Smart Waste Management System For Metropolitan Cities

Team id: PNT2022TMID08724

SUBMITTED BY

VIVIN ROSAN.M	727619BEC071
RAMANATHAN.S	727619BEC091
MONESHWAR.V	727619BEC095
ARUNACHALAM.S	727620BEC099

In partial fulfilment for the award of the degree of

BACHELOR OF ENGINEERING

in

ELECTRONICS AND COMMUNICATION ENGINEERING

Dr . MAHALINGAM COLLEGE OF ENGINEERING AND TECHNOLOGY An Autonomous Institution Affiliated to ANNAUNIVERSITY CHENNAI – 600 025

1.INTRODUCTION:

1.1.Project overview

Smart waste management system focuses on solving the previously mentioned solid waste management problems using sensors, intelligent monitoring systems, and mobile applications. The first smart waste management solution to make the waste collection process more efficient is sensors. Sensors can measure the fill level of the containers and provide updated information at any time and notify waste management services to empty them when they are full or almost full

These devices help optimize the best possible route containing fully filled containers and create smart schedules for drivers. The selection of the containers also minimizes the need for trash collection staff because their duties are deduced. They can also alert the waste management companies or municipalities if an undesirable incident happens such as sudden temperature rise or displacement of the container by their GPS features.

1.2.PURPOSE:

Reduction in the collection and unnecessary fuel consumption cost:

- Due to using smart dumpsters, there will be no need for a physical check for every container. This smart waste management solution reduces fuel consumption and cost. Therefore, this reduction allows waste collection companies or municipalities to allocate their resources efficiently.

Elimination of missed pickups:

- As route optimization has become a must for smart waste management, there will be no more overflowed trash bins while almost full ones will be taken into account when they are are completely full.

Waste generation geo-specific data analysis:

- Data constitutes the basis of the smart waste management system, and it is also used to follow the patterns that occur according to the regions. Data analysis produced by IoT gives customers demographic analysis and creates a chance to take action according to the filling patterns of that district.

Reduction of CO2 emissions:

- Due to the more strategic scheduling of garbage collection trucks with real-time data, the carbon footprint is reduced. Thus, smart waste management solutions make the traditional waste collection system more environmentally friendly in each step

2. LITERATURE SURVEY

2.1. Existing problem:

Waste bins are part of our lives for decades and mostly its condition are overflowing due to improperwaste dumping, collection and management, which leads in foul smell and unhygienic condition, thus inherently results in environment pollution. Therefore, in this paper, design of a Waste Bin with real time monitoring is presented and a smart waste management system is proposed using the recent technical advancements of automation and Internet of Things (IoT). The capacitance sensor in the bincontinuously monitors the level of the bin in real time and communicates to the central cloud where the bins are connected. Ultrasonic sensor is used to open and close the lid of the bin

whenever the persons are nearby the bin. Such smart bins are connected to the cloud, where the bin status are communicated, recorded and monitored by the local bodies through and android app or a centralizedserver. Thus the designed smart bin and proposed waste management system have better level of smartness compared to existing ones in metropolitan cities in a centralized manner.

2.2. REFERENCES:

[1] Hitesh Poddar, Rituraj Paul, Sourangsu Mukherjee & Budhaditya Bhattacharyya. (2017). Design of smart bin for smarter cities. In. IEEE Proceedings of Innovations in Power and Advanced Computing Technologies (i-PACT), Vellore.1-6.

[2] Rajkumar Joshi & Sirajuddin Ahmed. (2016). Status and challenges of municipal solid waste management in India: A review. Cogent Environmental Science, 2: 1139434, 1-18.

[3] Eunice Likotiko, Dmitry Petrov, Joseph Mwangoka & Ulrich Hilleringmann. (2018). Real time solid waste monitoring using cloud and sensors technologies. The Online Journal of Science and Technology, 8(1), 106-116.

[4] Sreejith S, Ramya R, Roja R. & Sanjay Kumar A. (2019). Smart Bin for Waste Management System. In. Poceedings of the IEEE 5 th International Conference on Advanced Computing & Communication Systems, Coimbatore, India, 1079-1082.

[5] Hassan, S. A., Jameel, N.G.M. & Boran. S. (2016). Smart Solid Waste Monitoring and Collection System. International Journal of Advanced Research in Smart Solid Waste Monitoring and Collection System, 6(10), 7–12.

[6] Bhor, V., Morajkar, P. & Amol Deshpande. (2015). Smart Garbage Management System. International Journal of Engineering Research & Technology (IJERT), 4(3), 1117–1120.

[7] Catania, V. & Ventura, D. (2014). An approach for monitoring and smart planning of urban solid waste management using smart-M3 platform. In Proceedings of the IEEE 15th Conference of Open Innovations Association FRUCT, Saint Petersburg, Russia, 24-31.

[8] Bashir, A., Banday, S. A., Ab. Rouf Khan & Mohammad Shafi. (2013). Concept, Design and Implementation of Automatic Waste Management System. International Journal on Recent and Innovation Trends in

Computing and Communication, 1(7), 604-609.

[9] Serbulent Tozlu; Murat Senel; Wei Mao. & Abtin Keshavarzian. (2012). Wi-Fi enabled sensors for internet of things: A practical approach. IEEE Communications Magazine, 50(6), 134-143.

[10] Xu Li, Rongxing Lu, Xiaohui Liang, and Xuemin (Sherman) Shen, Jiming Chen & Zhejiang Xiaodong Lin. (2011). Smart Community: An Internet of Things Application. IEEE Communications Magazine, 49(11), 68-75.

[11] Luca Foschini, Tarik Taleb &Antonio Corradi. (2011). M2M-based metropolitan platform for IMS-enabled road traffic management in IoT," in IEEE Communications Magazine, 49(11), 50-57.

[12] R. Prakash More & Anil. S. Hiwale. (2016). A Reconfigurable Smart Sensor Interface for Industrial WSN in IoT Environment. International Journal of Innovative Research in Science, Engineering and Technology, 5, 8986-8994.

[13] M. Li & H. Lin. (2015). Design and Implementation of Smart Home Control Systems Based on Wireless Sensor Networks and Power Line Communications. In. IEEE Transactions on Industrial Electronics, 62(7), 4430-4442.

[14] I. S. Hong, S. H. Park, B. S. Lee, J. K. Lee, D. B Jeong & S. H. Park. (2014). IoT-Based Smart Garbage System for Efficient Food Waste Management. The Scientific World Journal, 2014, 646953, 1-13.

[15] Cerchecci M, Luti F, Mecocci A, Parrino S, Peruzzi G & Pozzebon A. (2018). A Low Power IoT Sensor Node Architecture for Waste Management within Smart Cities Context. Sensors (Basel), 18(4), 1282, 1- 23.

[16] M. F., Omar, A.A. A., Termizi, D., Zainal, N. A., Wahap, N. M., Ismail & N., Ahmad (2016). Implementation of spatial smart waste management system in Malaysia. In IOP Conf. Series: Earth and Environmental Science, 37, 012059, Malaysia.

[17] Noor Salah & Rabee M.Hagem. (2018). Smart Recycle Bin System based on Wi-Fi and IoT. International Journal of Computer Applications, 181(4), 34-37.

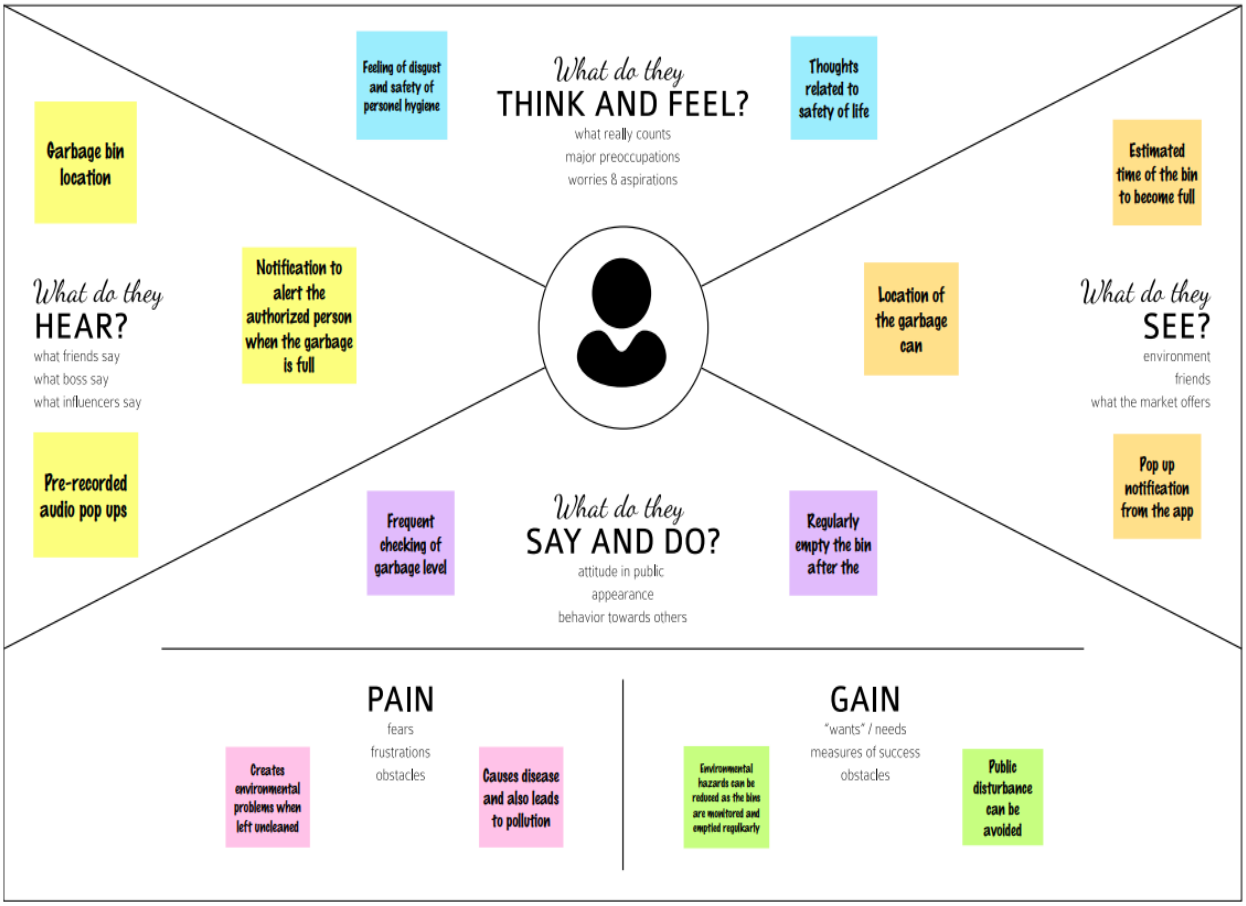
[18] Kellow Pardini, Joel J. P. C. Rodrigues, Sergei A. Kozlov, Neeraj Kumar & Vasco Furtado. (2019). IoTBased Solid Waste Management Solutions: A Survey. Journal of Sensor and Actuator Networks, 8(1).

2.3. ACKNOWLEDGMENT

Authors wish to thank Centre for Automation and Robotics, School of Mechanical Sciences, for the support and lab space provided to carry out this work under the Research Incubation Program, of Hindustan Institute Technology and Science, Padur, Chennai.

3. IDEATION & PROPOSED SOLUTION

3.1. Empathy Map Canvas



3.2. Ideation & brain storming

2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

RAM		VIVIN		ARUNACHALAM		MONESHWAR	
The sensor and actutator fixed in the garbage bin will sense the level of garbage in the bin	With the help of gps module in the bin send the location to the server	With the help of ultra-sonic sensor can measure the level of bin	People must have login to web-page	With the help of infra-red sensor can measure the level of bin	With the help of magnetic field of induction can seperate degradable and non-degradable	With the help of RFID technology can assign unique id to each bin	The synchronous work of sensor, microcontroller and web-page allows the system to work well in city or crowed place
If the bin is filled above 85% then the bin will send the notification to the server	The microcontroller in the bin must have network setting to connect to server	The truck should have automatic pick-up system	People can notify the officer incase the bin is fulled and truck does not arrive	With the help of load-cell sensor can measure the weight of bin	The data sensed by sensor in bin must reflect in municipality system	RFID tag in bin and RFID reader in truck to identify the bin	When the bin is full with the help of actutator can close the bin to avoid overflow
When the municipality staff see the notification in server they send location to the truck driver	with the help of microcontroller can send data to server	The truck should have automatic pick-up system	web-page must be user-friendly	must seperate eco-friendly and non eco-friendly waste	When truck driver is assigned with pick-up the driver detail must reflect in web-page along with pick-up location	With help of RFID and gps modules can send its location to server	This will help the garbage truck to reduce fuel and work
The truck should have the gps,tracking system to track the bin	The microcontroller must have storage facility to store the sensed data from the sensor	each bin should have unique-id	The sensor must be fitted perfectly so it does not get damaged incase of rough use	The web-page must show the level of waste in each bin	with the help of actuator the truck must pick-up the waste from bin automatically	The server must show each bin unique-id(RFID tag) in the web-page	The web-page must contain all the required information about the bin in the municipality surronding

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

Algorithm

Fix sensor and actutator in each bin and give each bin an unqiue-id

The device fixed in the bin must send the sensed data to cloud

The server has the data about the garbage in the bin and if the bin exceeds 85%, the server will send the notification to the truck.

The truck has GPS tracking system which allows it to track the bin's location and goes to the destination

After reaches the destination the actutator present in the truck will automatically pick-up the bin for unfill.

User interface and storage

Application is to be designed simple and easy to use

Using online cloud platform, we can store the sensed data from the bin

The application must contain the information of each bin present in the municipality.

Data and informations

Sensors like ultra-sonic, infrared can be used to sense the level of bin filled

Load cell can be used to measure the weight of the bin

RFID tag can be used to give each bin an unique-id

The GPS module in the bin must send the location to server

The tracking system must be present in the truck to locate the location of bin

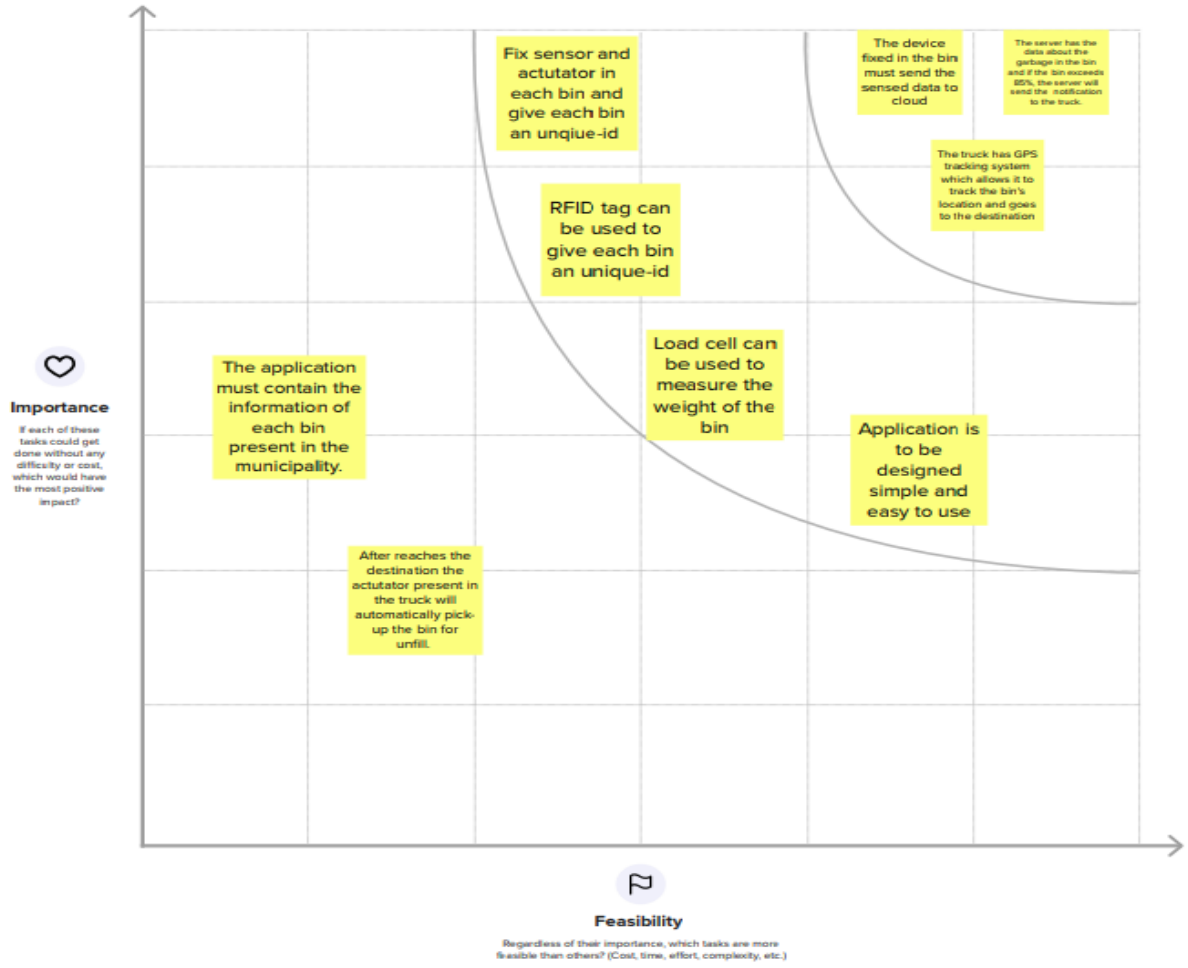
Micro-controller must be used to process the sensed data

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



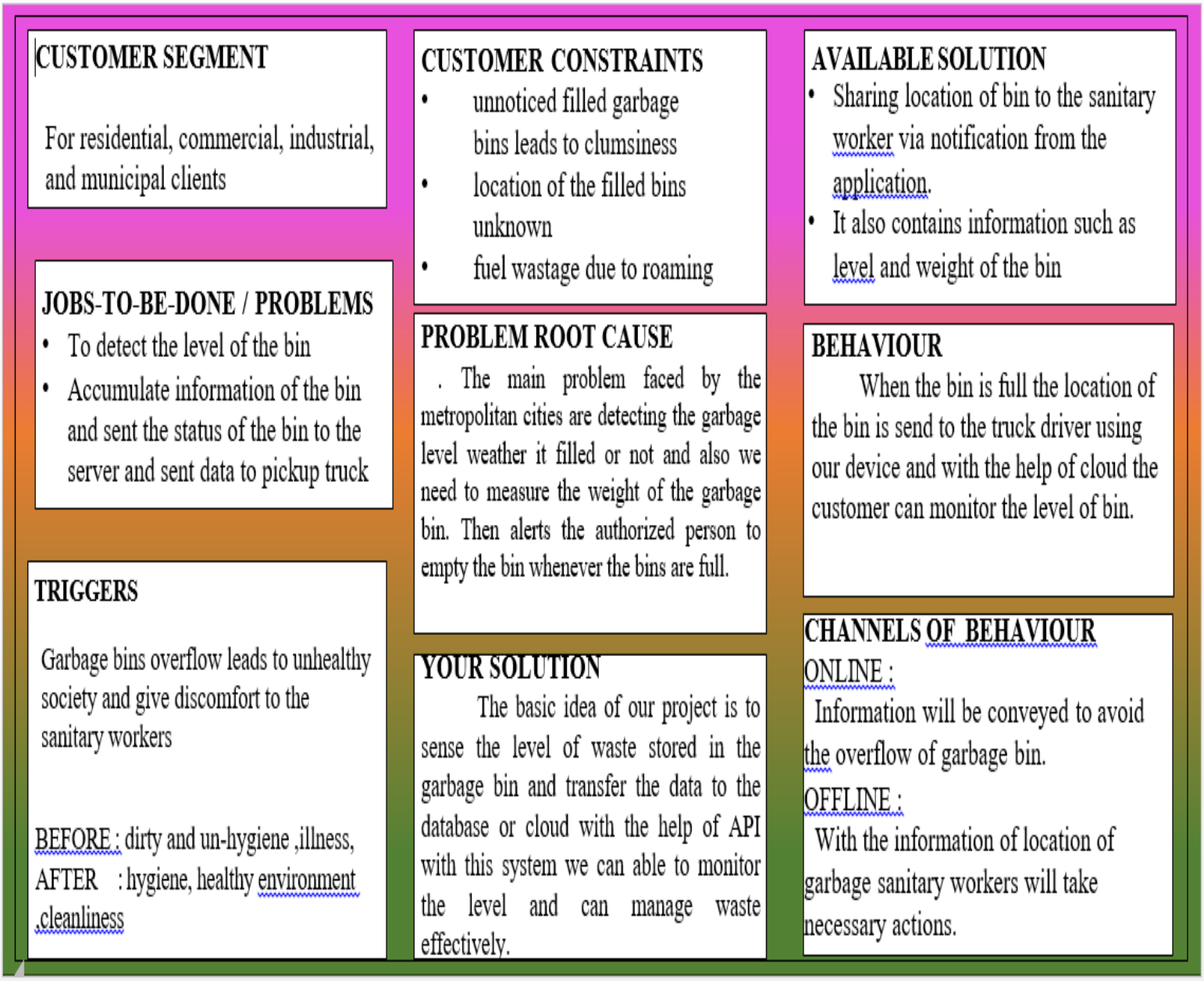
3.3. PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Waste management in metropolitan cities faces numerous challenges. The main problem faced by the metropolitan cities are detecting the garbage level weather it filled or not and also we need to measure the weight of the garbage bin. Then alerts the authorized person to empty the bin whenever the bins are full. We need to develop a web application to monitor the status of the bins remotely at anywhere. The application should provide the location of the every bin connected in the application with the help of global positioning system (GPS).The indication of the bins and the location of the every bin should be provided by web applications simultaneously.
2.	Idea / Solution description	Smart waste management system is the idea proposed to clean up the waste in the ecosystem.

		The basic idea of our project is to sense the level of waste stored in the garbage bin and transfer the data to the database or cloud with the help of API with this system we can able to monitor the level and can manage waste effectively.
3.	Novelty / Uniqueness	A reduction in the number of waste collections needed by up to 80%, resulting in less manpower, emissions, fuel use and traffic congestion.
4.	Social Impact / Customer Satisfaction	The social impact created by this system is that • it makes environment waste free and clean • it brings cleanliness and hygiene to the environment • it also save up unnecessary pick up and fuel usage • it modernizes waste management system
5.	Business Model (Revenue Model)	Revenue is generated by waste management and disposal services for residential, commercial, industrial, and municipal clients. The Company derives its revenue in the form of various fees

		associated with its maintenance charges and service offerings.
6.	Scalability of the Solution	Our system with modernized technology allows the municipality to upgrade the hardware and software whenever they feel to raise the standard of the system even more

3.3. PROBLEM SOLUTION FIT



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	Detailed bin inventory.	All monitored bins and stands can be seen on the map, and you can visit them at any time via the Street View feature from Google maps. Bins or stands are visible on the map as green, orange or red circles. You can see bin details in the Dashboard – capacity, waste type, last measurement, GPS location and Collection schedule or pick recognition.
FR-2	Real time bin monitoring.	The Dashboard displays real-time data on fill-levels of bins monitored by smart sensors. In addition to the % of fill-level, based on the historical data, the tool predicts when the bin will become full, one of the functionalities that are not included even in the best waste management software.. Sensors recognize picks as well; so you can check when the bin was last collected. With real-time data and predictions, you can eliminate the overflowing bins and stop collecting half-empty Ones.
FR-3	Expensive bins.	We help you identify bins that drive up your collection costs. The tool calculates a rating for each bin in terms of collection costs. The tool considers the average distance depo-bin-discharge in the area. The tool assigns bin a rating (1-10) and calculates distance from depo-bin discharge.
FR-4	Adjust bin distribution.	Ensure the most optimal distribution of bins. Identify areas with either dense or sparse bin distribution. Make sure all trash types are represented within a stand. Based on the historical data, you can adjust bin capacity Or location where necessary.
FR-5	Eliminate inefficient picks.	Eliminate the collection of half-empty bins. The sensors recognize picks. By using real-time data on fill-levels and pick recognition, we can show you how full the bins you Collect are.

		The report shows how full the bin was when picked. You immediately see any inefficient picks below 80% Full.
FR-6	Plan waste collection routes.	The tool semi-automates waste collection route planning. Based on current bin fill-levels and predictions of reaching full capacity, you are ready to respond and schedule waste collection. You can compare planned vs. executed routes to Identify any inconsistencies.

4.2. Non-functional Requirements:

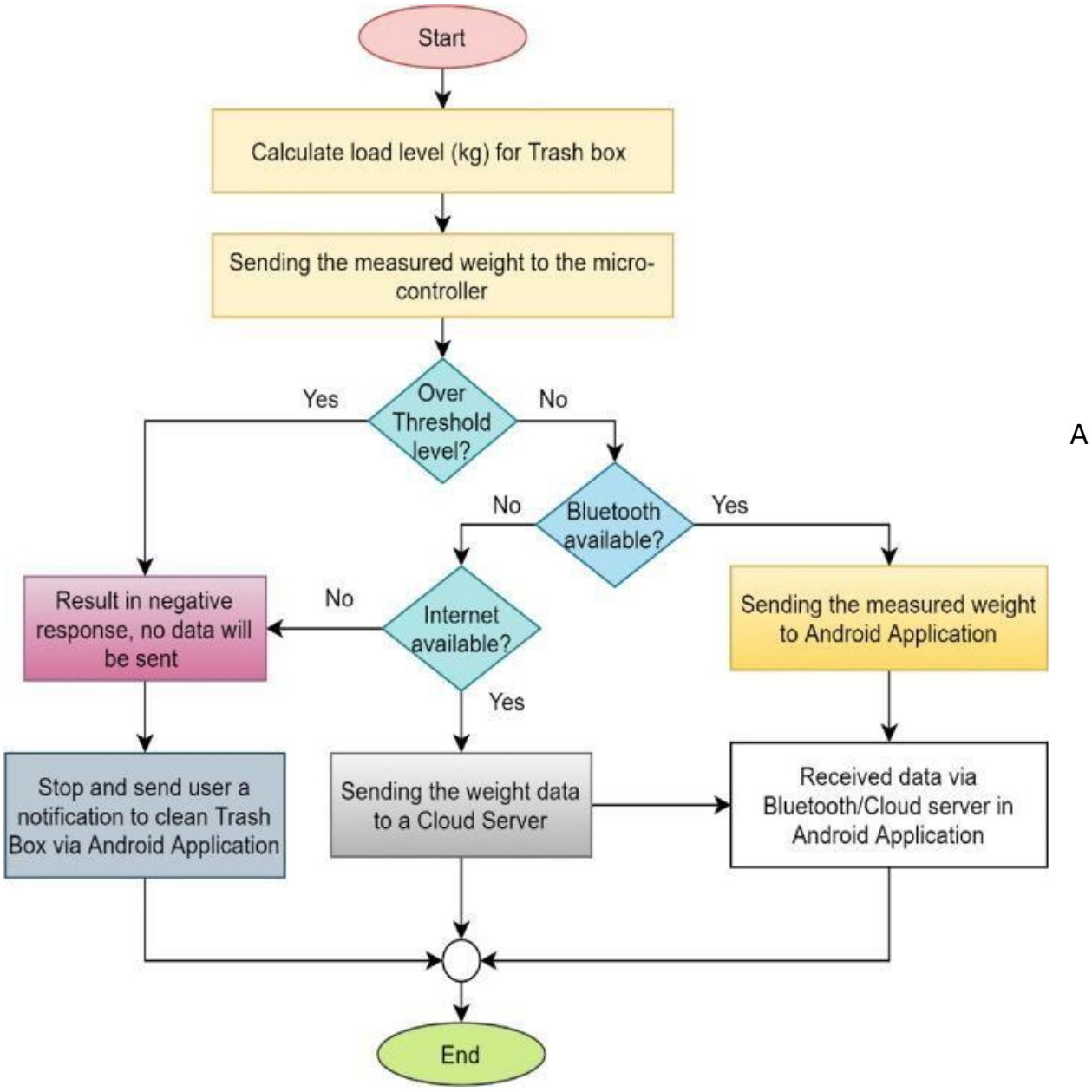
Following are the non-functional requirements of the proposed solution.

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 a 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 waste bins reduce the number of bins inside town , cities coz we able to monitor the

		garbage 24/7 more cost effect and scalability when we moves to smarter.

5. DESIGN FLOW

5.1. Data Flow Diagram:



A

Data Flow Diagram is a traditional visual representation of the information flows within a system.A neat and clear Data Flow Diagram can depict the right amount of the system requirement graphically.

It shows how data enters and leaves system, what changes the information andwhere the 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.

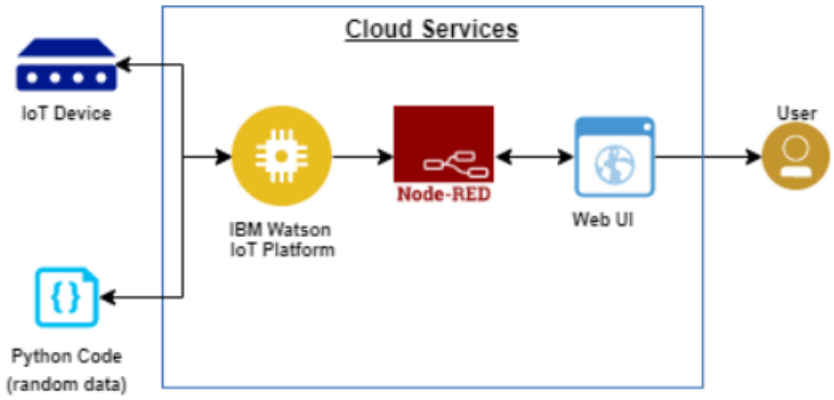
You can receive data on metric such as:

- The first test conducted is the situation where the garbage bin is empty or its garbage is very low

- Then, the bin is filled with more garbage until its level has surpassed the first threshold value, which is set to 80% then the first warning to cloud is being sent, as depicted
- The first notification to the cloud by the system is sent, once the waste reaches the level 85%full
- The second notification to cloud is sent by the system, indicating that bin is 95% full and the garbage needs to be collected immediately and the location is sent to the truck driver.
- 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
- The driving distance that could be saved

5.2. SOLUTION & TECHNICAL ARCHITECTURE

Project Description:



5.3. USER STORIES

5.3.1. Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Detailed bin inventory.	All monitored bins and stands can be seen on the map,and you can visit them at any time via the Street View feature from Google maps. Bins or stands are visible on the map as green, orange or red circles. You can see bin details in the Dashboard – capacity, waste type, last measurement, GPS location and collection schedule or pick recognition.
FR-2	Real time bin monitoring.	The Dashboard displays real-time data on fill-levels of bins monitored by smart sensors. In addition to the % of fill-level, based on the historical data, the tool predicts when the bin will become full, one of the functionalities that are not included even in the best waste management software.. Sensors recognize picks as well; so you can check when the bin was last collected. With real-time data and predictions, you can eliminate the overflowing bins and stop collecting half-empty ones.
FR-3	Expensive bins.	We help you identify bins that drive up your collection costs. The tool calculates a rating for each bin in termsof collection costs. The tool considers the average distance depo-bin-discharge in the area. The tool assigns bin a rating (1-10) and calculates distance from depo-bin discharge.
FR-4	Adjust bin distribution.	Ensure the most optimal distribution of bins. Identify areas with either dense or sparse bin distribution. Make sure all trash types are represented within a stand. Based on the historical data, you can adjust bin capacity or location where necessary.
FR-5	Eliminate inefficient picks.	Eliminate the collection of half-empty bins. The sensors recognize picks. By using real-time data on fill-levels and pick recognition, we can show you how full the bins you collect are.

		The report shows how full the bin was when picked. You immediately see any inefficient picks below 80% full.
FR-6	Plan waste collection routes.	The tool semi-automates waste collection route planning. Based on current bin fill-levels and predictionsof reaching full capacity, you are ready to respond and schedule waste collection. You can compare planned vs. executed routes to identify any inconsistencies.

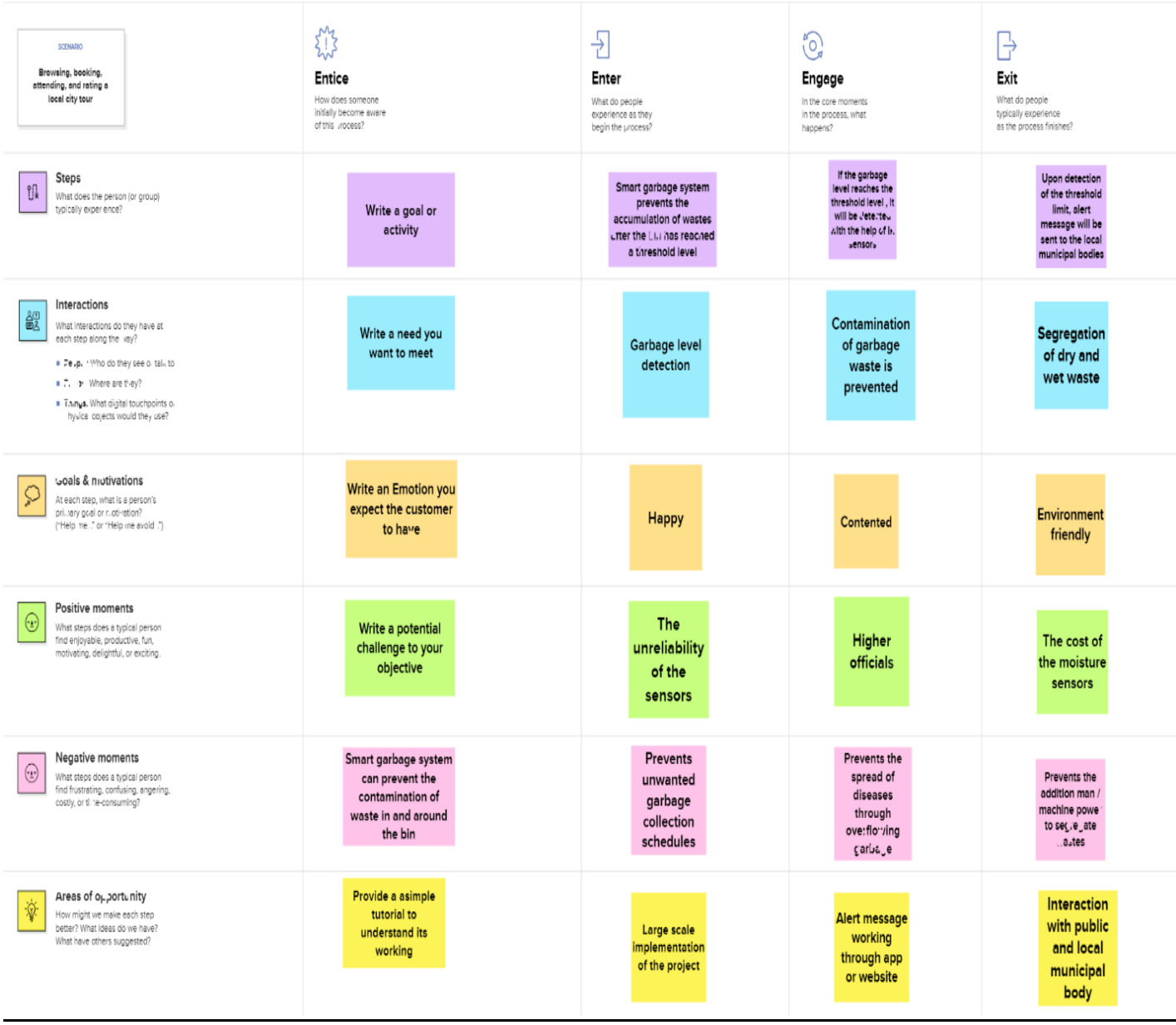
5.3.2. Non-functional Requirements:

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 a 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 waste bins reduce the number of bins inside town , cities coz we able to monitor the

		garbage 24/7 more cost effect and scalability when we moves to smarter.

5.3.3. CUSTOMER JOURNEY



6. PROJECT PLANNING AND SCHEDULING

6.1. SPRINT PLANNING & ESTIMATION

Sprint	Functional Requirement(Epic)	User Story Number	User Story/Task	Story Points	Priority	Team Members
Sprint-1	Login	USN-1	As a software developer, I will reduce the complexity of the website and make it easyto municipality offices to look the status of the bin's	20	High	Moneshwar
Sprint-2	Dashboard	USN-2	As a IOT developer, I will develop anintegration system of sensors and micro controller which sense and send the Information to the server/website.	20	High	Ramanathan
Sprint-3	Dashboard	USN-3	As a developer, I will collect the location of the filled bin and with the help of the GPS, I willsend the location to the server and the truck driver.	20	High	Vivin Rosan
Sprint-4	Dashboard	USN-4	As a testing engineer, I will test everyaspects of software and hardware implementations and sort out any errors.	20	High	Arunachalam

6.2. SPRINT DELIVERY SOLUTION

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date(Planned)	Story Points Completed (as on Planned End Date)	Sprint ReleaseDate (Actual)
Sprint-1	20	6Days	1Nov2022	5Nov2022	20	52022
Sprint-2	20	6Days	6Nov2022	10Nov2022	20	10Nov2022
Sprint-3	20	6Days	11Nov2022	15Nov2022	20	15Nov2022
Sprint-4	20	6Days	16Nov 2022	19Nov2022	20	19Nov2022

Velocity:

Imagine we have a 20-daysprint duration and the velocity of the team is 20 (points pers print).

Let's calculate the team's average velocity (AV) and periteration unit (story points per day).

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

7. CODING & SOLUTIONING

7.1. FEATURE 1

- **IOT DEVICE**
- **WOKWI SOFTWARE**
- **IOT WATSON PLATFORM**
- **NODE RED**
- **WEB UI**
- **CLOUDANT DB**

7.2. FEATURE 2

- **REGISTRATION**
- **LOGIN**
- **VERIFICATION**
- **SELECT THE CITY**
- **DISPLAY THE STATUS OF BIN**
- **ADD QUERY**

7.3. DATA BASE SCHEME

```
labl_0 = Label(base, text="Registration form",width=20,font=("bold",
20))
labl_0.place(x=90,y=53)
lb1= Label(base, text="Enter Name", width=10, font=("arial",12))
lb1.place(x=20, y=120)
en1= Entry(base) en1.place(x=200, y=120)
lb3= Label(base, text="Enter Email", width=10, font=("arial",12))
lb3.place(x=19, y=160)
en3= Entry(base) en3.place(x=200, y=160)
lb4= Label(base, text="Contact Number", width=13,font=("arial",12))
lb4.place(x=19, y=200)
en4= Entry(base) en4.place(x=200, y=200)
lb5= Label(base, text="Select Gender", width=15, font=("arial",12))
lb5.place(x=5, y=240)
var = IntVar()
Radiobutton(base, text="Male", padx=5,variable=var,
value=1).place(x=180, y=240)
Radiobutton(base, text="Female", padx =10,variable=var,
value=2).place(x=240,y=240)
Radiobutton(base, text="others", padx=15, variable=var,
value=3).place(x=310,y=240)
```

```
list_of_cntry = ("United States", "India", "Nepal", "Germany") cv =

StringVar()

drplist= OptionMenu(base, cv, *list_of_cntry) drplist.config(width=15)

cv.set("United States")

lb2= Label(base, text="Select Country", width=13,font=("arial",12))

lb2.place(x=14,y=280)

drplist.place(x=200, y=275)

lb6= Label(base, text="Enter Password", width=13,font=("arial",12))

lb6.place(x=19, y=320)

en6= Entry(base, show='*') en6.place(x=200, y=320)

lb7= Label(base, text="Re-Enter Password", width=15,font=("arial",12))

lb7.place(x=21, y=360)

en7 =Entry(base, show='*') en7.place(x=200, y=360)Button(base,

text="Register", width=10).place(x=200,y=400) base.mainloop()

 def generateOTP() :

# Declare a digits variable # which stores all digits digits = "0123456789"

OTP = ""

# length of password can be changed # by changing value in range

for i in range(4) :

OTP += digits[math.floor(random.random() * 10)] return OTP

# Driver code

if       name == "    main  " :
```

```
print("OTP of 4 digits:", generateOTP()) digits="0123456789"

OTP=""

for i in range(6): OTP+=digits[math.floor(random.random()*10)]

otp = OTP + " is your OTP" msg= otp

s = smtplib.SMTP('smtp.gmail.com', 587) s.starttls()

s.login("Your Gmail Account", "You app password") emailid =

input("Enter your email: ")

 s.sendmail('&&&&&&&&&&',emailid,msg) a = input("Enter Your OTP

>>: ")

if a == OTP: print("Verified")

else:

print("Please Check your OTP again")
```

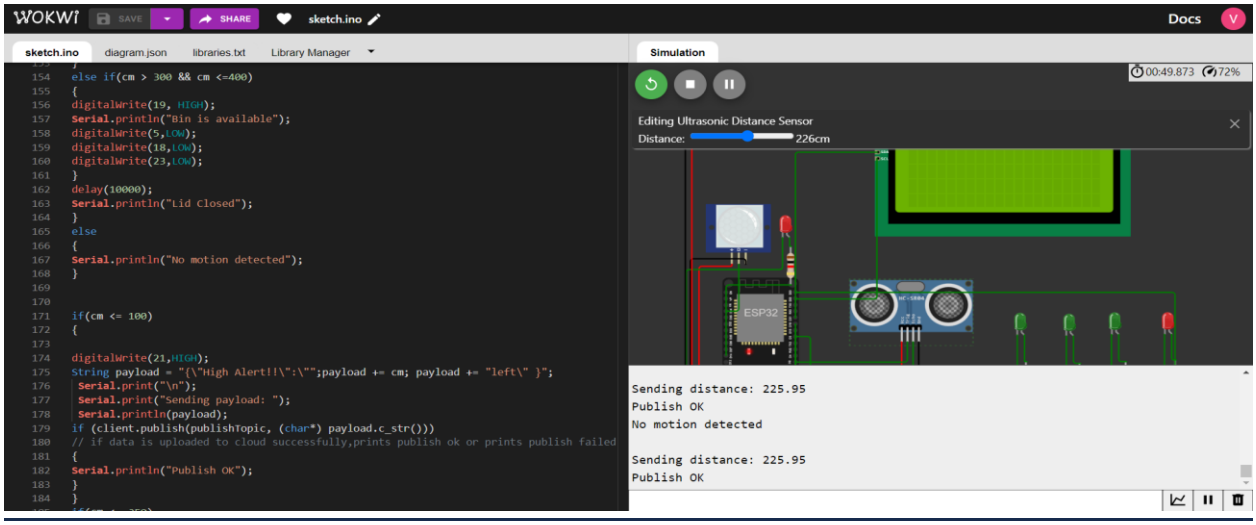
8. TESTING

S NO	TEST CASE	FEATURE	STEPS TO EXECUTE	EXPECTED RESULT	ACTUAL RESULT	EXECUTED BY
1	FUNCTIONAL	LOGIN	LOGIN TO EXECUTE BY FILLING THE DETAILS	CORRECT LOGIN CREDENTIALS	WORKING AS EXPECTED	MONESHWAR
2	FUNCTIONAL	REGISTRATION	REGISTRATION THROUGH FORMS	REGISTRATION FORM TO BE FILLED AND DISPLAYED	WORKING AS EXPECTED	MONESHWAR & RAMANATHAN
3	FUNCTIONAL	WOKWI	TO DEVELOP THE IOT DEVICE AND CODE THE IOT DEVICE	SENSE THE DATA	WORKING AS EXPECTED	RAMANATHAN
4	FUNCTIONAL	IBM WATSON	PUSH THE SENSED DATA FROM WOKWI	SENSED DATA IN IBM WATSON	WORKING AS EXPECTED	VIVIN ROSAN
5	FUNCTIONAL	NODE RED	TO CONNECT WITH THE IBM WATSON AND THEN COLLECT THE SENSED DATA AND DISPLAY IN NODE RED DASHBOARD	VISUAL REPRESENTATION OF SENSED DATA IN NODE RED DASHBOARD	WORKING AS EXPECTED	VIVIN ROSAN & ARUNACHALAM
6	TESTING	TEST THE ENTIRE WORK	TO CHECK ALL THE MENTIONED TESTCASE ARE WORKING PROPERLY	TEST CASE ARE WOKING PROPERLY	WORKING AS EXPECTED	ARUNACHALAM

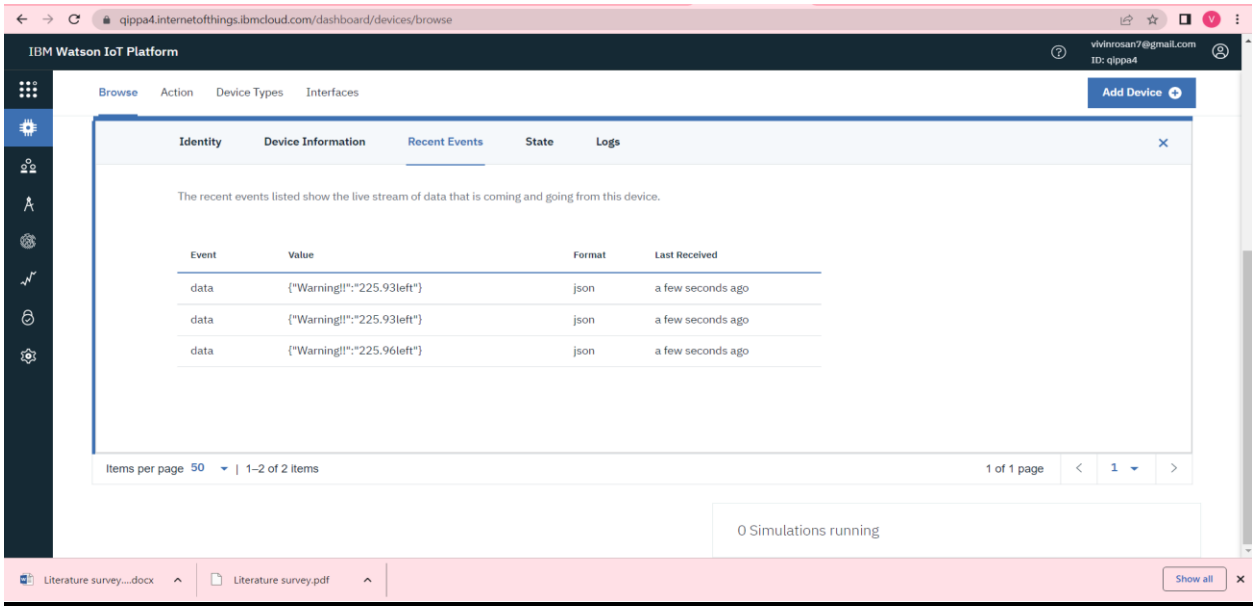
9. RESULT

9.1. PERFORMANCE METRICS

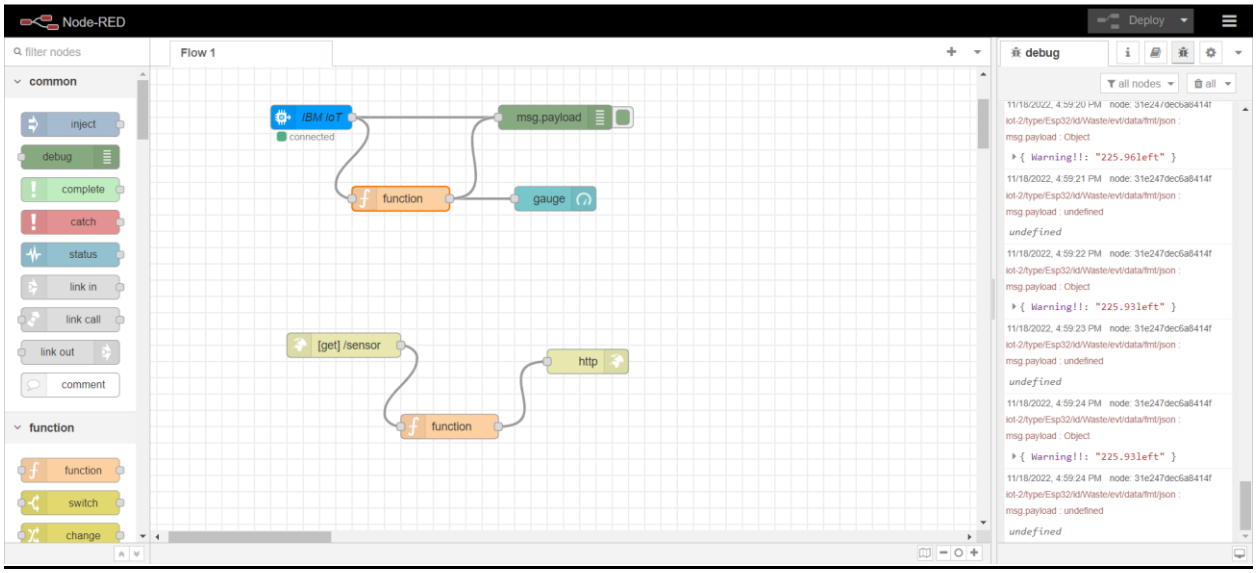
- **IOT DEVICE SIMULATION IN WOKWI SOFTWARE**



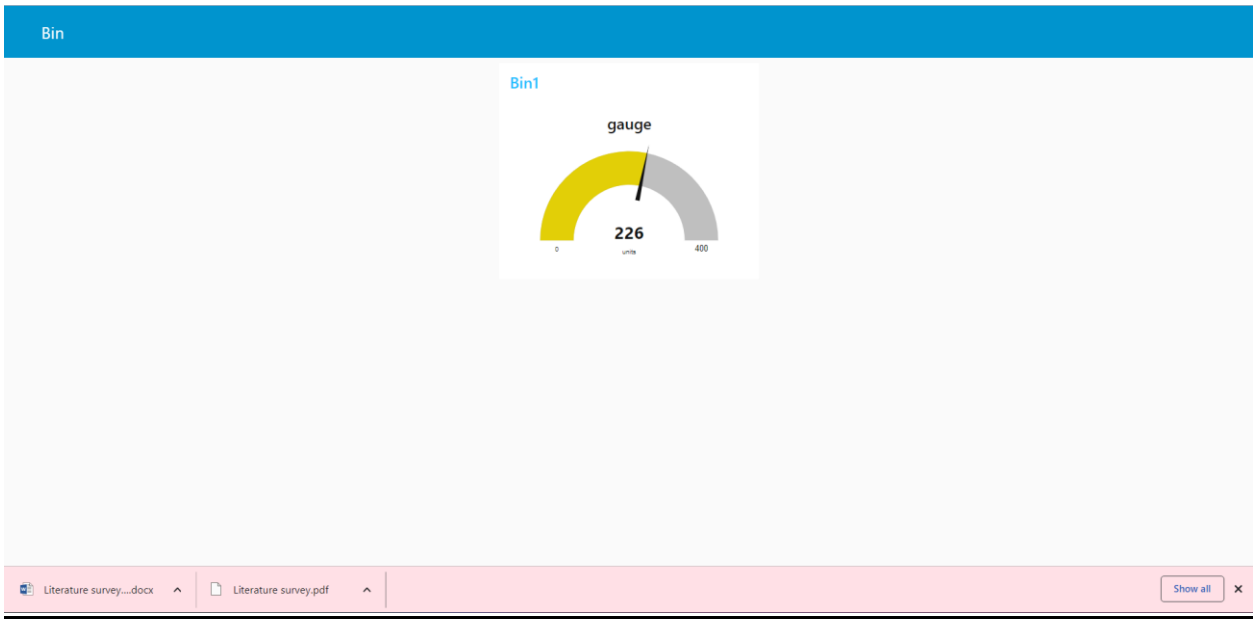
- **SENSED THE DATA VISUALIZATION IN IBM WATSON**



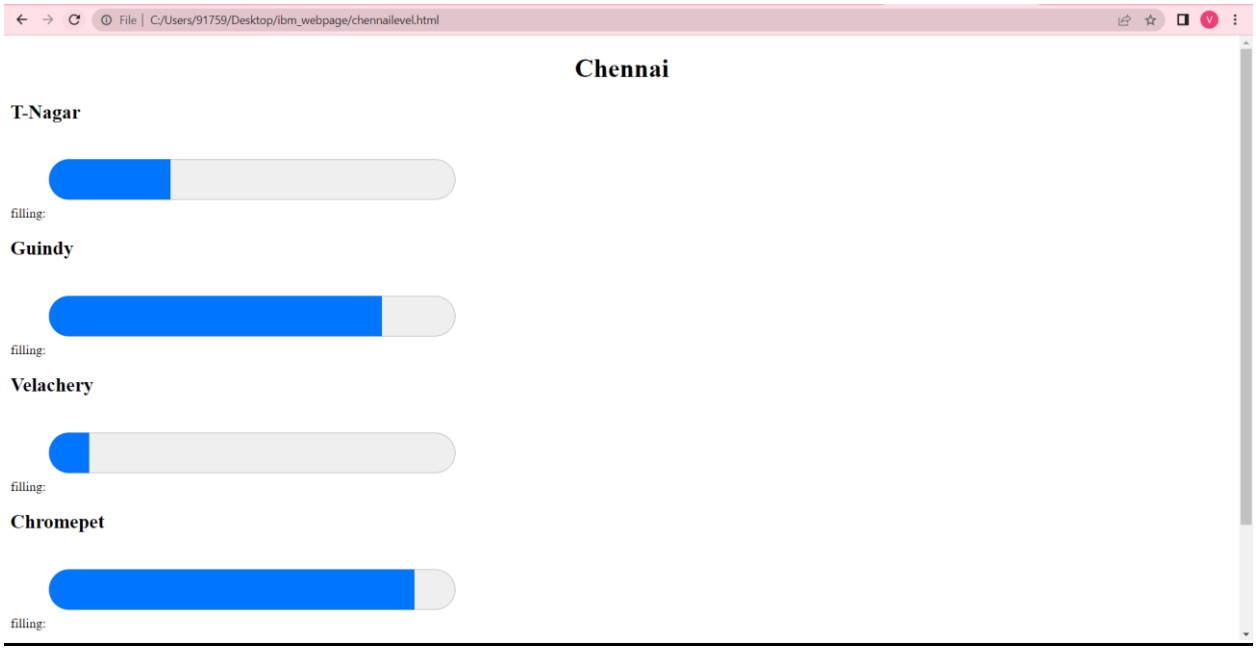
- **NODE RED CONNCETIONS**



- **VISUALIZATION OF SENSED DATA IN NODE RED DASHBOARD**



- **SAMPLE OF OUR WEBPAGE UI**



10. ADVANTAGES AND DISADVANTAGES

10.1. ADVANTAGES

- It saves time and money by using smart waste collection bins and systems equipped with fill level sensors. As smart transport vehicles go only to the filled containers or bins. It reduces infrastructure, operating and maintenance costs by up to 30%.
- It decreases traffic flow and consecutively noise due to less air pollution as result of less waste collection vehicles on the roads. This has become possible due to two way communication between smart dustbins and service operators.

- It keeps our surroundings clean and green and free from bad odour of wastes, emphasizes on healthy environment and keep cities more beautiful. It further reduces manpower requirements to handle the garbage collection process.
- Applying smart waste management process to the city optimizes management, resources and costs which makes it a "smart city".
- It helps administration to generate extra revenue by advertisements on smart devices.

10.2. DISADVANTAGES

- System requires more 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.
- Wireless technologies used in the system such as zigbee and Wi-Fi have shorter range and lower data speed.
- It reduces man power requirements which results into increase in unemployment for unskilled people.
- The training has to be provided to the people involved in the smart waste management system.

11. CONCLUSION

In most of the metro cities globally poses a challenge on effective waste solid waste management and maintenance of the waste bins. In this work an IOT enabled Smart Waste Bin with real time monitoring is designed and presented. In addition to the waste level measurement by using ultrasonic sensors, a sensing mechanism based on simple parallel plate capacitance is also developed and presented. Experimental investigations are carried out where the waste level of the smart bins is measured using the parallel plate capacitance and ultrasonic sensors and the statuses of the bins are communicated to the cloud effectively. The results prove the efficiency of the designed smart bins qualitatively. A smart waste management system incorporating robotic smart bins, where the smart bin has the mobility to move to the waste dockyard by localizing itself in the environment, is also proposed in this work. This system could find an application in smart buildings where the waste management could be practiced autonomously in a smarter way. Our future work is to investigate the performance of the proposed traditional and robotic waste management system in outdoor and indoor environment respectively in our Institutional campus.

12.FUTURE SCOPE:

New York has one of the more complicated waste management ecosystems in North America. It takes around 72 hundred waste collectors and a lot of infrastructure to keep the city of roughly 8.6 million people sanitary and clean. In Times Square alone, an estimated 500,000 pedestrians pass through on a daily basis, creating roughly 15,300 pounds of garbage. In March 2013, 30 Bigbelly smart waste and recycling stations were deployed in Times Square as part of the largest public space recycling initiative in New York City.

The Bigbelly units are equipped with waste compaction capabilities, real-time fill level monitoring, and collection notifications. With the Bigbelly smart stations, the total trash capacity was increased by nearly 200 percent and the frequency of collection per bin decreased by 50 percent. The program was such a success that the city expanded the deployment to 197 smart stations.

Future of waste management is one of the world's biggest challenges at the moment with the bulky, smelling waste that is being generated all over the world in quantities beyond the imaginable. Developed countries like Sweden and the U.S. somehow manage to dispose most of it and some gets even recycled or turned into compost.

In contrast, countries like India and China which are being overpopulated for decades now lack the resources and technology to effectively process all of the refuse that's been generated there. In fact, in 2010 China even imported 7.4 million tonnes of discarded plastic, 28 million tonnes of paper waste and over 50 million tonnes of discarded steel according to The Guardian.

To no one's amazement, this is still happening at even higher rates and in order to solve this global problem the world needs a waste management system of the future which will be more efficient in any aspect including: energy efficiency, volume of waste treatment and an increased use of robotics technology instead of people who suffer from diseases caused by the emitted pollutants. From Rubbish Begone – a rubbish removal London company we try to predict what the future of waste management would look like.

13. APPENDIX

13.1. Source code

- Code for IoT device in Wokwi

```
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2); // I2C address 0x3F, 16 column and 2 rows

int trigPin = 9;    // TRIG pin
int echoPin = 8;    // ECHO pin

float duration_us, distance_cm,distance;

void setup() {
  lcd.init();               // initialize the lcd
  lcd.backlight();
  pinMode(7,OUTPUT);
  pinMode(6,OUTPUT);
  pinMode(5,OUTPUT);
  pinMode(4,OUTPUT);       // open the backlight
  pinMode(trigPin, OUTPUT); // config trigger pin to output mode
  pinMode(echoPin, INPUT); // config echo pin to input mode
}

void loop() {
  // generate 10-microsecond pulse to TRIG pin
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // measure duration of pulse from ECHO pin
  duration_us = pulseIn(echoPin, HIGH);

  // calculate the distance
  distance_cm = 0.017 * duration_us;
  distance=400-distance_cm;
  lcd.clear();
```

```
    lcd.setCursor(0, 0); // start to print at the first row
    lcd.print("waste level: ");
    lcd.print(distance);
    digitalWrite(6,HIGH);
    digitalWrite(7,LOW);
    digitalWrite(5,LOW);
    digitalWrite(4,LOW);
    if(distance>=175)
    {
      digitalWrite(5,HIGH);
      digitalWrite(6,LOW);
      digitalWrite(7,LOW);
      digitalWrite(4,LOW);
    }
    if(distance>=275)
    {
      digitalWrite(4,HIGH);
      digitalWrite(6,LOW);
      digitalWrite(5,LOW);
      digitalWrite(7,LOW);
    }
    if(distance>=375)
    {
      digitalWrite(7,HIGH);
      digitalWrite(6,LOW);
      digitalWrite(5,LOW);
      digitalWrite(4,LOW);
    }

    delay(500);
  }
```

- Code for connecting to IoT Watson

```
#include <LiquidCrystal_I2C.h>
#include <WiFi.h>
#include <PubSubClient.h>
#include<WiFiClient.h>
LiquidCrystal_I2C lcd(0x27, 20, 4); // I2C address 0x3F, 16 column and 2 rows

int trigPin = 2;    // TRIG pin
int echoPin = 15;    // ECHO pin

#define ORG "qippa4"
#define DEVICE_TYPE "Esp32"
#define DEVICE_ID "Waste"
#define TOKEN "C72(GeQy)UPSVtHdUw"
char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // server name
char publishTopic[] = "iot-2/evt/data/fmt/json";
char topic[] = "iot-2/cmd/led/fmt/String"; // cmd Represent type and command is
test format of strings
char authMethod[] = "use-token-auth"; // authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //Client id

WiFiClient wifiClient; // creating instance for wifi
PubSubClient client(server, 1883, wifiClient);

void setup() {
  lcd.init();               // initialize the lcd
  lcd.backlight();
  pinMode(5,OUTPUT);
  pinMode(18,OUTPUT);
  pinMode(19,OUTPUT);
  pinMode(23,OUTPUT);
  pinMode(34,INPUT);
  pinMode(14,OUTPUT);
  // open the backlight
  pinMode(trigPin, OUTPUT); // config trigger pin to output mode
  pinMode(echoPin, INPUT);
  Serial.begin(115200);
  wifiConnect();
  mqttConnect();
```

```
 // config echo pin to input mode
}

float readcmCM()
{
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
int duration = pulseIn(echoPin, HIGH);
return duration * 0.034 / 2;
}

void loop()
{

lcd.clear();

publishData();
delay(500);
if (!client.loop())
{
mqttConnect();  // function call to connect to IBM
}
}

/*  -retrieving to cloud  */

void wifiConnect()
{
Serial.print("Connecting to "); Serial.print("Wifi");

WiFi.begin("Wokwi-GUEST", "", 6);
while (WiFi.status() != WL_CONNECTED)
{
delay(500);
Serial.print(".");
}
Serial.print("WiFi connected, IP address: ");
Serial.println(WiFi.localIP());
}
void mqttConnect()
{
if (!client.connected())
```

```
{
Serial.print("Reconnecting MQTT client to ");
Serial.println(server);
while (!client.connect(clientId, authMethod, token))
{
Serial.print(".");
delay(500);
}
initManagedDevice();
Serial.println();
}
}
void initManagedDevice()
{
if (client.subscribe(topic))
{
Serial.println("IBM subscribe to cmd OK");
}
else
{
Serial.println("subscribe to cmd FAILED");
}
}
void publishData()
{
float cm = readcmCM();

if(digitalRead(34))
{
Serial.println("Motion Detected"); Serial.println("Lid Opened"); digitalWrite(14,
HIGH);

}
else
{
digitalWrite(14, LOW);
} //PIR motion detection



if(digitalRead(34))
{
if(cm <= 100)
//Bin level detection
{
```

```
digitalWrite(23, HIGH);
Serial.println("High Alert!!!,Trash bin is about to be full");
Serial.println("Lid Closed");
lcd.print("Full! Don't use");
delay(2000);
lcd.clear();
digitalWrite(18,LOW);
digitalWrite(19,LOW);
digitalWrite(5,LOW);
}
else if(cm > 100 && cm < 200)
{
digitalWrite(5, HIGH);
Serial.println("Warning!!,Trash is about to cross 75% of bin level");
digitalWrite(18,LOW);
digitalWrite(19,LOW);
digitalWrite(23,LOW);
}
else if(cm > 200 && cm < 300)
{
digitalWrite(18, HIGH);
Serial.println("Warning!!,Trash is about to cross 50% of bin level");
digitalWrite(5,LOW);
digitalWrite(19,LOW);
digitalWrite(23,LOW);
}
else if(cm > 300 && cm <=400)
{
digitalWrite(19, HIGH);
Serial.println("Bin is available");
digitalWrite(5,LOW);
digitalWrite(18,LOW);
digitalWrite(23,LOW);
}
delay(10000);
Serial.println("Lid Closed");
}
else
{
Serial.println("No motion detected");
}


if(cm <= 100)
{
```

```
digitalWrite(21,HIGH);
String payload = "{\"High Alert!!\":\"";payload += cm; payload += "left\" }";
 Serial.print("\n");
 Serial.print("Sending payload: ");
 Serial.println(payload);
if (client.publish(publishTopic, (char*) payload.c_str()))
// if data is uploaded to cloud successfully,prints publish ok or prints publish
failed
{
Serial.println("Publish OK");
}
}
if(cm <= 250)
{
digitalWrite(22,HIGH);
 String payload = "{\"Warning!!\":\"";payload+= cm; payload += "left\" }";
Serial.print("\n");
Serial.print("Sending distance: ");
 Serial.println(cm);
if(client.publish(publishTopic, (char*) payload.c_str()))
{
Serial.println("Publish OK");
}
else
{
Serial.println("Publish FAILED");
}
}

float inches = (cm / 2.54); //print on LCD
lcd.setCursor(0,0);
lcd.print("Inches");
 lcd.setCursor(4,0);
lcd.setCursor(12,0);
lcd.print("cm"); lcd.setCursor(1,1);
 lcd.print(inches, 1);
lcd.setCursor(11,1);
lcd.print(cm,1);
lcd.setCursor(14,1);
delay(1000);
 lcd.clear();
}
```

- Code for login page

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {font-family: Arial, Helvetica, sans-serif;}

/* Full-width input fields */
input[type=text], input[type=password] {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  display: inline-block;
  border: 1px solid #ccc;
  box-sizing: border-box;
}

/* Set a style for all buttons */
button {
  background-color: #04AA6D;
  color: white;
  padding: 14px 20px;
  margin: 8px 0;
  border: none;
  cursor: pointer;
  width: 100%;
}

button:hover {
  opacity: 0.8;
}

/* Extra styles for the cancel button */
.cancelbtn {
  width: auto;
  padding: 10px 18px;
  background-color: #f44336;
}
```

```css
/* Center the image and position the close button */
.imgcontainer {
  text-align: center;
  margin: 24px 0 12px 0;
  position: relative;
}

img.avatar {
  width: 40%;
  border-radius: 10%;
}

.container {
  padding: 16px;
}

span.psw {
  float: right;
  padding-top: 16px;
}

/* The Modal (background) */
.modal {
  display: none; /* Hidden by default */
  position: fixed; /* Stay in place */
  z-index: 1; /* Sit on top */
  left: 0;
  top: 0;
  width: 100%; /* Full width */
  height: 100%; /* Full height */
  overflow: auto; /* Enable scroll if needed */
  background-color: rgb(0,0,0); /* Fallback color */
  background-color: rgba(0,0,0,0.4); /* Black w/ opacity */
  padding-top: 60px;
}

/* Modal Content/Box */
.modal-content {
  background-color: #fefefe;
  margin: 5% auto 15% auto; /* 5% from the top, 15% from the bottom and centered
*/
  border: 1px solid #888;
  width: 80%; /* Could be more or less, depending on screen size */
}
```

```
/* The Close Button (x) */
.close {
  position: absolute;
  right: 25px;
  top: 0;
  color: #000;
  font-size: 35px;
  font-weight: bold;
}

.close:hover,
.close:focus {
  color: red;
  cursor: pointer;
}

/* Add Zoom Animation */
.animate {
  -webkit-animation: animatezoom 0.6s;
  animation: animatezoom 0.6s
}

@-webkit-keyframes animatezoom {
  from {-webkit-transform: scale(0)}
  to {-webkit-transform: scale(1)}
}

@keyframes animatezoom {
  from {transform: scale(0)}
  to {transform: scale(1)}
}

/* Change styles for span and cancel button on extra small screens */
@media screen and (max-width: 300px) {
  span.psw {
     display: block;
     float: none;
  }
  .cancelbtn {
     width: 100%;
  }

}
</style>
</head>
```

```
<body style="text-align: center;">

<h1 style="padding-top: 200px; text-align: center;">Smart Waste Management System
For Metropolitan Cities</h1>

<button  onclick="document.getElementById('id01').style.display='block'"
style="width:auto; ">Login</button>

<div id="id01" class="modal">

  <form  class="modal-content animate" method="post">
    <div class="imgcontainer">
      <span onclick="document.getElementById('id01').style.display='none'"
class="close" title="Close Modal">&times;</span>
      <img src="goLogo.jpg" alt="Avatar" class="avatar">
    </div>

    <div class="container">
      <label for="uname"><b>Username</b></label>
      <input id="frm1" type="text" placeholder="Enter Username" name="uname"
required>

      <label for="psw"><b>Password</b></label>
      <input type="password" placeholder="Enter Password" name="psw" required>


      <button style="color: black"
onclick="window.location.href=('district.html')" type="submit">signin</button>
      <label>
        <input type="checkbox" checked="checked" name="remember"> Remember me
      </label>
    </div>

    <div class="container" style="background-color:#f1f1f1">
      <button type="button"
onclick="document.getElementById('id01').style.display='none'"
class="cancelbtn">Cancel</button>
      <span class="psw">Forgot <a href="#">password?</a></span>
    </div>
  </form>
</div>
<p style="text-align: center;font-size: 10px; color: #04AA6D;">Reuse, Recycle,
and Reduce the waste for a better future !</p>
```

```
<script>

function myFunction() {
  var x = document.getElementById("frm1");
  var text = "";
  var i;
  for (i = 0; i < x.length ;i++) {
    text += x.elements[i].value + "<br>";
  }
  document.getElementById("demo").innerHTML = text;

}
// Get the modal action_page.php
var modal = document.getElementById('id01');

// When the user clicks anywhere outside of the modal, close it
window.onclick = function(event) {
    if (event.target == modal) {
        modal.style.display = "none";
    }
}
</script>

</body>
</html>
```

- Code for district select

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {font-family: Arial, Helvetica, sans-serif;}

/* Full-width input fields */
input[type=text], input[type=password] {
  width: 100%;
  padding: 12px 20px;
```

```
  margin: 8px 0;
  display: inline-block;
  border: 1px solid #ccc;
  box-sizing: border-box;
}

/* Set a style for all buttons */
button {
  background-color: #04AA6D;
  color: white;
  padding: 14px 20px;
  margin: 8px 0;
  border: none;
  cursor: pointer;
  width: 100%;
}

button:hover {
  opacity: 0.8;
}

/* Extra styles for the cancel button */
.cancelbtn {
  width: auto;
  padding: 10px 18px;
  background-color: #f44336;
}

/* Center the image and position the close button */
.imgcontainer {
  text-align: center;
  margin: 24px 0 12px 0;
  position: relative;
}

img.avatar {
  width: 40%;
  border-radius: 10%;
}

.container {
  padding: 16px;
}

span.psw {
```

```
  float: right;
  padding-top: 16px;
}

/* The Modal (background) */
.modal {
  display: none; /* Hidden by default */
  position: fixed; /* Stay in place */
  z-index: 1; /* Sit on top */
  left: 0;
  top: 0;
  width: 100%; /* Full width */
  height: 100%; /* Full height */
  overflow: auto; /* Enable scroll if needed */
  background-color: rgb(0,0,0); /* Fallback color */
  background-color: rgba(0,0,0,0.4); /* Black w/ opacity */
  padding-top: 60px;
}

/* Modal Content/Box */
.modal-content {
  background-color: #fefefe;
  margin: 5% auto 15% auto; /* 5% from the top, 15% from the bottom and centered
*/
  border: 1px solid #888;
  width: 80%; /* Could be more or less, depending on screen size */
}

/* The Close Button (x) */
.close {
  position: absolute;
  right: 25px;
  top: 0;
  color: #000;
  font-size: 35px;
  font-weight: bold;
}

.close:hover,
.close:focus {
  color: red;
  cursor: pointer;
}

/* Add Zoom Animation */
```

```
.animate {
  -webkit-animation: animatezoom 0.6s;
  animation: animatezoom 0.6s
}

@-webkit-keyframes animatezoom {
  from {-webkit-transform: scale(0)}
  to {-webkit-transform: scale(1)}
}

@keyframes animatezoom {
  from {transform: scale(0)}
  to {transform: scale(1)}
}

/* Change styles for span and cancel button on extra small screens */
@media screen and (max-width: 300px) {
  span.psw {
     display: block;
     float: none;
  }
  .cancelbtn {
     width: 100%;
  }

}
</style>
</head>
<body style="text-align: center;">

<h1 style="padding-top: 200px; text-align: center;">Smart Waste Management System
For Metropolitan Cities</h1>

<button  onclick="document.getElementById('id01').style.display='block'"
style="width:auto; ">Login</button>

<div id="id01" class="modal">

  <form  class="modal-content animate" method="post">
    <div class="imgcontainer">
      <span onclick="document.getElementById('id01').style.display='none'"
class="close" title="Close Modal">&times;</span>
      <img src="goLogo.jpg" alt="Avatar" class="avatar">
    </div>
```

```
    <div class="container">
      <label for="uname"><b>Username</b></label>
      <input id="frm1" type="text" placeholder="Enter Username" name="uname"
required>

      <label for="psw"><b>Password</b></label>
      <input type="password" placeholder="Enter Password" name="psw" required>


      <button style="color: black"
onclick="window.location.href=('district.html')" type="submit">signin</button>
      <label>
        <input type="checkbox" checked="checked" name="remember"> Remember me
      </label>
    </div>

    <div class="container" style="background-color:#f1f1f1">
      <button type="button"
onclick="document.getElementById('id01').style.display='none'"
class="cancelbtn">Cancel</button>
      <span class="psw">Forgot <a href="#">password?</a></span>
    </div>
  </form>
</div>
<p style="text-align: center;font-size: 10px; color: #04AA6D;">Reuse, Recycle,
and Reduce the waste for a better future !</p>


<script>

function myFunction() {
  var x = document.getElementById("frm1");
  var text = "";
  var i;
  for (i = 0; i < x.length ;i++) {
    text += x.elements[i].value + "<br>";
  }
  document.getElementById("demo").innerHTML = text;

}
// Get the modal action_page.php
var modal = document.getElementById('id01');

// When the user clicks anywhere outside of the modal, close it
window.onclick = function(event) {
```

```
    if (event.target == modal) {
        modal.style.display = "none";
    }
}
</script>

</body>
</html>
```

- Code of Coimbatore district

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {font-family: Arial, Helvetica, sans-serif;}

/* Full-width input fields */
input[type=text], input[type=password] {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  display: inline-block;
  border: 1px solid #ccc;
  box-sizing: border-box;
}

/* Set a style for all buttons */
button {
  background-color: #04AA6D;
  color: white;
  padding: 14px 20px;
  margin: 8px 0;
  border: none;
  cursor: pointer;
  width: 100%;
}
```

```
button:hover {
  opacity: 0.8;
}

/* Extra styles for the cancel button */
.cancelbtn {
  width: auto;
  padding: 10px 18px;
  background-color: #f44336;
}

/* Center the image and position the close button */
.imgcontainer {
  text-align: center;
  margin: 24px 0 12px 0;
  position: relative;
}

img.avatar {
  width: 40%;
  border-radius: 10%;
}

.container {
  padding: 16px;
}

span.psw {
  float: right;
  padding-top: 16px;
}

/* The Modal (background) */
.modal {
  display: none; /* Hidden by default */
  position: fixed; /* Stay in place */
  z-index: 1; /* Sit on top */
  left: 0;
  top: 0;
  width: 100%; /* Full width */
  height: 100%; /* Full height */
  overflow: auto; /* Enable scroll if needed */
  background-color: rgb(0,0,0); /* Fallback color */
  background-color: rgba(0,0,0,0.4); /* Black w/ opacity */
```

```
  padding-top: 60px;
}

/* Modal Content/Box */
.modal-content {
  background-color: #fefefe;
  margin: 5% auto 15% auto; /* 5% from the top, 15% from the bottom and centered
*/
  border: 1px solid #888;
  width: 80%; /* Could be more or less, depending on screen size */
}

/* The Close Button (x) */
.close {
  position: absolute;
  right: 25px;
  top: 0;
  color: #000;
  font-size: 35px;
  font-weight: bold;
}

.close:hover,
.close:focus {
  color: red;
  cursor: pointer;
}

/* Add Zoom Animation */
.animate {
  -webkit-animation: animatezoom 0.6s;
  animation: animatezoom 0.6s
}

@-webkit-keyframes animatezoom {
  from {-webkit-transform: scale(0)}
  to {-webkit-transform: scale(1)}
}

@keyframes animatezoom {
  from {transform: scale(0)}
  to {transform: scale(1)}
}

/* Change styles for span and cancel button on extra small screens */
```

```
@media screen and (max-width: 300px) {
  span.psw {
     display: block;
     float: none;
  }
  .cancelbtn {
     width: 100%;
  }

}
</style>
</head>
<body style="text-align: center;">

<h1 style="padding-top: 200px; text-align: center;">Smart Waste Management System
For Metropolitan Cities</h1>

<button  onclick="document.getElementById('id01').style.display='block'"
style="width:auto; ">Login</button>

<div id="id01" class="modal">

  <form  class="modal-content animate" method="post">
    <div class="imgcontainer">
      <span onclick="document.getElementById('id01').style.display='none'"
class="close" title="Close Modal">&times;</span>
      <img src="goLogo.jpg" alt="Avatar" class="avatar">
    </div>

    <div class="container">
      <label for="uname"><b>Username</b></label>
      <input id="frm1" type="text" placeholder="Enter Username" name="uname"
required>

      <label for="psw"><b>Password</b></label>
      <input type="password" placeholder="Enter Password" name="psw" required>


      <button style="color: black"
onclick="window.location.href=('district.html')" type="submit">signin</button>
      <label>
        <input type="checkbox" checked="checked" name="remember"> Remember me
      </label>
    </div>
```

```
    <div class="container" style="background-color:#f1f1f1">
      <button type="button"
onclick="document.getElementById('id01').style.display='none'"
class="cancelbtn">Cancel</button>
      <span class="psw">Forgot <a href="#">password?</a></span>
    </div>
  </form>
</div>
<p style="text-align: center;font-size: 10px; color: #04AA6D;">Reuse, Recycle,
and Reduce the waste for a better future !</p>


<script>

function myFunction() {
  var x = document.getElementById("frm1");
  var text = "";
  var i;
  for (i = 0; i < x.length ;i++) {
    text += x.elements[i].value + "<br>";
  }
  document.getElementById("demo").innerHTML = text;

}
// Get the modal action_page.php
var modal = document.getElementById('id01');

// When the user clicks anywhere outside of the modal, close it
window.onclick = function(event) {
    if (event.target == modal) {
        modal.style.display = "none";
    }
}
</script>

</body>
</html>
```

- Code for Chennai district

```
<!DOCTYPE html>
<html>
    <head>
        <title>Chennai</title>
    </head>
    <body >

        <h1 style="text-align: center;"><b>Chennai</b></h1>
    <h2>T-Nagar</h2>
    <label for="file">filling:</label>
<progress style="width: 500px ; height: 100px;" id="file" value="30" max="100">
70% </progress>
<h2>Guindy</h2>
<label for="file">filling:</label>
<progress style="width: 500px ; height: 100px;" id="file" value="82" max="100">
42% </progress>

<h2>Velachery</h2>
<label for="file">filling:</label>
<progress style="width: 500px ; height: 100px;"  id="file" value="10" max="100">
100% </progress>
<h2>Chromepet</h2>
<label for="file">filling:</label>
<progress style="width: 500px ; height: 100px;" id="file" value="90" max="100">
10% </progress>
<h2>Avadi</h2>

<label for="file">filling:</label>
<progress style=" width: 500px ;height: 100px;" id="file" value="52" max="100">
92% </progress>

    </body>
</html>
```

- Code for Madurai district

```
<!DOCTYPE html>
<html>
    <head>
        <title>Madurai</title>
    </head>
    <body>
        <h1 style="width:auto; text-align: center"><b>Madurai</b></h1>
    <h2>Usilampatti</h2>
    <label for="file">filling:</label>
<progress style="width: 500px ; height: 100px;" id="file" value="70" max="100">
70% </progress>
<h2>Peraiyur</h2>
<label for="file">filling:</label>
<progress style="width: 500px ; height: 100px;" id="file" value="42" max="100">
42% </progress>

<h2>Papawadi</h2>
<label for="file">filled:</label>
<progress style="width: 500px ; height: 100px;"  id="file" value="100" max="100">
100% </progress>
<h2>Vadipatti</h2>
<label for="file">filling:</label>
<progress style="width: 500px ; height: 100px;" id="file" value="10" max="100">
10% </progress>
<h2>Mulipuram</h2>

<label for="file">filled:</label>
<progress style=" width: 500px ;height: 100px;" id="file" value="100" max="100">
92% </progress>
    </body>
</html>
```

- Code for Dindugal district

```
<!DOCTYPE html>
<html>
    <head>
        <title>Dindigul</title>
    </head>
    <body>
        <h1 style="width:auto; text-align: center"><b>Dindigul</b></h1>
    <h2>Palani</h2>
    <label for="file">filled:</label>
<progress style="width: 500px ; height: 100px;" id="file" value="100" max="100">
70% </progress>
<h2>Chinnalapatti</h2>
<label for="file">filling:</label>
<progress style="width: 500px ; height: 100px;" id="file" value="42" max="100">
42% </progress>

<h2>Ottanchatram</h2>
<label for="file">filled:</label>
<progress style="width: 500px ; height: 100px;"  id="file" value="70" max="100">
100% </progress>
<h2>Virupachi</h2>
<label for="file">filling:</label>
<progress style="width: 500px ; height: 100px;" id="file" value="10" max="100">
10% </progress>
<h2>Natham</h2>

<label for="file">filling:</label>
<progress style=" width: 500px ;height: 100px;" id="file" value="92" max="100">
92% </progress>
    </body>
</html>
```

13.2. GIT HUB LINK

https://github.com/IBM-EPBL/IBM-Project-6563-1658831915

13.3. WORKING MODEL VIDEO LINK

https://github.com/IBM-EPBL/IBM-Project-6563-1658831915/blob/main/Final%20deliverables/Smart%20waste%20Final%20video.mp4