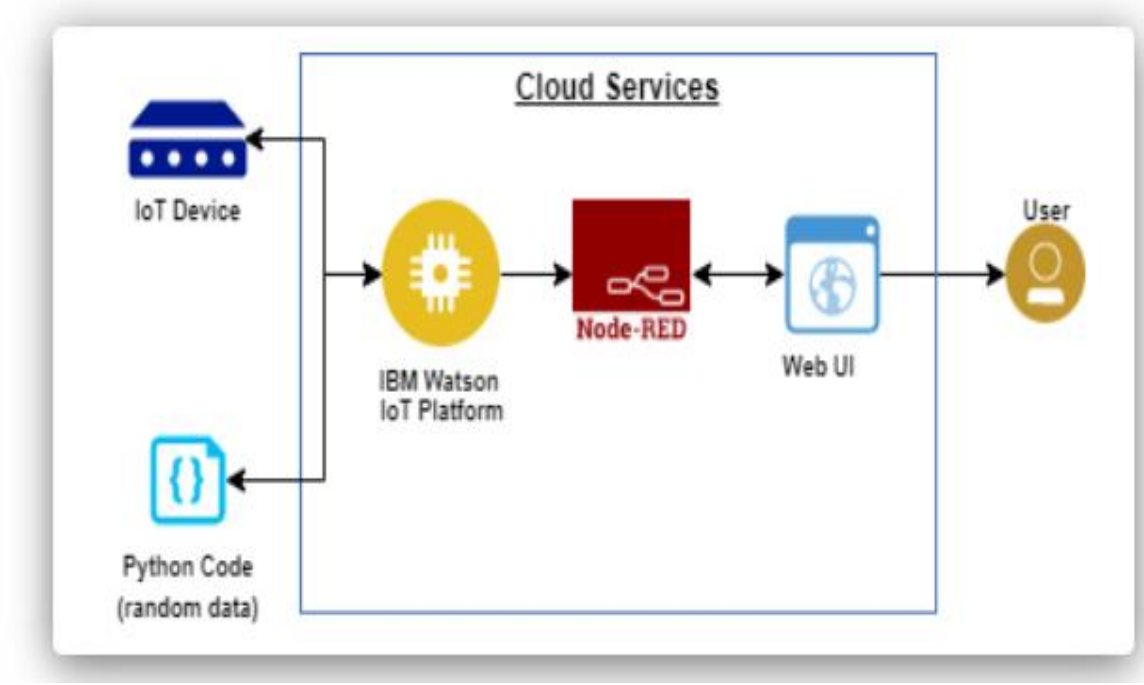
```
loadcell= random.randint(5,15)
data= {'dist':distance,'load':loadcell}

#weight of the bin
if loadcell <= 13 and loadcell >= 15:
    load = "90 %"

elif loadcell <= 10 and loadcell >= 12:
    load = "75 %"

elif loadcell <= 8 and loadcell >= 10:
    load = "75 %"

elif loadcell <= 4 and loadcell >= 7:
    load = "45 %"

else:
    load = "20 %"

#empty distance in the bin
if distance <= 15:
    dist = '90 %'

elif distance <= 30 and distance >= 16:
    dist = '75 %'

elif distance <= 45 and distance >= 31 :
    dist = '60 %'

elif distance <= 60 and distance >= 46:
    dist = '45 %'

else:
    dist = '25 %'
```

```
 #alert and warning for garbage level and weight

 if load == "90 %" or dist == "90 %":
     warn = 'alert : Garbage level in the trash can is going to be full,
Time to collect '

 elif load == "75 %" or dist == "75 %":
     warn = 'alert : Garbage level is above 75%'

 elif load == "60 %" or dist == "60 %":
     warn = 'alert : Garbage level is above 60%'

 else :
     warn = 'alert :' 'No need to collect right now '

 def
myOnPublishCallback(lat=10.677849504740825,long=78.5996641
1452089):
     print("Sethurapatti, Trichy")
     print("distance = %s " %distance,"loadcell:%s " %loadcell,"lon
= %s " %long,"lat = %s" %lat)
     print(load)
     print(dist)
     print(warn)

 time.sleep(5)


 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
```

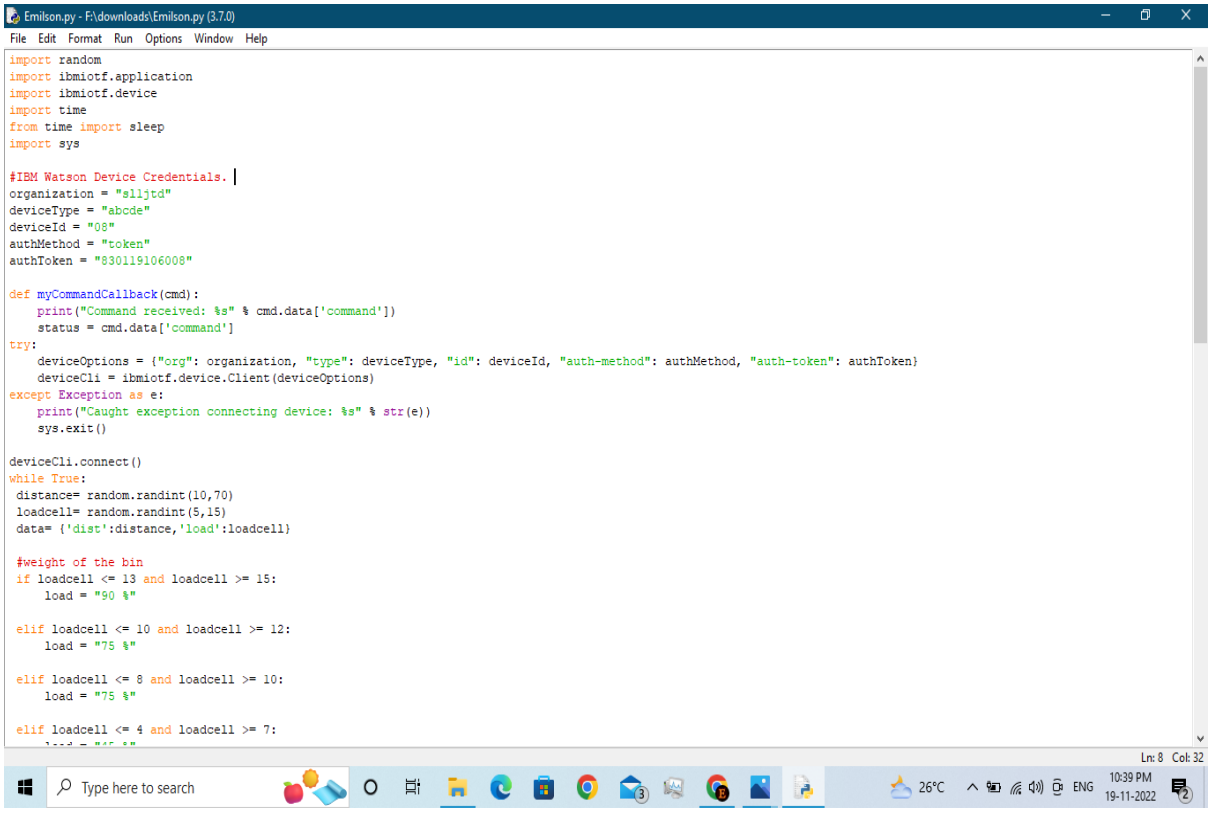
DATABASE SCHEMA

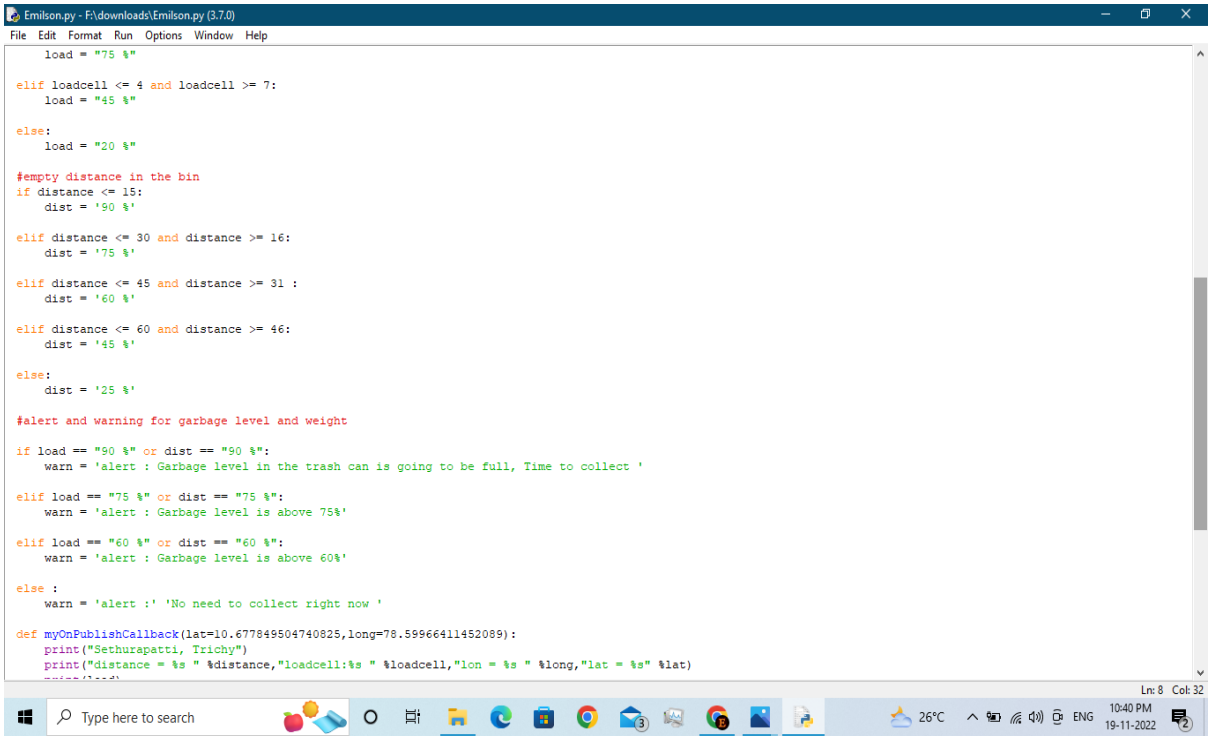


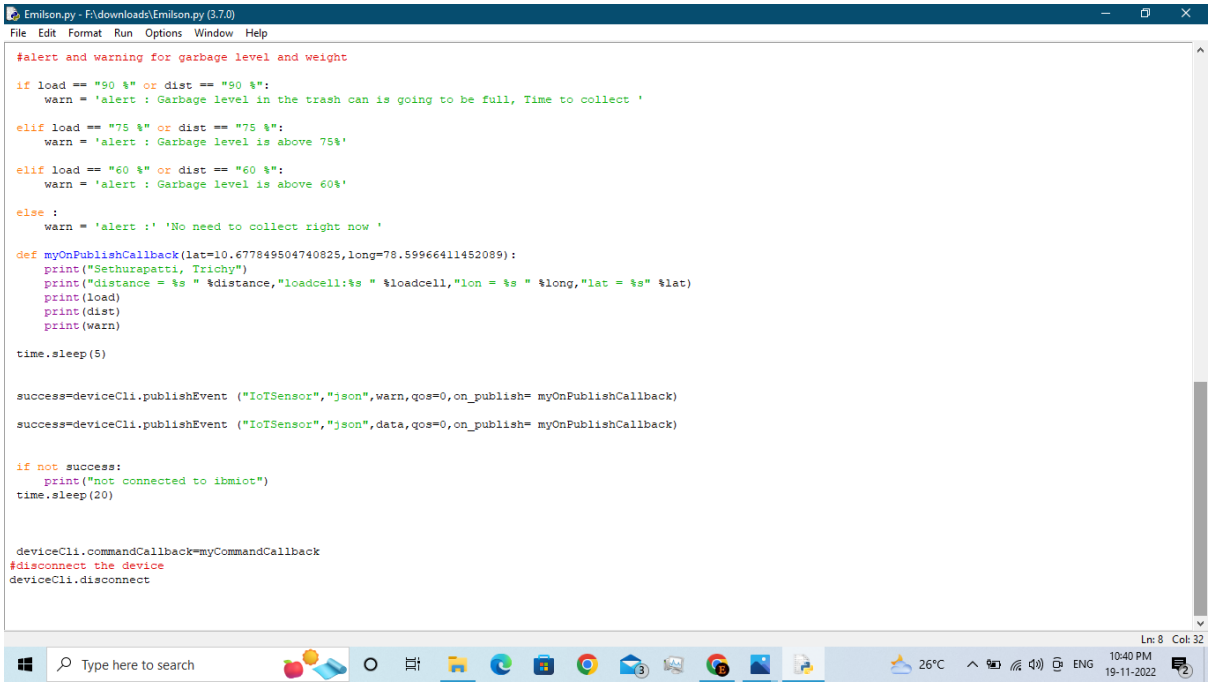
8. TESTINGS

8.1 TESTCASES

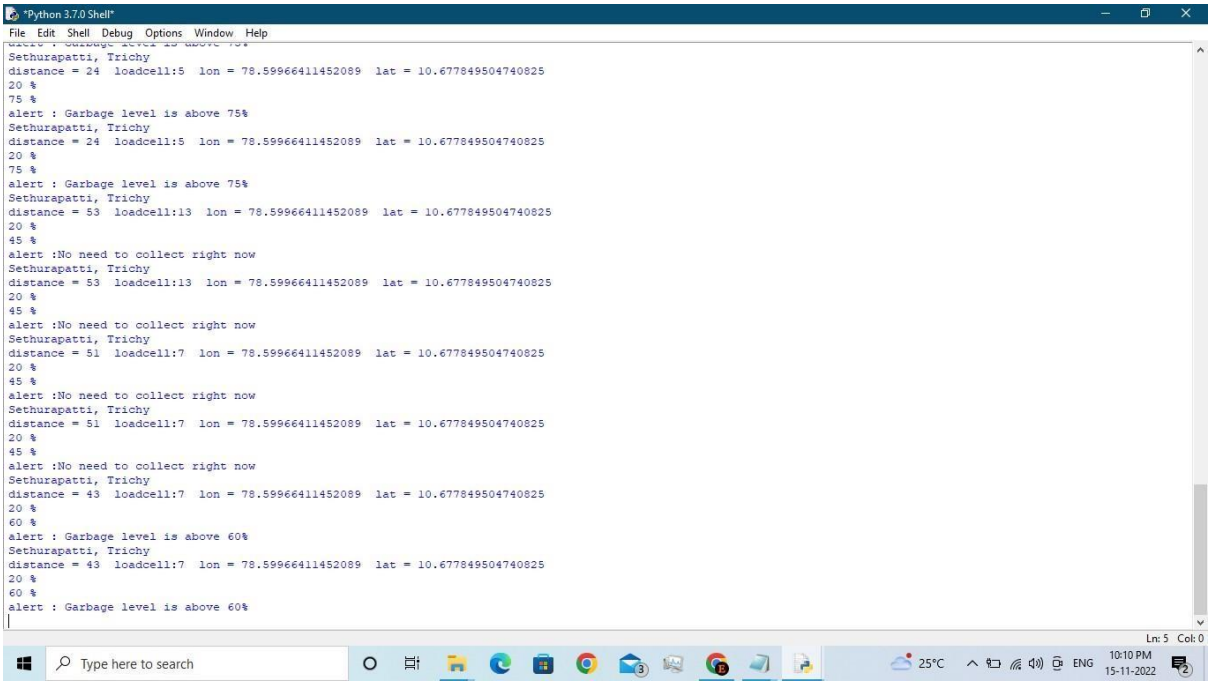
The training output of the source code seen in the above chapter was executed successfully and got the output



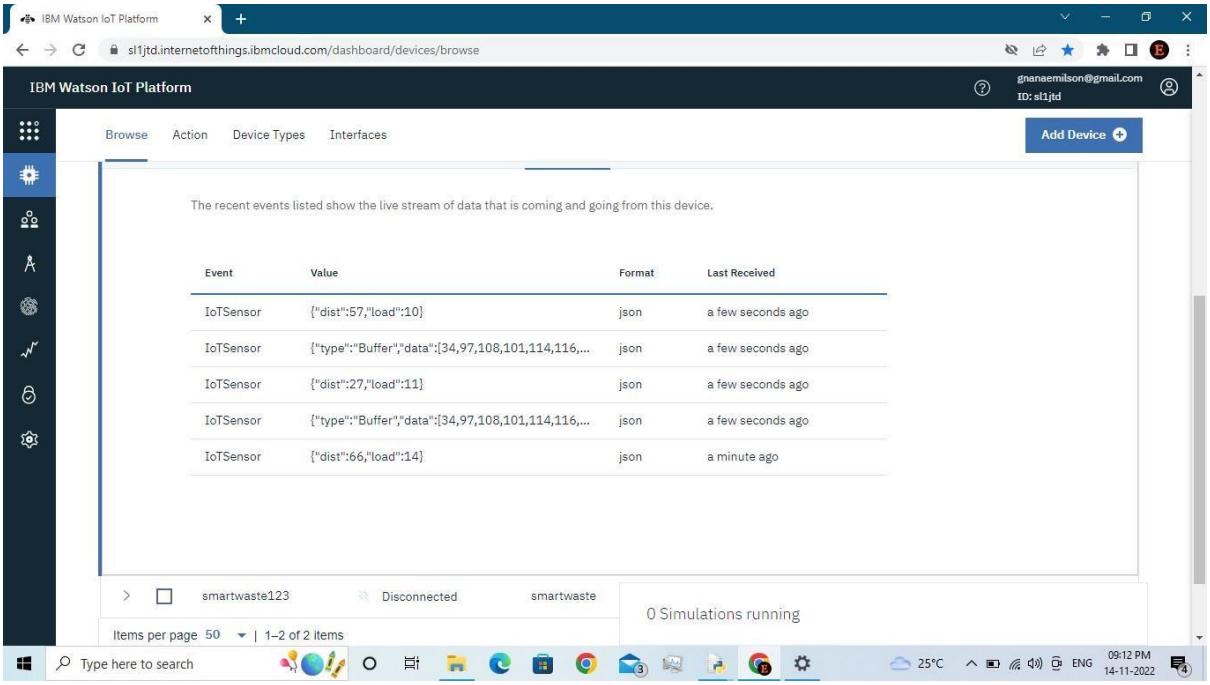

Emilson.py - F:\downloads\Emilson.py (3.7.0)
File Edit Format Run Options Window Help
load = "75 %"
elif loadcell <= 4 and loadcell >= 7:
load = "45 %"
else:
load = "20 %"
#empty distance in the bin
if distance <= 15:
dist = '90 %'
elif distance <= 30 and distance >= 16:
dist = '75 %'
elif distance <= 45 and distance >= 31 :
dist = '60 %'
elif distance <= 60 and distance >= 46:
dist = '45 %'
else:
dist = '25 %'
#alert and warning for garbage level and weight
if load == "90 %" or dist == "90 %":
warn = 'alert : Garbage level in the trash can is going to be full, Time to collect '
elif load == "75 %" or dist == "75 %":
warn = 'alert : Garbage level is above 75%'
elif load == "60 %" or dist == "60 %":
warn = 'alert : Garbage level is above 60%'
else :
warn = 'alert :' 'No need to collect right now '
def myOnPublishCallback(lat=10.677849504740825,long=78.59966411452089):
print("Sethurapatti, Trichy")
print("distance = %s " %distance,"loadcell:%s " %loadcell,"lon = %s " %long,"lat = %s" %lat)
Ln: 8 Col: 32
Type here to search
26°C
ENG
10:40 PM
19-11-2022


Emilson.py - F:\downloads\Emilson.py (3.7.0)
File Edit Format Run Options Window Help
#alert and warning for garbage level and weight
if load == "90 %" or dist == "90 %":
warn = 'alert : Garbage level in the trash can is going to be full, Time to collect '
elif load == "75 %" or dist == "75 %":
warn = 'alert : Garbage level is above 75%'
elif load == "60 %" or dist == "60 %":
warn = 'alert : Garbage level is above 60%'
else :
warn = 'alert :' 'No need to collect right now '
def myOnPublishCallback(lat=10.677849504740825,long=78.59966411452089):
print("Sethurapatti, Trichy")
print("distance = %s " %distance,"loadcell:%s " %loadcell,"lon = %s " %long,"lat = %s" %lat)
print(load)
print(dist)
print(warn)
time.sleep(5)
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
Ln: 8 Col: 32
Type here to search
26°C
ENG
10:40 PM
19-11-2022

OUTPUT:



IBM WATSON:

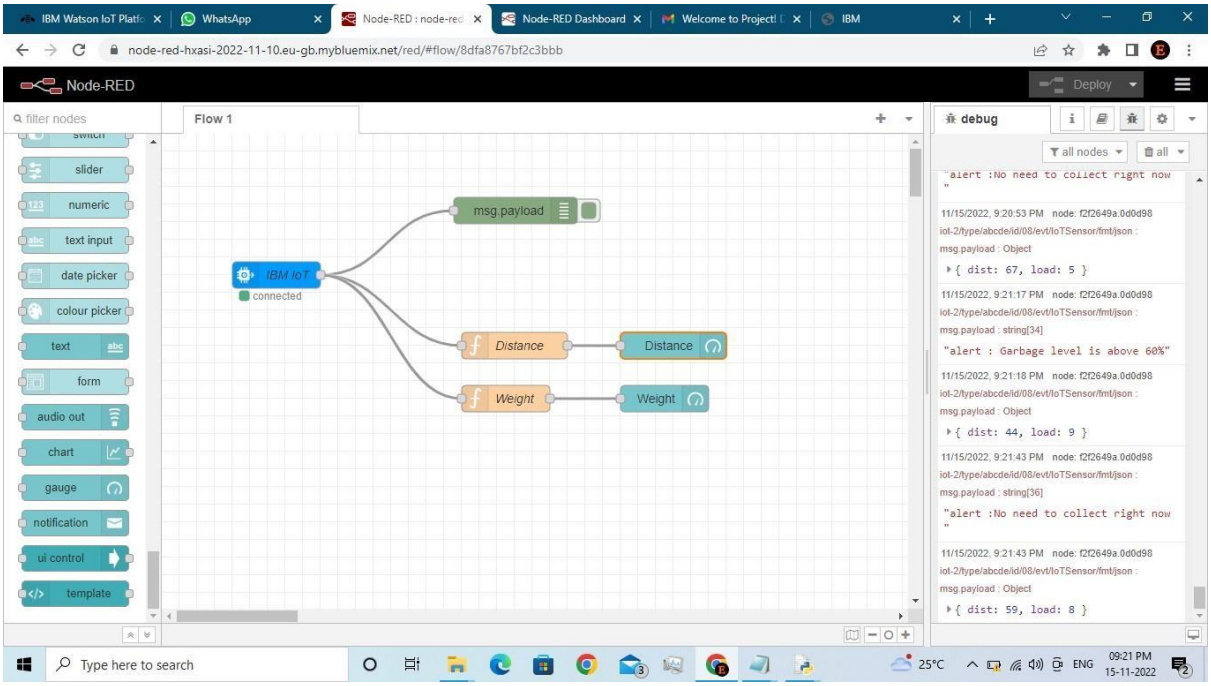


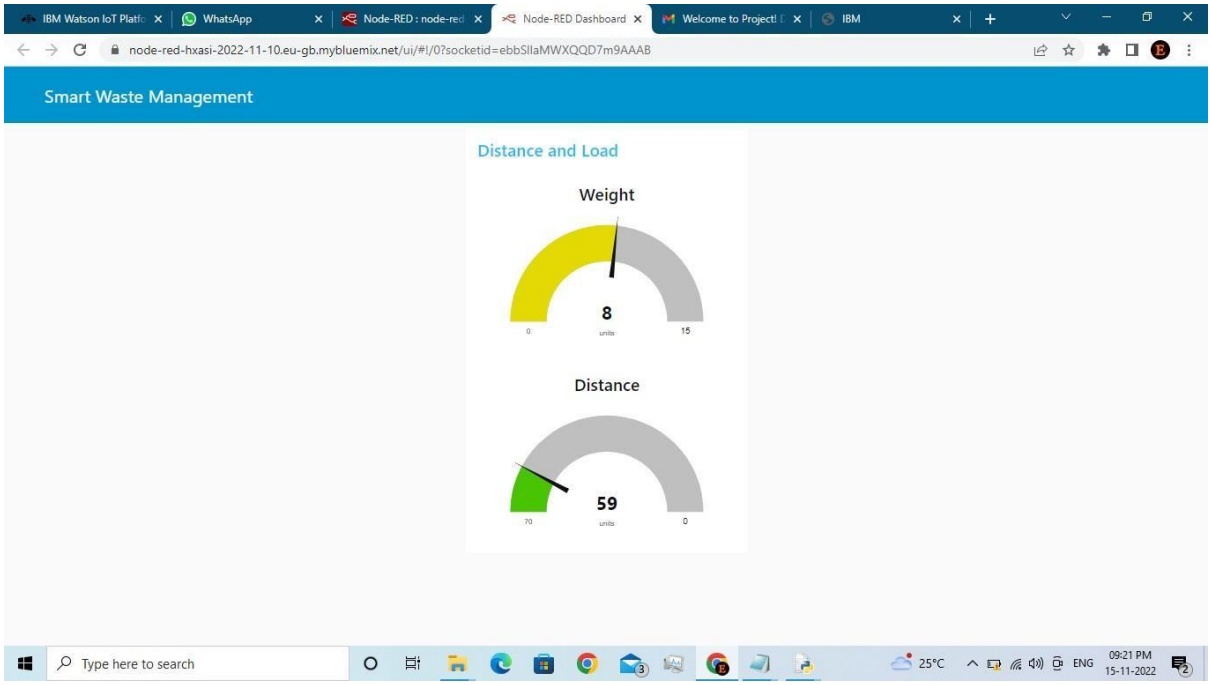
8.2 USER ACCEPTANCE TESTING:

The obtained output get stimulated and got the result with the random values.

Display the image and pre-process the level of the Node-RED web UI and display weight and distance.

.





9. RESULTS

We have successfully completed the project works that the integrated Smart systems for metropolitan cities and measuring the load and level of the bin and to inform the municipality workers using web application.

10.ADVANTAGES AND DISADVANTAGES

10.1 ADVANTAGES

1] Reduction in Collection Cost

2] No Missed Pickups

3] Reduced Overflows

4] Waste Generation Analysis

5] CO2 Emission Reduction

10.2 DISADVANTAGES

- System requires a greater number of waste bins for separate waste collection as per population in the city.

- 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.

According to the author there may be several disadvantages such as increasing cost of the dustbin. For example, if there are three different levels then three sensors has to be placed; one sensor for each level. Also rough action and usage of the user may cause damages to the sensors.

11.CONCLUSION

A Smart Waste Management system that is more effective than the one in use now is achievable by using sensors to monitor the filling of bins. Our conception of a "smart waste management system" focuses on monitoring waste management, offering intelligent technology for waste systems, eliminating human intervention, minimizing human time and effort, and producing a healthy and trash-free environment. The suggested approach can be implemented in smart cities where residents have busy schedules that provide little time for garbage management. If desired, the bins might be put into place in a metropolis where a sizable container would be able to hold enough solid trash for a single unit. But these may price bit high.

12.FUTURE SCOPE:

There are several future works and improvements for the proposed system, including the following:

- Changes the system of user authentication and atomic lock of bins, which would aid in protecting the bin from damage.

- Having case study or data analytics on the type and times waste is collected on different days or seasons, making the bin level predictable and remove the reliance on electronic components, and fixing the coordinates.

13.APPENDIX

SOURCE CODE

```
import random
import ibmiotf.application
import ibmiotf.device
import time
from time import sleep
import sys

#IBM Watson Device Credentials.
organization = "sl1jtd"
deviceType = "abcde"
deviceId = "08"
authMethod = "token"
authToken = "830119106008"

def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status = cmd.data['command']
try:
    deviceOptions = {"org": organization, "type": deviceType, "id":
deviceId, "auth-method": authMethod, "auth-token": authToken}
    deviceCli = ibmiotf.device.Client(deviceOptions)
except Exception as e:
    print("Caught exception connecting device: %s" % str(e))
    sys.exit()

deviceCli.connect()
while True:
 distance= random.randint(10,70)
 loadcell= random.randint(5,15)
 data= {'dist':distance,'load':loadcell}
```

```
#weight of the bin
if loadcell <= 13 and loadcell >= 15:
    load = "90 %"

elif loadcell <= 10 and loadcell >= 12:
    load = "75 %"

elif loadcell <= 8 and loadcell >= 10:
    load = "75 %"

elif loadcell <= 4 and loadcell >= 7:
    load = "45 %"

else:
    load = "20 %"

#empty distance in the bin
if distance <= 15:
    dist = '90 %'

elif distance <= 30 and distance >= 16:
    dist = '75 %'

elif distance <= 45 and distance >= 31 :
    dist = '60 %'

elif distance <= 60 and distance >= 46:
    dist = '45 %'

else:
    dist = '25 %'

#alert and warning for garbage level and weight

if load == "90 %" or dist == "90 %":
```

```
    warn = 'alert : Garbage level in the trash can is going to be full,
Time to collect '

 elif load == "75 %" or dist == "75 %":
    warn = 'alert : Garbage level is above 75%'

 elif load == "60 %" or dist == "60 %":
    warn = 'alert : Garbage level is above 60%'

 else :
    warn = 'alert :' 'No need to collect right now '

 def
myOnPublishCallback(lat=10.677849504740825,long=78.5996641
1452089):
    print("Sethurapatti, Trichy")
    print("distance = %s " %distance,"loadcell:%s " %loadcell,"lon
= %s " %long,"lat = %s" %lat)
    print(load)
    print(dist)
    print(warn)

 time.sleep(5)


 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
```

GitHub & Project Demo Link

https://github.com/IBM-EPBL/IBM-Project-46019-1660734749

https://drive.google.com/file/d/1ZuD7DpBWVwxUeOWUyyopNH0rmE_y6kS2/view?usp=drivesdk