

PROJECT REPORT

1.INTRODUCTION:

1.1.Project Overview

1.2.Purpose

2.LITRATURE SURVEY:

2.1.Existing problem

2.2.References

2.3.Problem Statement Definition

3. IDEATION & PROPOSED SOLUTION

3.1. Empathy Map Canvas

3.2. Ideation & Brainstorming

3.3. Proposed Solution

3.4. Problem Solution fit

4. REQUIREMENT ANALYSIS

4.1. Functional requirement

4.2. Non-Functional requirements

5. PROJECT DESIGN

5.1. Data Flow Diagrams

5.2. Solution & Technical Architecture

5.3. User Stories

6. PROJECT PLANNING & SCHEDULING

6.1. Sprint Planning & Estimation

6.2. Sprint Delivery Schedule

6.3. Reports from JIRA

7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1. Feature 1

7.2. Feature 2

7.3. Database Schema (if Applicable)

8. TESTING

8.1. Test Cases

8.2. User Acceptance Testing

9. RESULTS

9.1. Performance Metrics

10. ADVANTAGES & DISADVANTAGES

11. CONCLUSION

12. FUTURE SCOPE

13. APPENDIX

13.1. Links

Source Code

GitHub & Project Demo Link

Project Report

| | |
|---------------------|---|
| Team ID | PNT2022TMID46943 |
| Project Name | Smart Waste Management System For Metropolitan Cities |

1.INTRODUCTION:

1.1.Project Overview:

As the population is increasing, solid waste is also increasing in urban and rural areas, and waste management has become a global concern. A certain number of employees need to be appointed to attend to a certain number of dustbins. This is done every day periodically. This leads to a very inefficient and unclean system in which some dumpsters will be overflowing while others might not be even half full. This is caused by variation in population density in the city or some other random factor. This makes it impossible to determine which part needs immediate attention. Here, a waste management system is introduced in which each dumpster is embedded in a monitoring system that will notify the corresponding person if the dumpster is full. In this system, it is also possible to separate wet and dry waste into two separate containers. This system provides an effective solution to the waste management problem.

The garbage produced in the residential area can be collected directly from homes or by it is making an arrangement for mass collection in that area and can be lifted using vehicles. In the case of restaurants, malls, and other commercial establishments, garbage can be collected directly from the unit using vehicles. Industrial garbage, which includes waste produced in construction sites and various industries, can also be disposed of in different ways. For effective handling of these wastes, like collection and disposal, the Internet of Things (IOT) concept is being used, which mainly deals with sensing, actuating, data gathering, storing, and processing by connecting physical and virtual devices to the Internet.

1.2.Purpose:

A waste management system is the strategy an organization uses to dispose, reduce, reuse, and prevent waste. Possible waste disposal methods are recycling, composting, incineration, landfills, bioremediation, waste to energy, and waste minimization. As for waste management, it is the measures utilized to manage waste in its entire life cycle, from waste generation to disposal or recovery.

2.LITRATURE SURVEY:

2.1.Existing Problem:

Manual systems in which employees clear the dumpsters periodically.No systematicapproach towards clearing the dumpsters.unclear about the status of a particular location.Employees are unaware of the need for a particular location.much less effective in cleaning the city. The fill level of solid waste in each of the containers, which are strategically situated across the communities, is detected using ultrasonic sensors. A Wireless Fidelity (Wi-Fi) communication link is used to transmit the sensor data to an IoT cloud platform known as ThingSpeak. Depending on the fill level, the system sends appropriate notification message (in form of tweet) to alert relevant authorities and concerned citizen(s) for necessary action. Also, the fill level is monitored on ThingSpeak in real-time. The system performance shows that the proposed solution may be found useful for efficient waste management in smart and connected communities.

2.2.Reference:

1. Teemu Nuortioa, Jari KytoÄjokib, Harri Niskaa, Olli BraÄysyb Improved route planning and scheduling of waste collection and transport, Expert Systems with Applications 30 (2006 223 232,
2. M. Arebey, M. Hannan, H. Basri, and H. Abdullah, "Solid waste monitoring and management using RFID, GIS and GSM", The IEEE Student Conference on Research and Development (SCOReD), 16-18 November 2009, UPM Serdang, Malaysia, 2009

3. M. Hannan, M. Arebey, R. A. Begum, and H. Basri, "Radio Frequency Identification (RFID) and communication technologies for solid waste bin and truck monitoring system", Waste Management, Vol. 31, pp. 2406-2413, 2011.
4. S. Longhi, D. Marzioni, E. Alidori, G. Di Buo, M. Prist, M. Grisostomi, et al., "Solid Waste Management Architecture Using Wireless Sensor Network Technology", The 5th International Conference on New Technologies, Mobility and Security (NTMS), 7-10 May 2012, Istanbul, pp. 1-5, 2012. 147.
5. Waikhom Reshmi, RamKumar Sundaram, M. Rajeev Kumar, Sensor Unit for Waste Management: A Better Method,, International conference on Science, Engineering and Management Research, ©2014 IEEE.

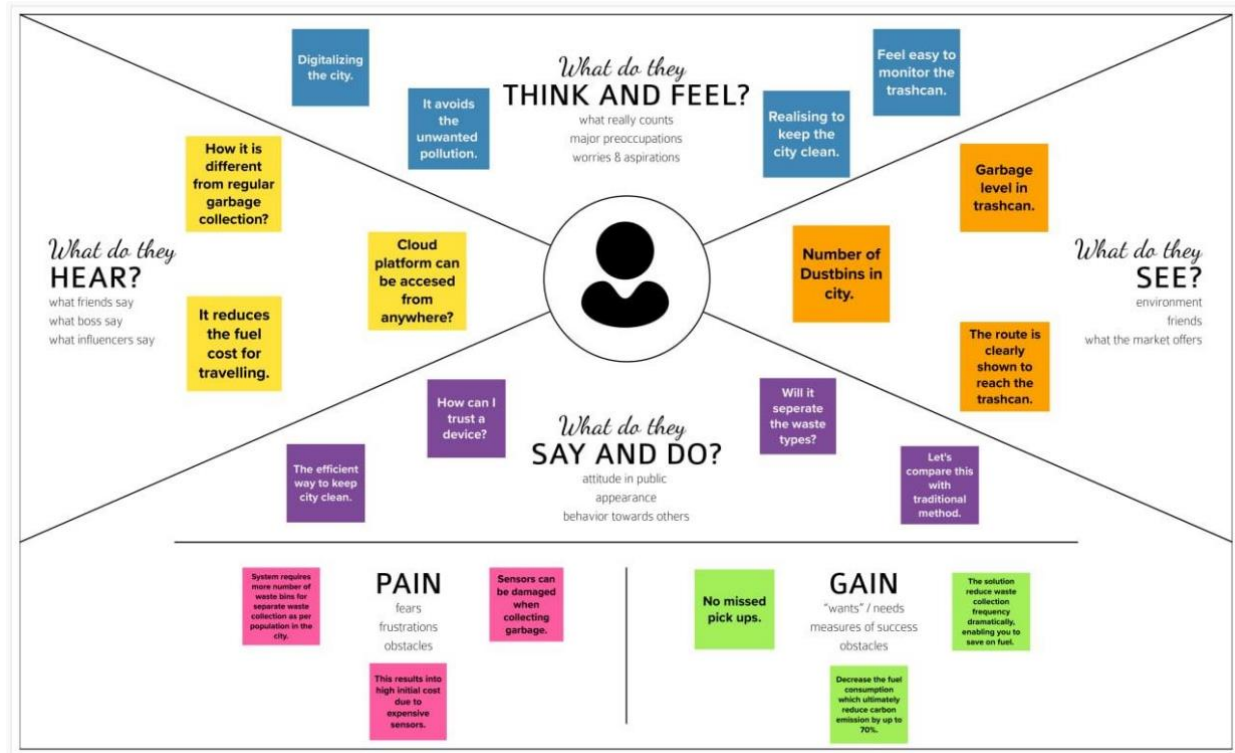
2.3.Problem Statement Definition:

| User Story Number | User Story / Task | Acceptance criteria | Priority |
|-------------------|---|---------------------------------|----------|
| USN-1 | As an admin, I can monitor every dustbin and its garbage levels. | I can monitor the system. | High. |
| USN-2 | As an admin, I will inform the authorized person to empty the trashcan. | I can inform authorized person. | Medium. |
| USN-3 | As an admin, I can notice the trash level of every dustbin. | I can notice the trash level. | Low. |
| USN-4 | As a Co-Admin, I can send alert message to the truck drivers. | I can alert truck driver. | High. |

| | | | |
|-------|---|--|---------|
| USN-5 | As a trash van driver, I will follow the route to the dustbin. | I can reach the filled trashcans. | High. |
| USN-6 | As a waste collector, I will collect all the trash from the dumpsters and load it to the truck. | I can empty the trashcans. | Medium. |
| USN-7 | As a municipality officer, I can supervise the process and ensure the cleanliness of city. | I can manage all these process going good. | High. |

3. IDEATION & PROPOSED SOLUTION:


3.1. Empathy Map Canvas:



3.2. Ideation & Brainstorming:




Step-1: Team Gathering, Collaboration and Select the Problem Statement:

Template




Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.


 10 minutes to prepare
 1 hour to collaborate
 2-8 people recommended


[Share template feedback](#)





Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.


 10 minutes

**Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

**Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.


**Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.


[Open article](#) →





Define your problem statement


What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.


 5 minutes


**PROBLEM**
How might we enhance the metropolitan city of smart waste management system?


**Key rules of brainstorming**
To run a smooth and productive session


 Stay on topic.

 Encourage wild ideas.

 Defer judgment.

 Listen to others.

 Go for volume.

 If possible, be visual.

Step-2: Brainstorm, Idea Listing and Grouping:

2 Brainstorm

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

10 minutes

TIP

You can switch to sticky notes and hit the panel2 switch to switch back to stick drawing!

J.JUSTIN RAJ

- Use a smart display to show the status of the trashcan
- Use a smart display to show the status of the trashcan
- Use a smart display to show the status of the trashcan

D.KARAN

- Use a smart display to show the status of the trashcan
- Use a smart display to show the status of the trashcan
- Use a smart display to show the status of the trashcan

W.MERUN RAJ

- Use a smart display to show the status of the trashcan
- Use a smart display to show the status of the trashcan
- Use a smart display to show the status of the trashcan

WESTEPHEN RAJ

- Use a smart display to show the status of the trashcan
- Use a smart display to show the status of the trashcan
- Use a smart display to show the status of the trashcan

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 can break it up into smaller sub-groups.

20 minutes

TIP

Add customizable tags to sticky notes to make it easier to find, remove, organize, and categorize important ideas as themes within your mural.

Indulge with recycle bin ultra eye waste news, detect and integrated with smart IoT app.

When the trashcan smells bad, the already food fragrance releases the gas for good smelling.

Add customizable tags to sticky notes to make it easier to find, remove, organize, and categorize important ideas as themes within your mural.

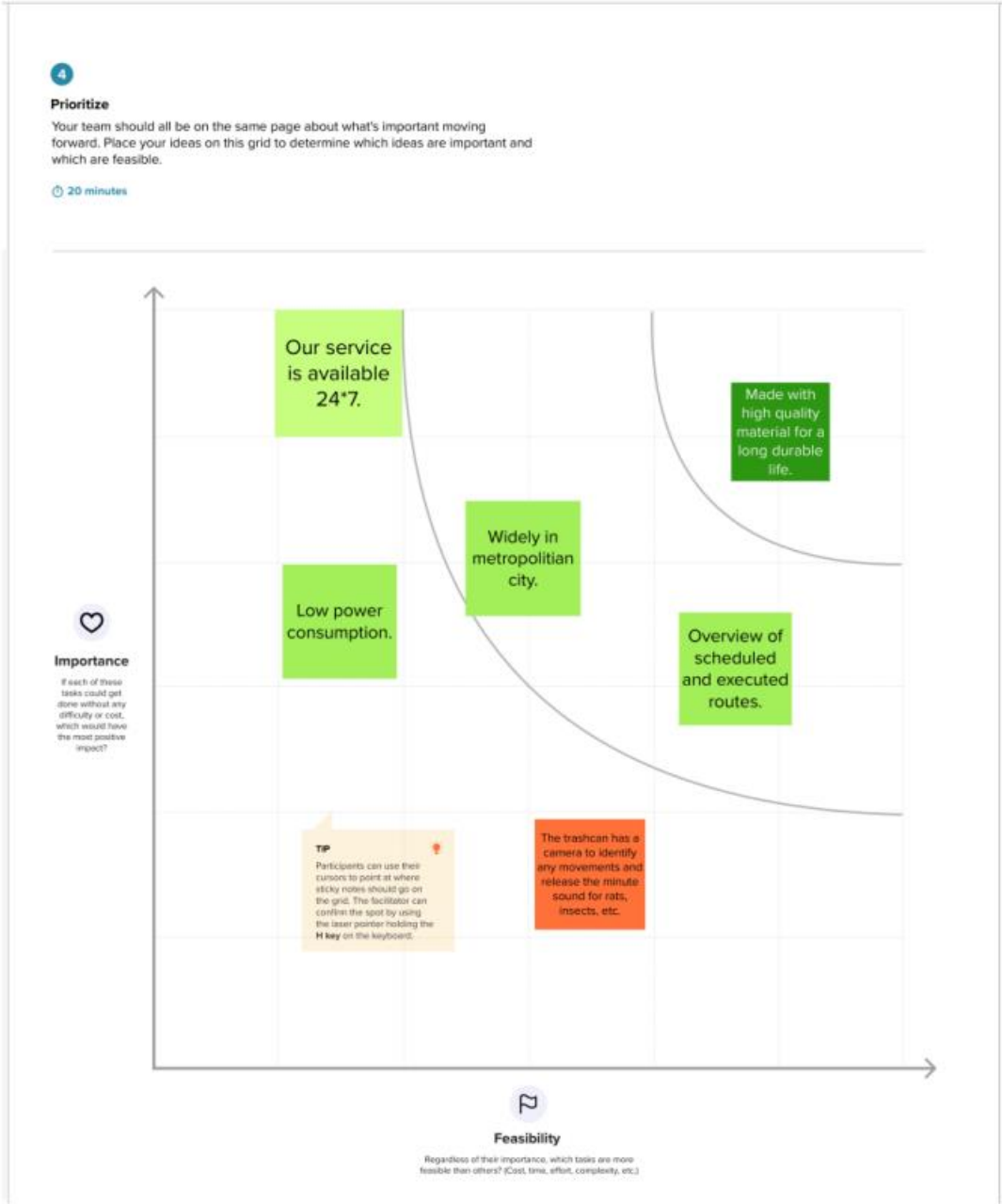
Paper waste, plastic waste, food waste and other community generated waste.

Smart indicator on the front display panel.

UVC technology kills germs and bacteria inside the bin.

The trashcan has a camera to identify any movements and release the impulse sound for rats, insects, etc.

Step-3: Idea Prioritization:



3.3.Proposed Solution:

| S.No. | Parameter | Description |
|-------|---|---|
| 1 | Problem Statement (Problem to be solved). | At present, solid waste management is a major concern in the metropolitan cities of the developing and developed countries. As the population is growing, the garbage is also increasing. This huge, unmanaged accumulation of garbage is polluting the environment, spoiling the beauty of the area and also posing a health hazard. The dumpsters often overflow and make the city unclean. |
| 2 | Idea / Solution description. | A system is introduced to manage waste in big cities effectively without having to monitor the parts 24/7 manually. Here, the problem of unorganised and non-systematic waste |

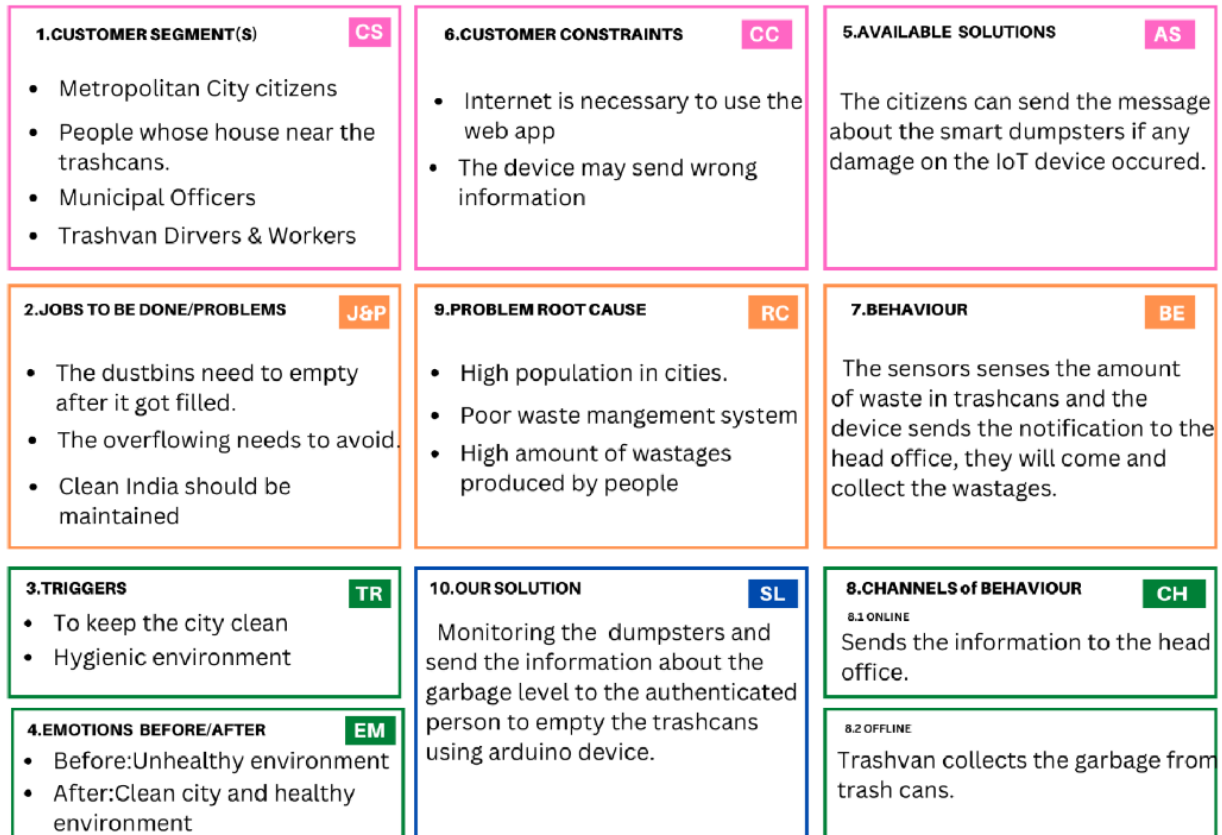
| | | |
|---|-----------------------|--|
| | | collection is solved by designing an embedded IoT system that will monitor each dumpster individually for the amount of waste deposited. The IR sensor is used for detecting the presence of any waste the IR sensor used. The device is connected to the cloud. Whenever the bin gets filled, the message will be sent to the municipal office. |
| 3 | Novelty / Uniqueness. | The problem of unorganized and nonsystematic waste collection is solved by designing an embedded IoT system that will monitor each dumpster individually for the amount of waste deposited. The IR sensor is used for detecting the presence of any debris the IR sensor used. The device is connected to the cloud. Whenever the bin gets filled, the message will be sent to the municipal office. |

| | | |
|---|--|--|
| 4 | Social Impact / Customer Satisfaction. | <p>This project is very effective in managing waste in any big city. Rather than using conventional periodic collection methods, a priority system is used to ensure the city is clean all the time without any overflowing dumpsters. It has been tested and verified properly to ensure all the different parts work together for a smooth function of the whole system. In most of the metro cities globally poses a challenge to effective waste management and maintenance of the waste bins.</p> |
| 5 | Business Model (Revenue Model). | <ul style="list-style-type: none"> • The cost to develop the project is about the sensors used here. • The Arduino device and Cloud platform used here play a vital role in cost. • If any damage occurs to the device during pick-ups of the trashcan we |

| | | |
|---|------------------------------|---|
| | | <p>need to fix it.</p> <ul style="list-style-type: none"> • The contribution of the municipality is necessary to make the project succeed in the market |
| 6 | Scalability of the Solution. | <p>The project design is a part of the implication that can be used to improve the waste management of a locality. All the technical aspects have been thoroughly designed keeping all the constraints in mind. The project resolves around whether the project will be able to meet the future needs of the users. This project-based on IoT gives users the freedom of changing hardware as well as software specifications as per the arising need. IoT based projects are already designed while keeping future demands in mind and in a rising economy like India where the concept of smart cities is new the demand for our project will keep on</p> |

| | | |
|--|--|-------------|
| | | increasing. |
|--|--|-------------|

3.4.Problem Solution fit:



4. REQUIREMENT ANALYSIS:

4.1. Functional requirement:

| FR No. | Functional Requirement (Epic) | Sub Requirement (Story / Sub-Task) |
|--------|--------------------------------------|--|
| FR-1 | Fitting IoT device in the trashcans. | The IoT device need to be fixed in the dustbin with water proof safety. The IoT device consists Ultrasonic sensor, IR sensor, Weight sensor. To send data to the cloud GSM/GPRS is used. |
| FR-2 | Connecting to the cloud. | The device should configure to connect to |

| | | |
|------|------------------------------|--|
| | | the cloud. The data of sensors need to be received and processed. |
| FR-3 | Predictions for bin fulness. | In this system, a 24×7 monitoring system is designed for monitoring dumpsters, A smart and organized system is designed for selective clearing the ultrasonic sensor is used for measuring the level of waste in the dustbin, DC motor powered platform is used for segregating wet and dry waste, IR sensor and moisture sensor is used for separating wet and dry waste. If either of the containers is full then an alert message is sent from the dustbin to employees and the cloud. In turn, employees can clear the corresponding dumpster. |
| FR-4 | Real-time waste monitoring | Trash and recycling containers can be outfitted or produced with low-cost sensors that monitor everything from the amount and types of material in a container to temperature, odour and |

| | | |
|------|-------------------------|---|
| | | location of the bin. |
| FR-5 | Do not miss a pick | For periodically picked bins, we provide Pick evaluation. The tool records picks (sensor) and compares them to the schedule. Authorized person can immediately identify any missed, or off-schedule picks. |
| FR-6 | Routes to the dumpsters | Based on current bin fill-levels and predictions of reaching full capacity, you are ready to respond and schedule waste collection. driver can compare planned vs. executed routes to identify any inconsistencies. |

4.2.Non-Functional requirements:

| FR No. | Non-Functional Requirement | Description |
|--------|----------------------------|---|
| NFR-1 | Usability | IoT solutions for waste management problems offer municipalities data intelligence and realtime insights. In that regard, |

| | | |
|-------|-------------|--|
| | | <p>the fill patterns of specific containers can be identified by historical data and managed accordingly in the long term. In addition to hardware solutions, mobile applications are used to overcome the challenges in the regular waste management system, such as keeping track of the drivers while they are operating on the field</p> |
| NFR-2 | Security | <p>Building and deploying IoT-based smart waste management in cities can be a complex, timeconsuming and resource-intensive process. Many municipal IT departments will not have the resources or in-house skills to support such a project internally.</p> |
| NFR-3 | Reliability | <p>One of the difficult operational problems of municipal and local authorities are facing is the collection of municipal solid waste. In recent years, due to environmental concerns</p> |

| | | |
|-------|--------------|---|
| | | <p>and number of costs, most of the municipalities have been forced for assessing their solid waste management and examining their costeffectiveness and environmental impact, for example, designing the collection of routes. During the past 15 years.</p> |
| NFR-4 | Performance | <p>An integrated Arduino program is developed to synchronize the identification system, automated lid system, micro-controller, display system, and communication system. An ultrasonic sensor is attached to the front side of the garbage bin. The transmitter of the ultrasonic sensor emits an ultrasonic sound that is beyond the human ear listening range, and the receiver receives the reflected sound waves by the solid objects.</p> |
| NFR-5 | Availability | <p>Another purpose of this project is to make the proposed waste management system as</p> |

| | | |
|---------|-------------|--|
| | | cheap as possible. A cost in BDT is presented in the following Table 3 needs for the construction of the proposed smart bin. |
| NFR – 6 | Scalability | The city diverts about 80% of its waste from landfills and hopes to go “zero waste” by the end of 2020. Besides strict regulations and high waste management fees for end consumers and businesses . |

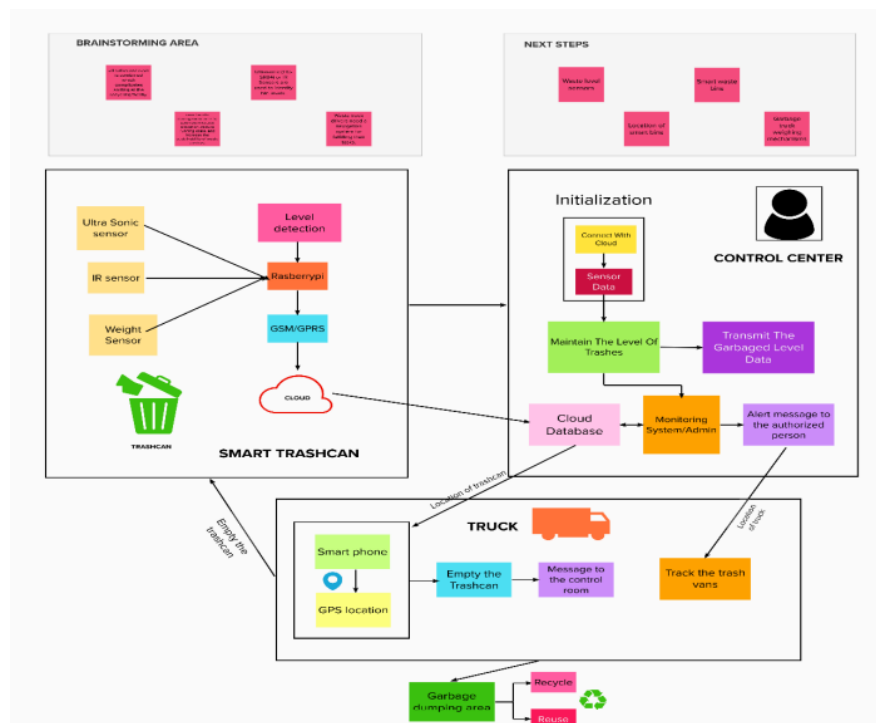
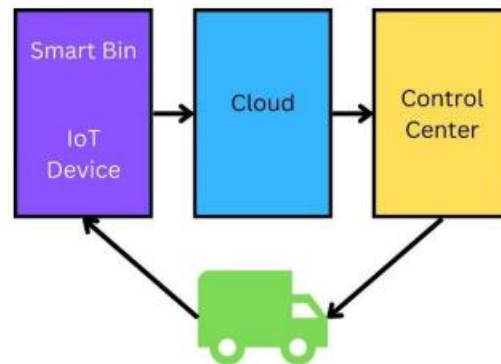
5.PROJECT DESIGN:

5.1.Data Flow Diagrams:

The IoT device is fitted in the trashcans.

- The sensors in the device senses the garbage level.
- The GSM/GPRS will send the information about the garbage level to the cloud.
- The admin in the control center notifies the authorized person to collect the garbage.
- The truck driver will be notified the route to the filled dumpsters.
- The trashes are loaded to the truck.
- The more number of bins needed in high populated area.
- The overflowing of trashcans can be avoided.
- No missed pickups of trashcans.
- New smart dustbins can be install by just connecting the IoT device to the cloud.

5.2.Data flow diagram:



5.2.Solution & Technical Architecture:

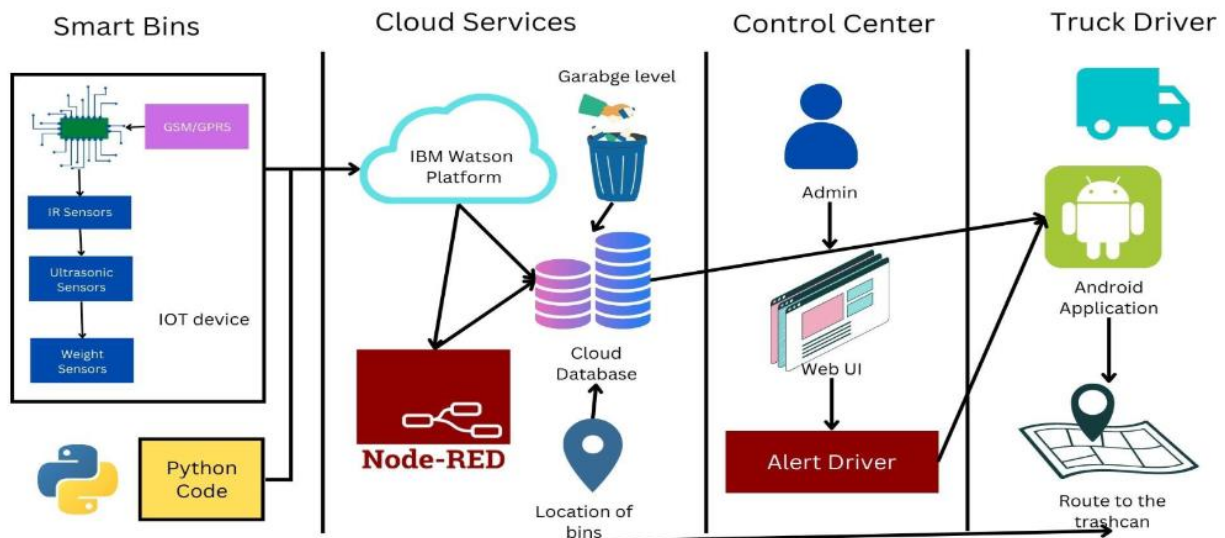


Table-1 : Components & Technologies:

| S.No | Component | Description | Technology |
|------|---------------------|--|--|
| 1 | Arduino Uno | The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller. | Arduino programming itself is done in C++. |
| 2 | Application Logic-1 | Logic for IR sensor data. | C++/Python. |
| 3 | Application Logic-2 | Logic for Ultrasonic sensor data. | C++/Python. |
| 4 | Application Logic-3 | Logic for a Weight sensor data | C++/Python. |
| 5 | GPRS/GSM | The Arduino GSM shield allows an Arduino board to connect to the internet, send and receive SMS, | C++/Python. |

| | | | |
|---|----------------|--|--------------------------------------|
| | | and make voice calls using the GSM library. | |
| 6 | Cloud Sever | Application deployment on Local System / Cloud | IBM Watson IoT Platform, Node Red |
| 7 | Cloud Database | Database Service on Cloud | IBM Watson IoT platform, Cloudant DB |

Table-2: Application Characteristics:

| S.No | Characteristics | Description | Technology |
|------|-----------------------------|--|--|
| 1 | Open-Source Microcontroller | Arduino Uno is used to make the IoT device | C++/Python |
| 2 | Security | Encryption/Decryption used for security purpose | GSM/GPRS,Python |
| 3 | Scalable Architecture | New features can be added. | Node Red |
| 4 | Availability | Web application can be accessed from anywhere | IBM Watson IoT Platform, HTML, CSS, JavaScript |
| 5 | Performance | All truck drivers can access the application at same time. | Cloudant DB, IBM Watson IoT Platform |
| | | | |

5.3 User Stories:

| User Type | Functional Requirement (Epic) | User Story Number | User Story / Task | Acceptance criteria | Priority | Release |
|-----------|-------------------------------|-------------------|-------------------|---------------------|----------|---------|
| Admin | Login | USN-1 | As an | I can | High | Sprint- |

| | | | | | | |
|------------------|-------|-------|--|-----------------------------------|--------|----------|
| | | | admin, I can monitor every dustbin and its garbage levels | monitor the system. | | 4 |
| | Login | USN-2 | As an admin, I will inform the authorized person to empty the trashcan | I can inform authorized person. | Medium | Sprint-2 |
| | Login | USN-3 | As an admin, I can notice the trash level of every dustbin | I can notice the trash level. | Low | Sprint-2 |
| Admin 2 | Login | USN-4 | As a Co-Admin, I can send alert message to the truck drivers | I can alert truck driver | High | Sprint-1 |
| Trash Van Driver | Login | USN-5 | As a trash van driver, I will follow the route to the dustbin | I can reach the filled trashcans. | High | Sprint-2 |

| | | | | | | |
|-------------------|----------|-------|---|--|--------|----------|
| Garbage Collector | Login | USN-6 | As a waste collector, I will collect all the trash from the dumpsters and load it to the truck. | I can empty the trashcans | Medium | Sprint-2 |
| Municipal officer | Login | USN-7 | As a municipal officer, I can supervise the process and ensure the cleanliness of city | I can manage all these process going good. | High | Sprint-1 |
| Trashcan Monitor | Register | USN-8 | As a trashcan monitor, I can initialize new trashcans. | I can manage all these process going good. | Medium | Sprint-3 |
| | | USN-9 | As a trashcan monitor, I can check the quality of IoT device's quality. | I can check the IoT device | Medium | Sprint-3 |

6. PROJECT PLANNING & SCHEDULING:

6.1 Sprint Planning & Estimation:

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

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 | Monitoring | USN-1 | The IoT device will monitor the garbage level in trashcans. | 20 | High | Stephenraj.W Justinraj.J Merunraj.W Karan.D |
| Sprint-1 | Registration | USN-2 | As a trashcan monitor I can initialize new trashcans. | 20 | Low | Stephenraj.W Justinraj.J Merunraj.W Karan.D Stephenraj.W Justinraj.J Merunraj.W Karan.D |
| Sprint-2 | Dashboard | USN-3 | As an admin, I can monitor | 20 | High | Stephenraj.W Justinraj.J Merunraj.W Karan.D |

| | | | | | | |
|----------|---------------|-------|--|----|--------|--|
| | | | every dustbin and its garbage levels | | | |
| Sprint-3 | Alert | USN-4 | As a Co-Admin, I can send alert message to the truck drivers. | 20 | High | Stephenraj.W Justinraj.J Merunraj.W Karan.D |
| Sprint-4 | Location View | USN-5 | As a trash van driver, I will follow the route to the dustbin. | 20 | Medium | Stephenraj.W Justinraj.J Merunraj.W Karan.D |

6.2. Sprint Delivery Schedule:

Project Tracker, Velocity & Burndown Chart: (4 Marks)

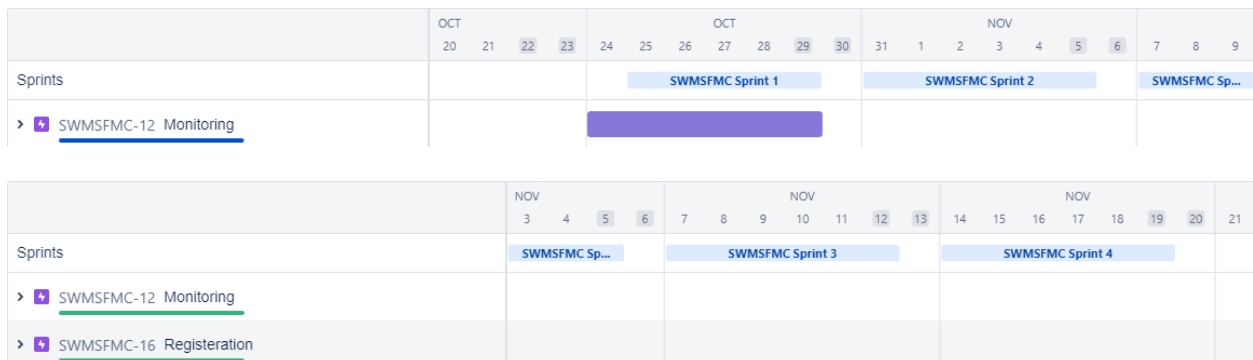
| Sprint | Total Story Points | Duration | Sprint Start Date | Sprint End Date (Planned) | Story Points Completed (as on Planned End Date) | Sprint Release Date (Actual) |
|---------------|---------------------------|-----------------|--------------------------|----------------------------------|--|-------------------------------------|
| Sprint-1 | 20 | 6 Days | 24 Oct 2022 | 29 Oct 2022 | 20 | 29 Oct 2022 |
| Sprint-2 | 20 | 6 Days | 31 Oct | 05 Nov | 20 | 05 Nov |

| | | | | | | |
|----------|----|--------|------------|-------------|----|-------------|
| | | | 2022 | 2022 | | 2022 |
| Sprint-3 | 20 | 6 Days | 07 Nov2022 | 12 Nov 2022 | 20 | 12 Nov 2022 |
| Sprint-4 | 20 | 6 Days | 14 Nov2022 | 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).

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

6.3 Reports from JIRA:



7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1.Wokwi:

WOKWI

esp32-dht22.ino copy - Wokwi A. X

wokwi.com/projects/347927859012043348

SAVE SHARE

Docs

esp32-dht22.ino diagram.json libraries.txt Library Manager

```

1 #include <WiFi.h> // library for wifi
2 #include <PubSubClient.h> // library for MQTT
3 #include <LiquidCrystal_I2C.h>
4 LiquidCrystal_I2C lcd(0x27, 20, 4);
5
6 //----- credentials of IBM Accounts -----
7
8 #define ORG "jrb15n" // IBM organisation id
9 #define DEVICE_TYPE "Assignment4" // Device type mentioned in i
10 #define DEVICE_ID "12345" // Device ID mentioned in ibm watson io
11 #define TOKEN "12345678" // Token
12
13 //----- customise above values -----
14
15 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
16 char publishTopic[] = "iot-2/evt/data/fmt/json";
17 char topic[] = "iot-2/cmd/led/fmt/String";
18 char authMethod[] = "use-token-auth";
19 char token[] = TOKEN;
20 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
21
22 //-----
23
24 WiFiClient wifiClient;
25 PubSubClient client(server, 1883, wifiClient);
26
27 #define ECHO_PIN 12
28 #define TRIG_PIN 13
29 float dist;
30
31 void setup()
32 {
33   pinMode(ECHO_PIN, OUTPUT);
34   pinMode(TRIG_PIN, OUTPUT);
35 }
36
37 void loop()
38 {
39   if (client.connected()) {
40     client.publish(topic, "cmd OK");
41   }
42 }

```

Simulation

00:32.757 45%

Connecting to Wifi..WiFi connected, IP address: 10.10.0.2
Reconnecting MQTT client to
jrb15n.messaging.internetofthings.ibmcloud.com
IBM subscribe to cmd OK

7.2. Watson:

Node-RED: node-red x Node-RED Dashboard x Cloudant Dashboard x IBM Watson IoT Platform x MIT App Inventor x IBM-Project-44887-16 x

uzesig.internetofthings.ibmcloud.com/dashboard/devices/browse

Dashboard IBM Watson IoT Platform Smart Waste Mana... Course | ML0201EN... IBM-EPBL/IBM-Proj... node red IBM App Developm... Node-RED on IBM...

IBM Watson IoT Platform

821919104027@smartinternz.com ID: uzesig

Browse Action Device Types Interfaces

Add Device

Search by Device ID

Device Simulator

| Device ID | Status | Device Type | Class ID | Date Added |
|-----------|--------------|-------------|----------|----------------------|
| 12345 | Disconnected | Arduino | Device | Nov 14, 2022 2:50 PM |

Identity Device Information Recent Events State Logs

| | |
|-------------------|-------------------------------|
| Device ID | 12345 |
| Device Type | Arduino |
| Date Added | Nov 14, 2022 2:50 PM |
| Added By | 821919104027@smartinternz.com |
| Connection Status | Disconnected |

Items per page 50 | 1-2 of 2 items

0 Simulations running

```
Python 3.7.9 Shell
File Edit Shell Debug Options Window Help

Python 3.7.9 (tags/v3.7.9:13c9474c7, Aug 17 2020, 18:58:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>

===== RESTART: C:\Users\steph\OneDrive\Desktop\Arduinosp32.py =====
2022-11-18 10:42:21,248 ibmiotf.device.Client INFO Connected successfully: dnuzeisig:Arduino:12345
Published Weight of Trashcan is = 73 C IR Sensor = 94 % Ultrasonic Sensor = 96 % to IBM Watson
Published Weight of Trashcan is = 3 C IR Sensor = 77 % Ultrasonic Sensor = 5 % to IBM Watson
Published Weight of Trashcan is = 48 C IR Sensor = 32 % Ultrasonic Sensor = 67 % to IBM Watson
Published Weight of Trashcan is = 66 C IR Sensor = 53 % Ultrasonic Sensor = 25 % to IBM Watson
Published Weight of Trashcan is = 53 C IR Sensor = 1 % Ultrasonic Sensor = 39 % to IBM Watson
Published Weight of Trashcan is = 42 C IR Sensor = 26 % Ultrasonic Sensor = 82 % to IBM Watson
Published Weight of Trashcan is = 22 C IR Sensor = 13 % Ultrasonic Sensor = 18 % to IBM Watson
Published Weight of Trashcan is = 90 C IR Sensor = 7 % Ultrasonic Sensor = 52 % to IBM Watson
Published Weight of Trashcan is = 37 C IR Sensor = 4 % Ultrasonic Sensor = 86 % to IBM Watson
Published Weight of Trashcan is = 18 C IR Sensor = 51 % Ultrasonic Sensor = 29 % to IBM Watson
Published Weight of Trashcan is = 39 C IR Sensor = 46 % Ultrasonic Sensor = 63 % to IBM Watson
Published Weight of Trashcan is = 83 C IR Sensor = 77 % Ultrasonic Sensor = 52 % to IBM Watson
Published Weight of Trashcan is = 96 C IR Sensor = 43 % Ultrasonic Sensor = 2 % to IBM Watson
Published Weight of Trashcan is = 13 C IR Sensor = 89 % Ultrasonic Sensor = 86 % to IBM Watson
Published Weight of Trashcan is = 28 C IR Sensor = 63 % Ultrasonic Sensor = 31 % to IBM Watson
|

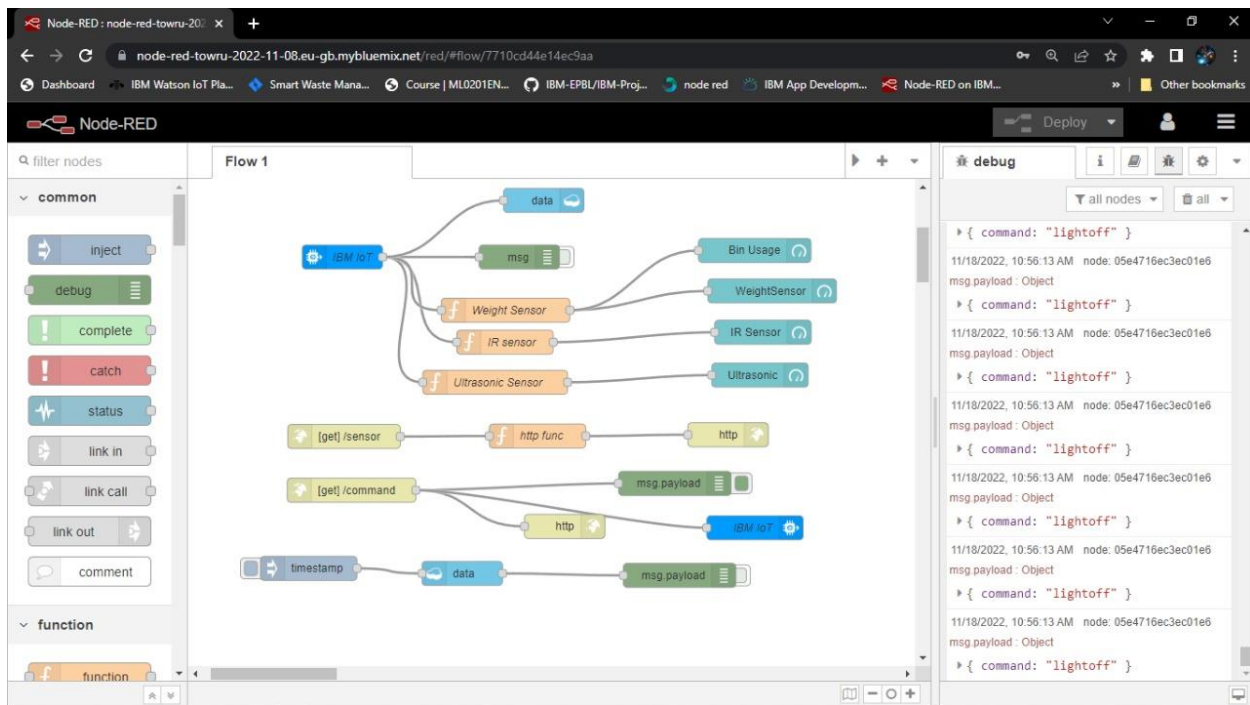
Ln: 21 Col: 0
```

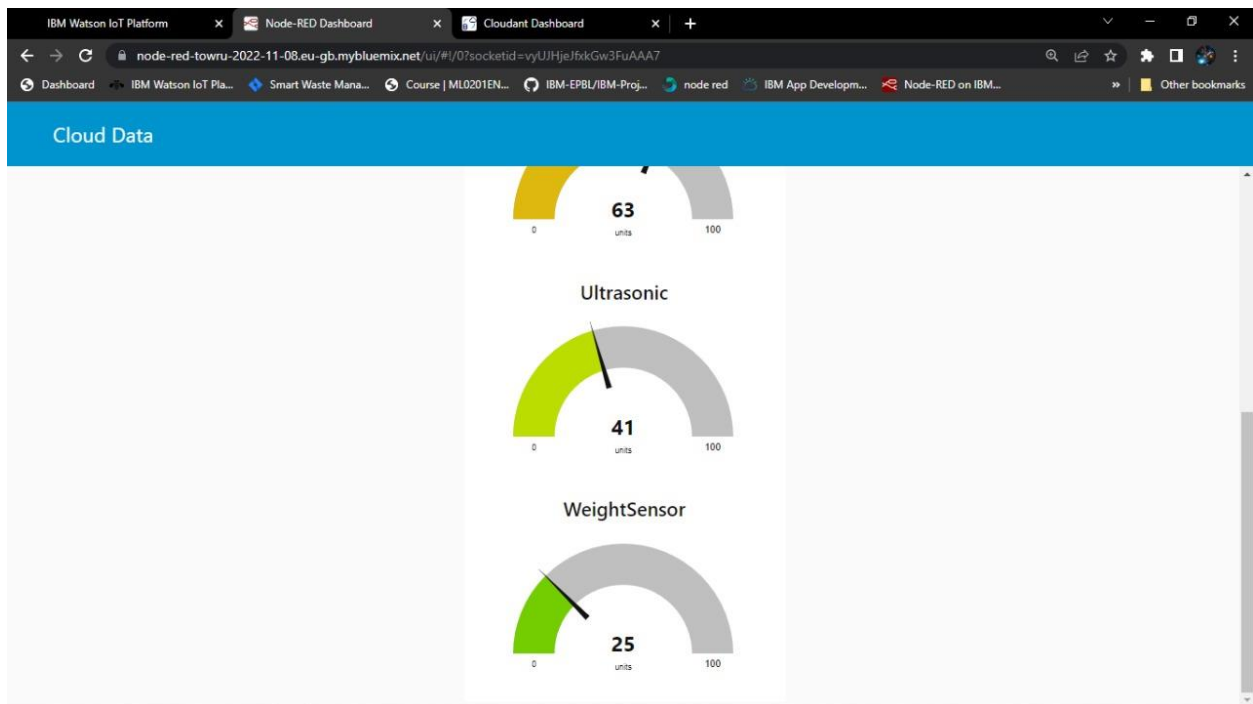
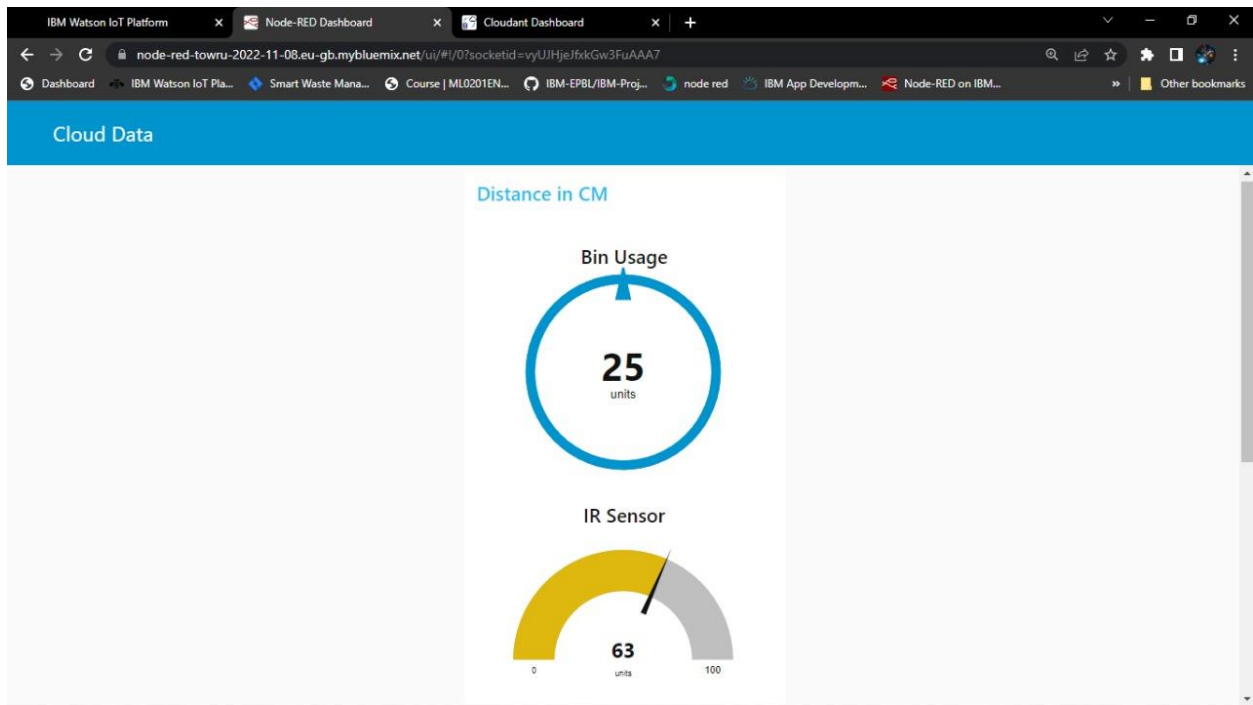
7.3.Cloudant:

The screenshot shows the IBM Cloudant Dashboard interface. The left sidebar contains navigation options: All Documents, Query, Permissions, Changes, and Design Documents. The main area displays a table of documents with columns for id, key, and value. The table lists 20 documents, each with a unique id and a JSON value containing a 'rev' field. The bottom of the interface shows 'Showing document 1 - 20' and 'Documents per page: 20'.

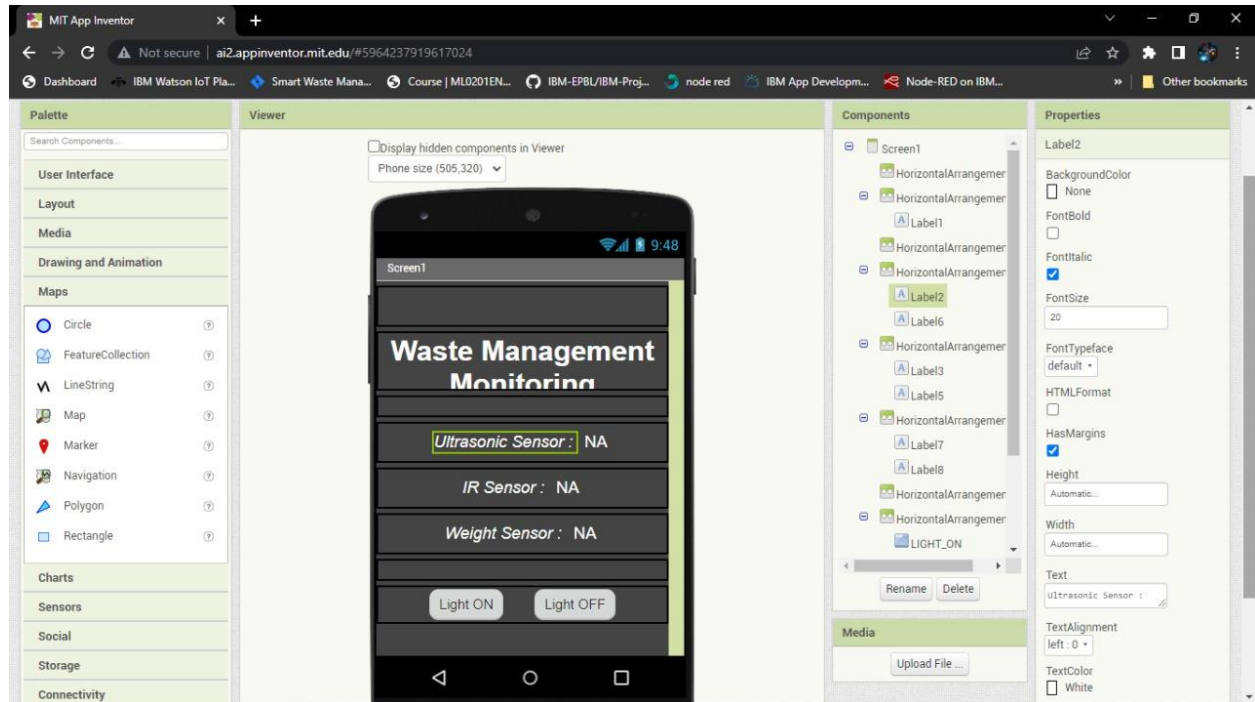
| id | key | value |
|-----------------------------------|-----------------------------------|--|
| 015e5f6f53d22612fb2ba43d58bf16... | 015e5f6f53d22612fb2ba43d58bf16... | { "rev": "1-abde29fa9f4a1b308e93cc..." } |
| 015e5f6f53d22612fb2ba43d58bf35... | 015e5f6f53d22612fb2ba43d58bf35... | { "rev": "1-1f5ed08c32b437ecd5028..." } |
| 015e5f6f53d22612fb2ba43d58bf4a... | 015e5f6f53d22612fb2ba43d58bf4a... | { "rev": "1-d408e4c07691d4a085c8..." } |
| 015e5f6f53d22612fb2ba43d58bf6d... | 015e5f6f53d22612fb2ba43d58bf6d... | { "rev": "1-d6c53f6e405a84051a032..." } |
| 015e5f6f53d22612fb2ba43d58c032... | 015e5f6f53d22612fb2ba43d58c032... | { "rev": "1-5584e7b1a5ead1cdc45a8..." } |
| 0384c12e6c59f36c1351fc758d3c7d... | 0384c12e6c59f36c1351fc758d3c7d... | { "rev": "1-0dc097571ee4e492f1a00..." } |
| 0384c12e6c59f36c1351fc758d40c1... | 0384c12e6c59f36c1351fc758d40c1... | { "rev": "1-25fb428aa314a4c225d5b..." } |
| 0384c12e6c59f36c1351fc758d40d... | 0384c12e6c59f36c1351fc758d40d... | { "rev": "1-25fb428aa314a4c225d5b..." } |
| 0384c12e6c59f36c1351fc758d40f8... | 0384c12e6c59f36c1351fc758d40f8... | { "rev": "1-25fb428aa314a4c225d5b..." } |
| 0384c12e6c59f36c1351fc758d412... | 0384c12e6c59f36c1351fc758d412... | { "rev": "1-617f1e98e0a56741598c1..." } |
| 0384c12e6c59f36c1351fc758d41a6... | 0384c12e6c59f36c1351fc758d41a6... | { "rev": "1-0490f59218f137db1c7c4..." } |

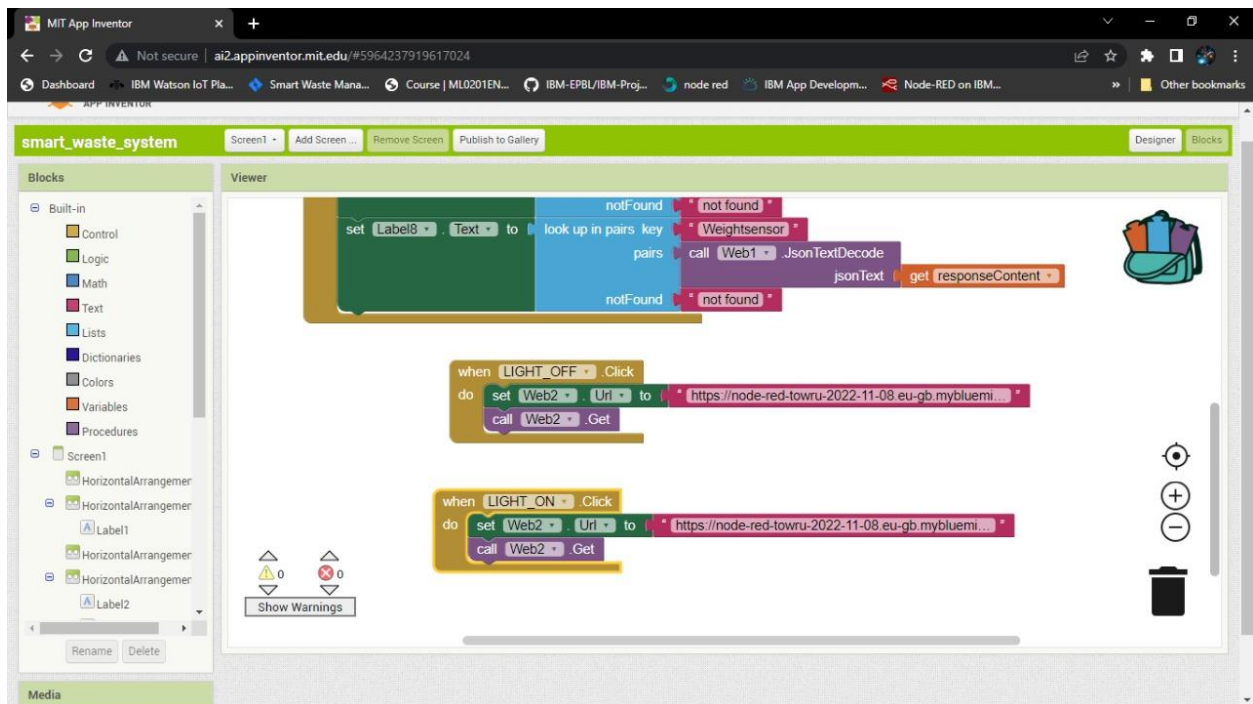
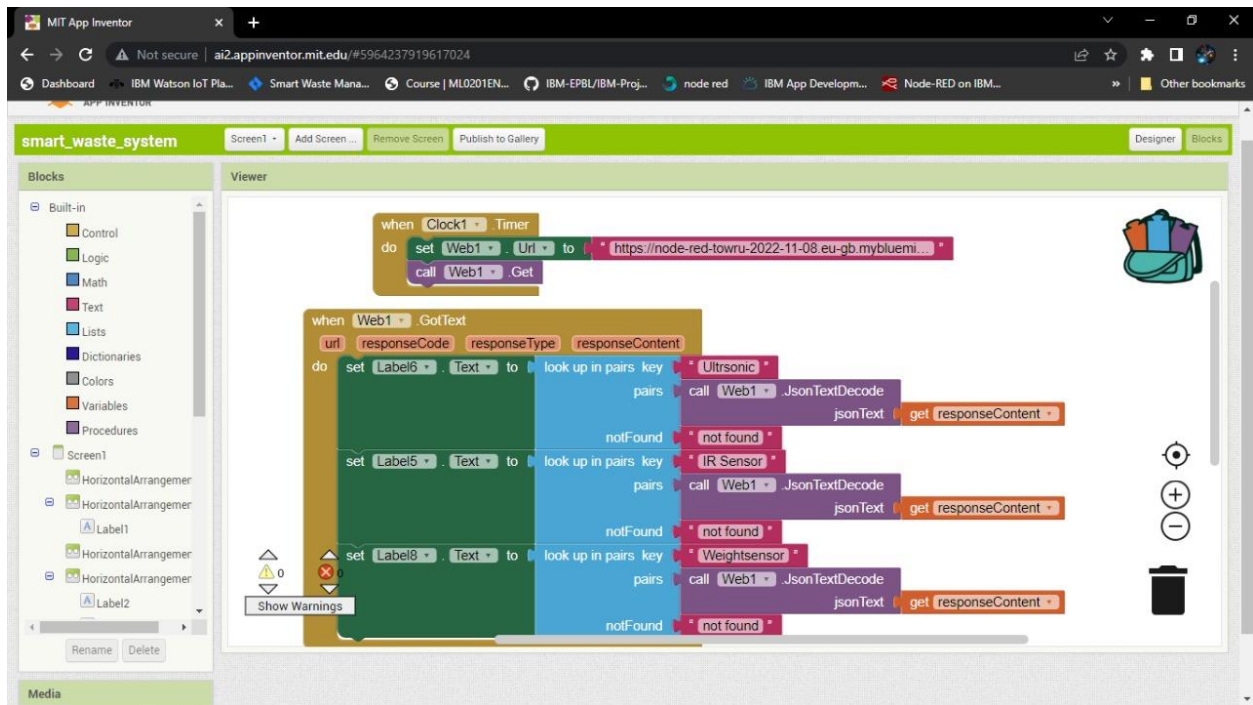
7.4.Nodered:





7.5.MIT App:





11:07



Screen1

Waste Management Monitoring

Ultrasonic Sensor : 41

IR Sensor : 63

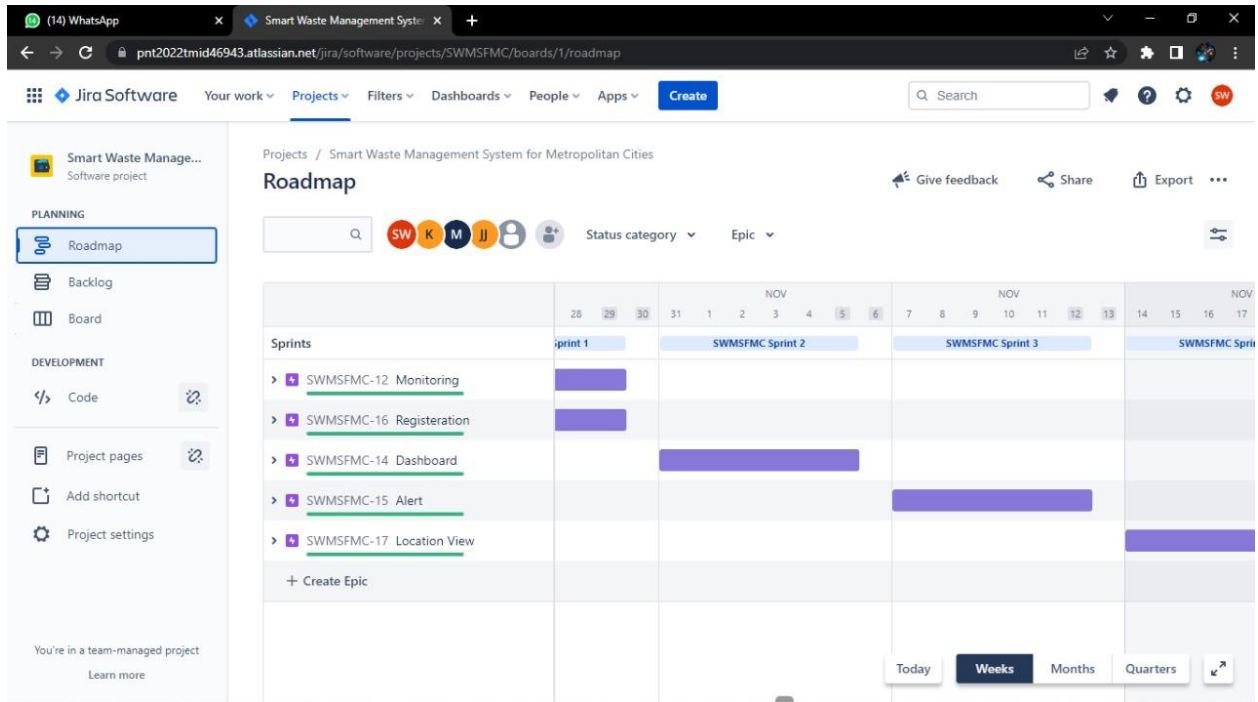
Weight Sensor : 25

Light ON

Light OFF

8.RESULTS & TESTING:

8.1.Performance Metrics:



9. ADVANTAGES & DISADVANTAGES:

ADVANTAGES:

- A reduction in the number of waste collections needed by up to 80%.
- Resulting in less manpower.
- Emissions.
- Fuel use and traffic congestion.

DISADVANTAGES:

- Increasing cost of the dustbin.
- Some wastes cannot be recycled.

- Technological push needed.
- Separation of useful material from waste difficult.

10.CONCLUSION:

This project is very effective in managing waste in any big city. Rather than using conventional periodic collection methods, a priority system is used to ensure the city is clean all the time without any overflowing dumpsters. It has been tested and verified properly to ensure all the different parts work together for a smooth function of the whole system. In most of the metro cities globally poses a challenge to effective 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.

11.FUTURE SCOPE:

- Pollution prevention and source reduction.
- Reuse or redistribution of unwanted.
- Surplus materials; treatment, reclamation.
- And recycling of materials within the waste.
- And disposal through incineration, treatment, or land burial.

12) APPENDIX:

Source Code

```
import time

import sys

import ibmiotf.application

import ibmiotf.device
```

```
import random
```

```
#Provide your IBM Watson Device Credentials
```

```
organization = "uzesig"
```

```
deviceType = "Arduino"
```

```
deviceId = "12345"
```

```
authMethod = "token"
```

```
authToken = "12345678"
```

```
# Initialize GPIO
```

```
def myCommandCallback(cmd):
```

```
    print("Command received: %s" % cmd.data['command'])
```

```
    status=cmd.data['command']
```

```
    if status=="lighton":
```

```
        print ("led is on")
```

```
    else :
```

```
        print ("led is off")
```

```
    #print(cmd)
```

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()
```

Connect and send a datapoint "hello" with value "world" into the cloud as an event of type "greeting" 10 times

```
deviceCli.connect()
```

while True:

```
    #Get Sensor Data from esp32
```

```
    weightSensor=random.randint(0,100)
```

```
    irSensor=random.randint(0,100)
```

```
    ultrasSensor=random.randint(0,100)
```



```

data = { 'weight' : weightSensor, 'ir':irSensor, 'ultrasonic':ultrasSensor }

#print data

def myOnPublishCallback():

    print ("Published Weight of Trashcan is = %s C" % weightSensor, "IR
Sensor = %s %" % irSensor, "Ultrasonic Sensor = %s %" % ultrasSensor, "to
IBM Watson")

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

    if not success:

        print("Not connected to IoTF")

        time.sleep(1)

    deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud
deviceCli.disconnect()

```

Wokwi Code

```

#include <WiFi.h>                // library for wifi

#include <PubSubClient.h>        // library for MQTT

#include <LiquidCrystal_I2C.h>

```

```
LiquidCrystal_I2C lcd(0x27, 20, 4);
```

```
//----- credentials of IBM Accounts -----
```

```
#define ORG "jrbl5n" // IBM organisation id
```

```
#define DEVICE_TYPE "Arduino" // Device type mentioned in ibm  
watson iot platform
```

```
#define DEVICE_ID "12345" // Device ID mentioned in ibm watson iot  
platform
```

```
#define TOKEN "12345678" // Token
```

```
//----- customise above values -----  
-----
```

```
char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; //  
server name
```

```
char publishTopic[] = "iot-2/evt/data/fmt/json"; // topic name  
and type of event perform and format in which data to be send
```

```
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  
wificlient
```

```
PubSubClient client(server, 1883, wifiClient);
```

```
#define ECHO_PIN 12
```

```
#define TRIG_PIN 13
```

```
float dist;
```

```
void setup()
```

```
{
```

```
  Serial.begin(115200);
```

```
  pinMode(LED_BUILTIN, OUTPUT);
```

```
  pinMode(TRIG_PIN, OUTPUT);
```

```
  pinMode(ECHO_PIN, INPUT);
```

```
  //pir pin
```

```
  pinMode(34, INPUT);
```

```
  //ledpins
```

```
  pinMode(23, OUTPUT);
```

```
pinMode(2, OUTPUT);  
pinMode(4, OUTPUT);  
pinMode(15, OUTPUT);
```

```
lcd.init();  
lcd.backlight();  
lcd.setCursor(1, 0);  
lcd.print("");  
wifiConnect();  
mqttConnect();  
}
```

```
float readcmCM()  
{  
    digitalWrite(TRIG_PIN, LOW);  
    delayMicroseconds(2);  
    digitalWrite(TRIG_PIN, HIGH);  
    delayMicroseconds(10);  
    digitalWrite(TRIG_PIN, LOW);  
    int duration = pulseIn(ECHO_PIN, 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))                //pir motion detection
    {
        Serial.println("Motion Detected");
        Serial.println("Lid Opened");
        digitalWrite(15, HIGH);
    }
}
```

```
if(digitalRead(34)== true)
{
  if(cm <= 60)                                     //Bin level detection
  {
    digitalWrite(2, 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(4, LOW);
    digitalWrite(23, LOW);
  }
  else if(cm > 60 && cm < 120)
  {
    digitalWrite(4, HIGH);
    Serial.println("Warning!!,Trash is about to cross 50% of bin level");
    digitalWrite(2, LOW);
    digitalWrite(23, LOW);

  }
  else if(cm > 120)
  {
```



```
    digitalWrite(23, HIGH);  
    Serial.println("Bin is available");  
    digitalWrite(2, LOW);  
    digitalWrite(4, LOW);  
  
    }  
    delay(10000);  
    Serial.println("Lid Closed");  
    }  
else  
{  
    Serial.println("No motion detected");  
    digitalWrite(2, LOW);  
    digitalWrite(15, LOW);  
    digitalWrite(4, LOW);  
    digitalWrite(23, LOW);  
    }  
  
}  
else  
{  
    digitalWrite(15, LOW);
```

```
}
```

```
    if(cm <= 60)
    {
        digitalWrite(21,HIGH);
        String payload = "{\"High_Alert\":\"";
        payload += cm;
        payload += " }";
        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 else prints publish failed
        {
            Serial.println("Publish OK");
        }
    }

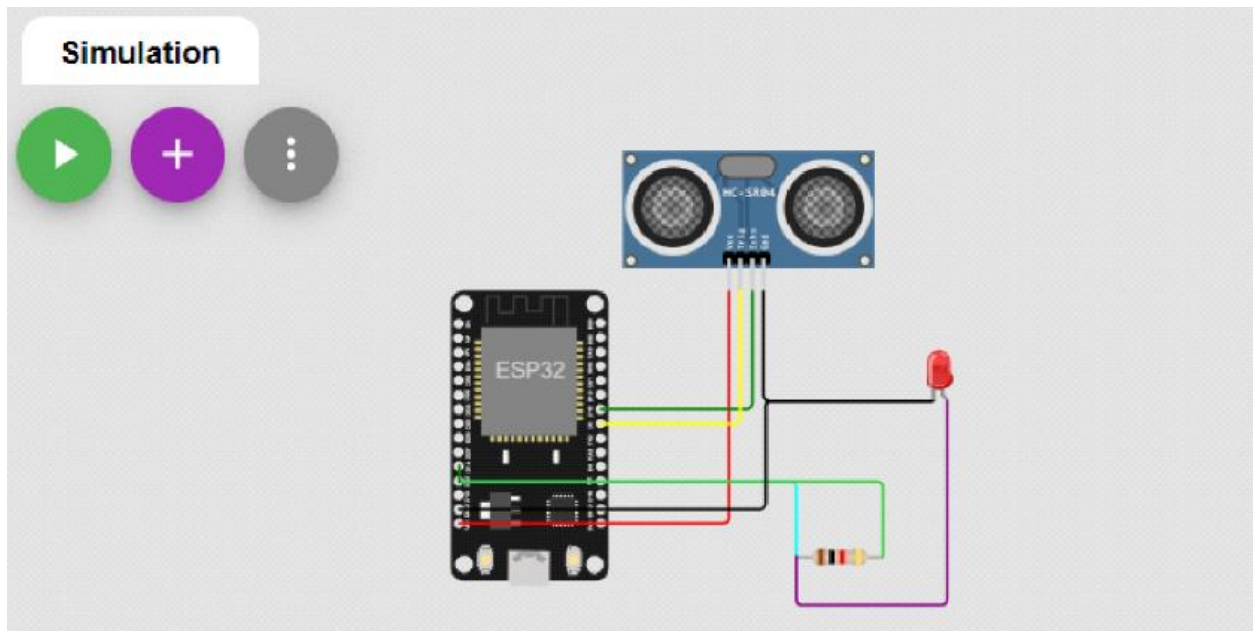
    else if(cm <= 120)
    {
        digitalWrite(22,HIGH);
```

```
String payload = "{\"Warning\":\"";
payload += cm ;
payload += " }";
Serial.print("\n");
Serial.print("Sending payload: ");
Serial.println(payload);
if(client.publish(publishTopic, (char*) payload.c_str()))
{
    Serial.println("Publish OK");
}
else
{
    Serial.println("Publish FAILED");
}
}
else
{
    Serial.println();
}

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();  
}
```

13. OUTPUT PICTURE:



IBM Watson IoT Platform | sketchino - Wokwi Arduino and |

wokwi.com/projects/347476374690202194

WOKWI | SAVE | SHARE | sketchino | Docs

sketch.ino | diagram.json | libraries.txt | Library Manager

```

1 #include <Wi-Fi.h>
2 #include <PubSubClient.h>
3 WiFiClient wifiClient;
4 String data3;
5 #define ORG "6vypnd"
6 #define DEVICE_TYPE "ESP32"
7 #define DEVICE_ID "Assignment4"
8 #define TOKEN "12345678"
9 #define speed 0.034
10 #define led 14
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
12 char publishTopic[] = "iot-2/evt/data/fmt/json";
13 char topic[] = "iot-2/cmd/home/fmt/String";
14 char authMethod[] = "use-token-auth";
15 char token[] = TOKEN;
16 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
17 PubSubClient client(server, 1883, WiFiClient);
18 void publishData();
19 const int trigpin = 5;
20 const int echopin = 18;
21 String command;
22 String data = "";
23 long duration;
24 float dist;
25
26 void setup()
27 {
28   Serial.begin(115200);
29   pinMode(led, OUTPUT);
30   pinMode(trigpin, OUTPUT);

```

Simulation

00:16.506 93%

Sending payload: {"Normal Distance":44.97}
Publish OK

Sending payload: {"Normal Distance":44.97}
Publish OK

Sending payload: {"Normal Distance":44.97}
Publish OK

Sending payload: {"Normal Distance":44.97}
Publish OK

Type here to search

IBM Watson IoT Platform | 6vypnd.internetofthings.ibmcloud.com/dashboard/devices/browse

IBM Watson IoT Platform | 021919104000@smarterintenz.com | ID: 6vypnd

Browse | Action | Device Types | Interfaces

Add Device

| Device ID | Status | Device Type | Class ID | Date Added |
|-------------|--------------|-------------|----------|---------------------|
| Assignment4 | Disconnected | ESP32 | Device | Nov 5, 2022 1:01 AM |

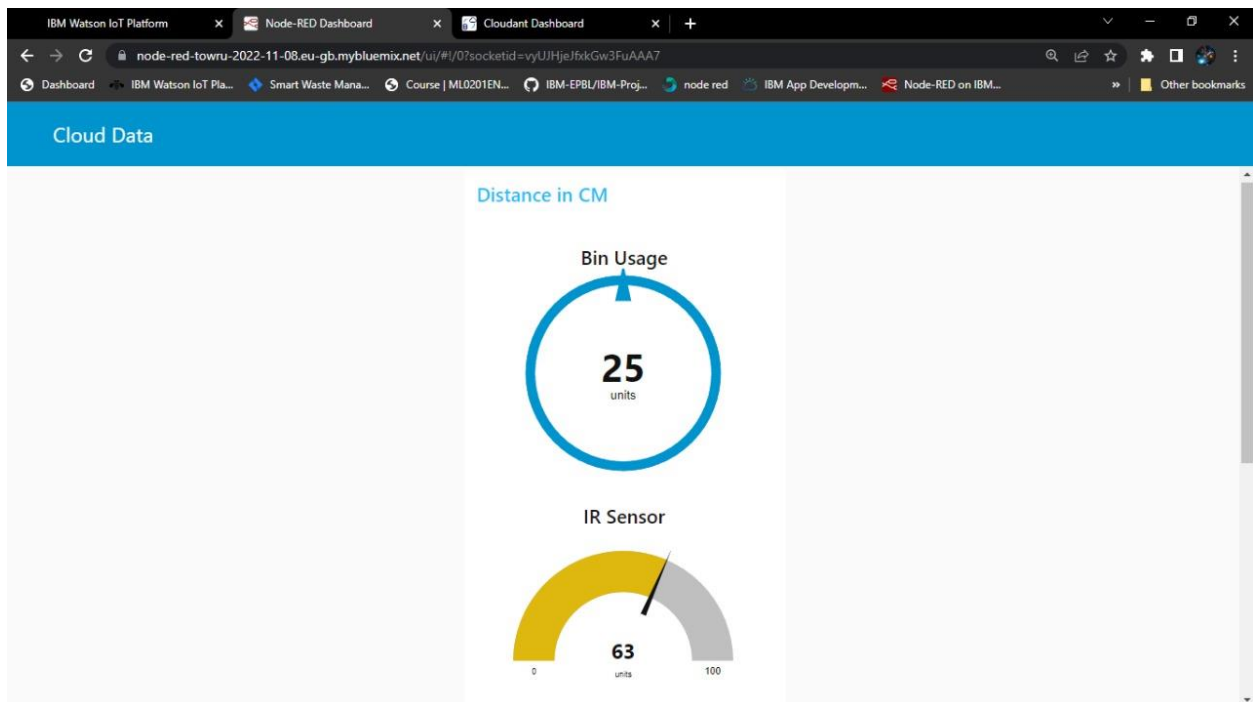
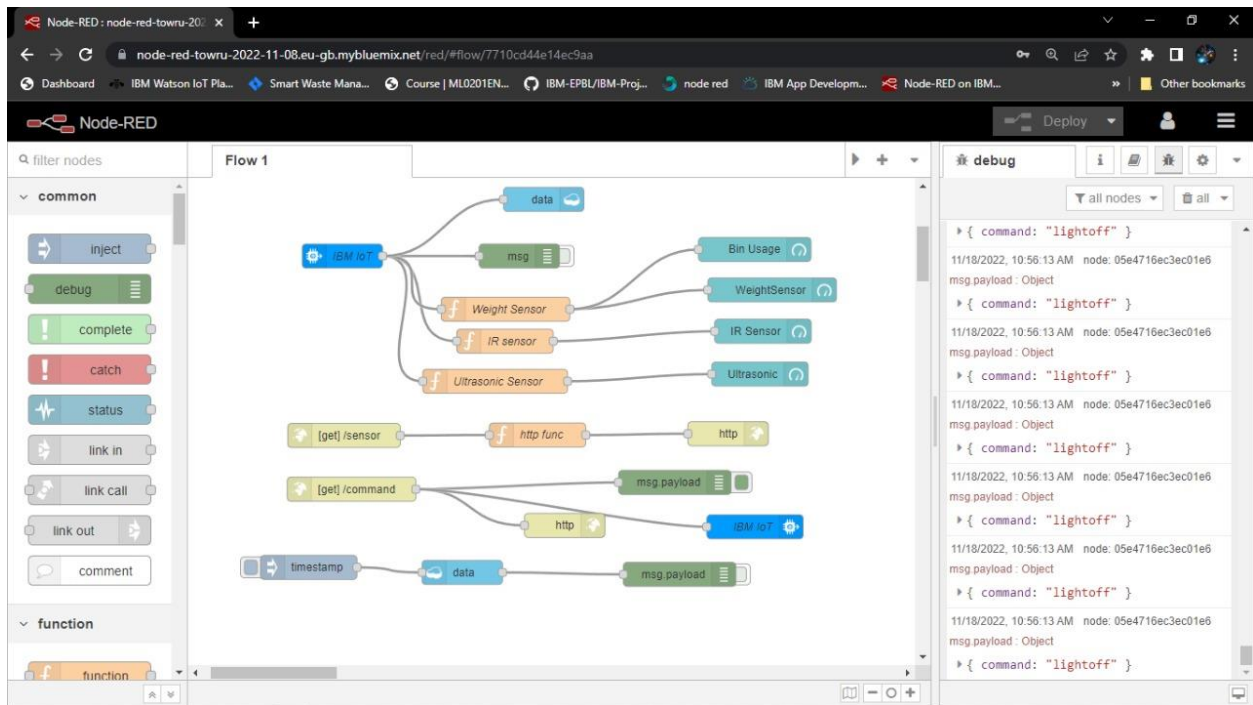
Identity | Device Information | Recent Events | State | Logs

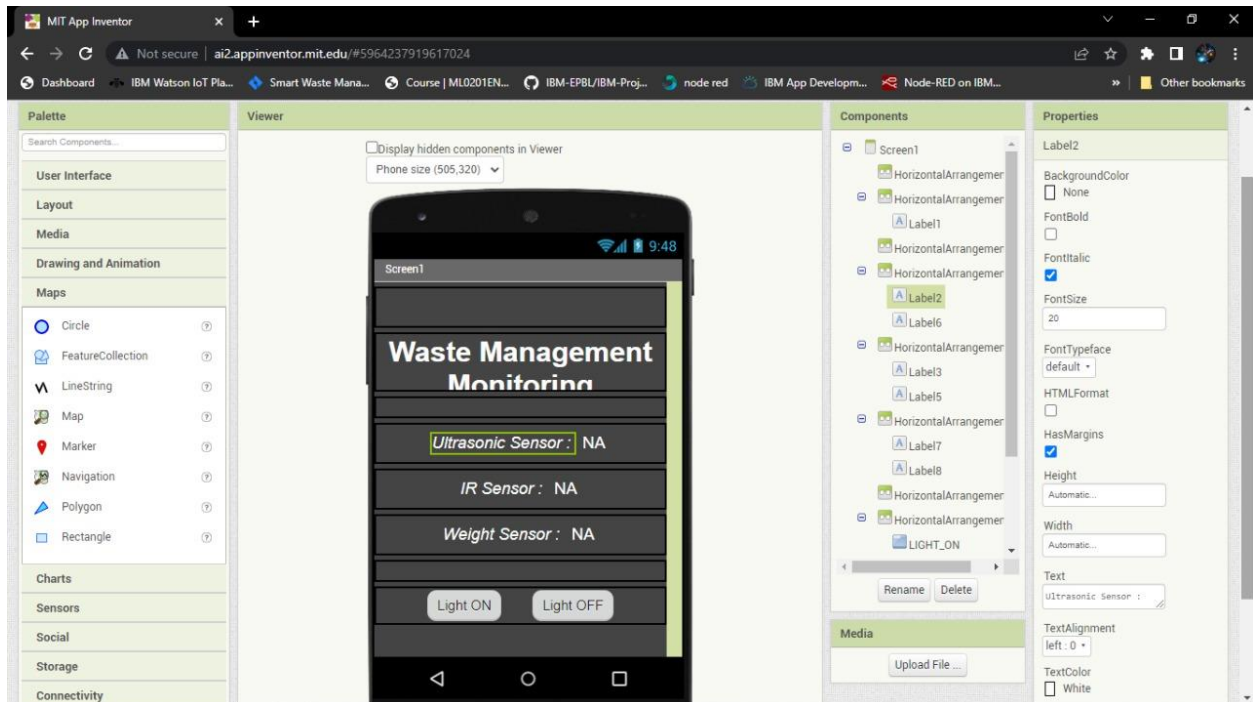
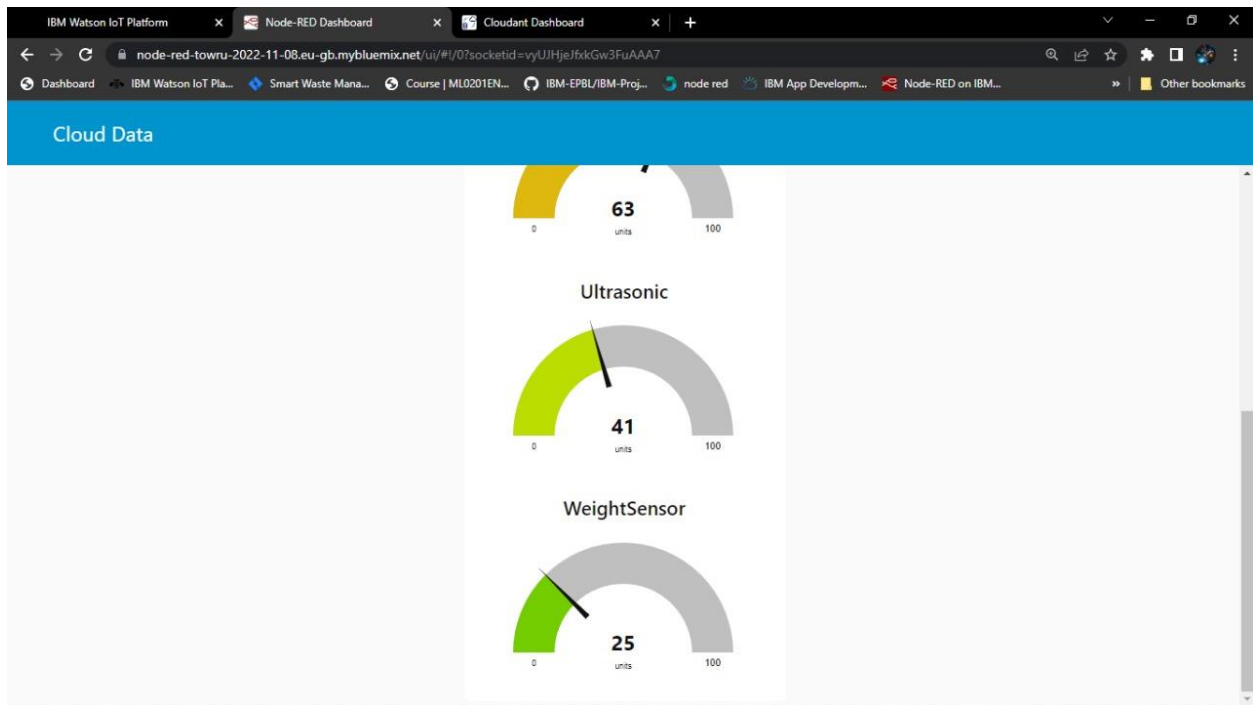
The recent events listed show the live stream of data that is coming and going from this device.

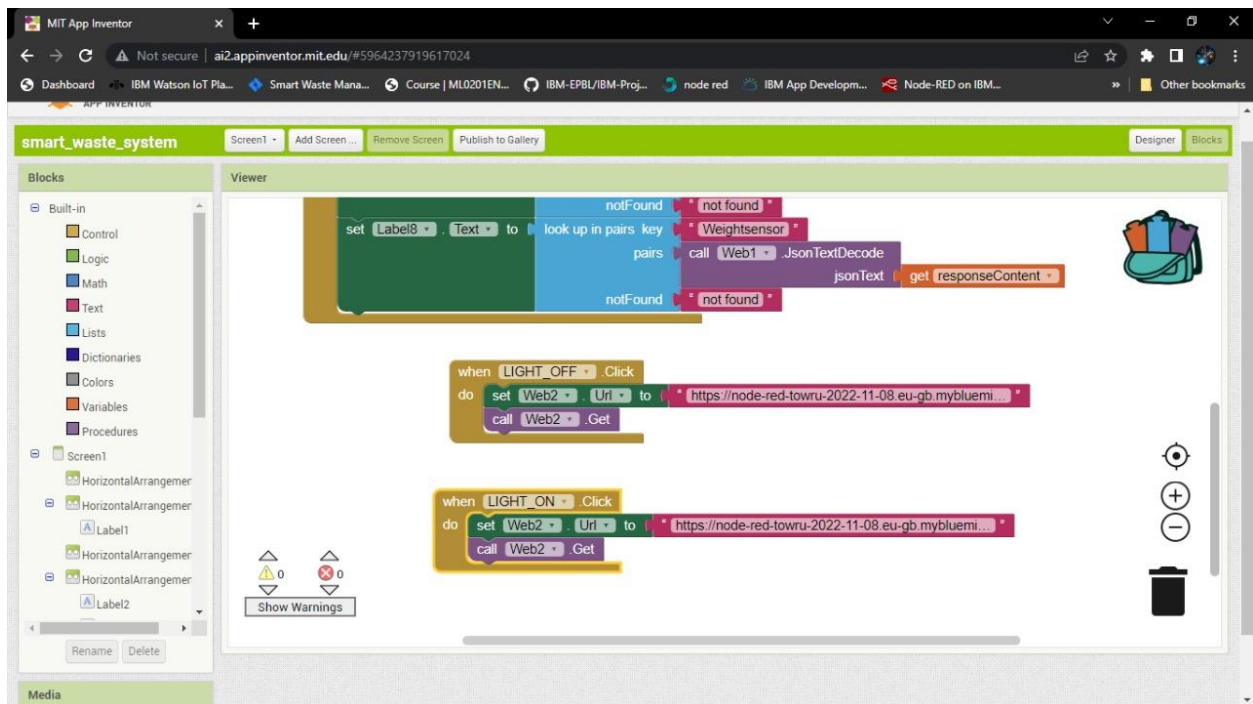
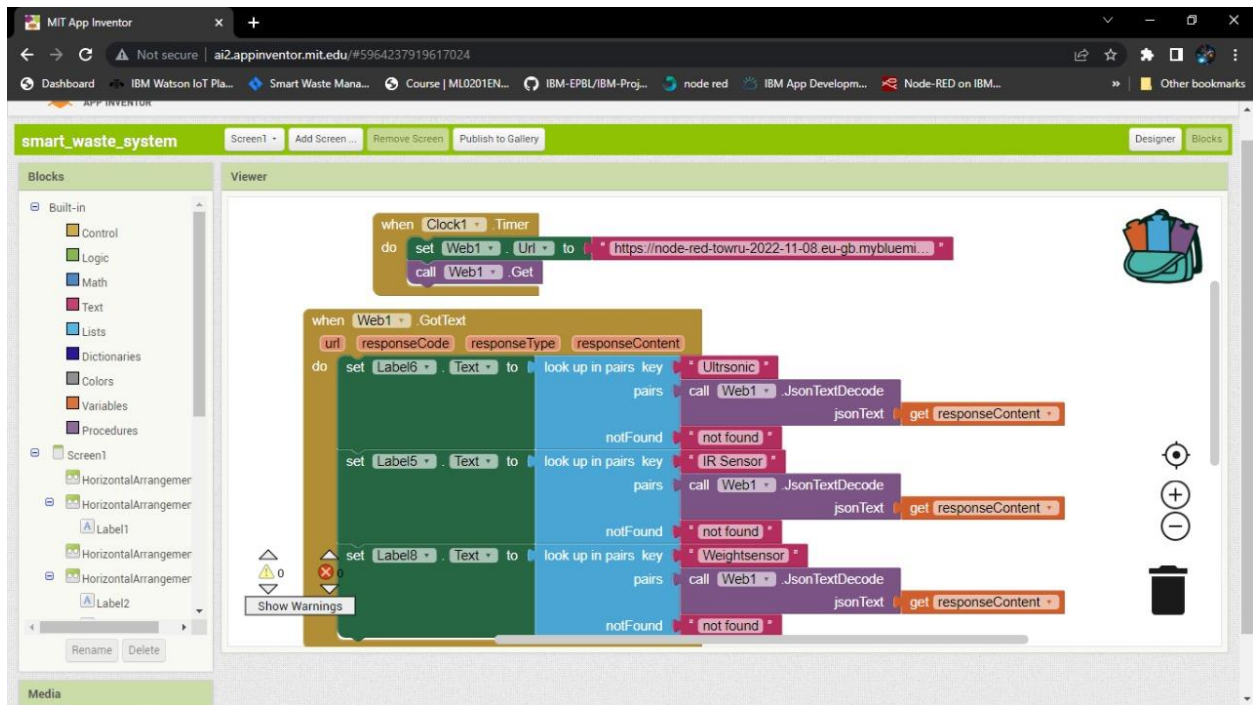
| Event | Value | Format | Last Received |
|-------|---------------------------|--------|-------------------|
| data | {"Normal Distance":44.97} | json | a few seconds ago |
| data | {"Normal Distance":44.98} | json | a few seconds ago |
| data | {"Normal Distance":45.03} | json | a few seconds ago |
| data | {"Normal Distance":44.97} | json | a few seconds ago |
| data | {"Normal Distance":44.97} | json | a few seconds ago |

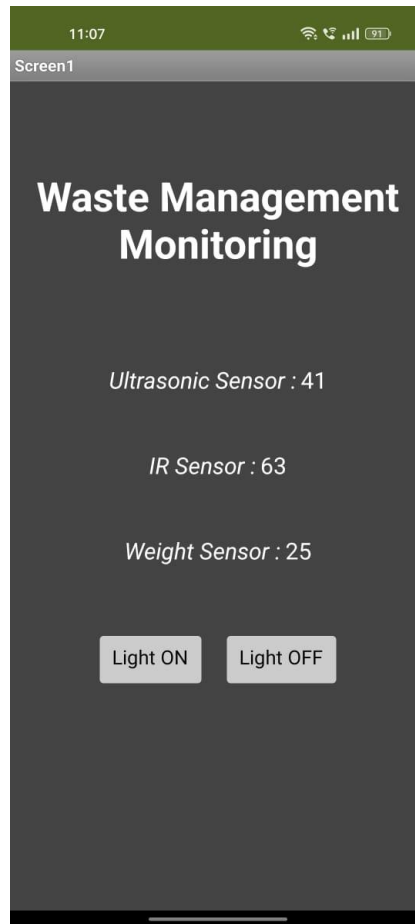
0 Simulations running

Type here to search









13.1. LINKS:

GitHub Link:

<https://github.com/IBM-EPBL/IBM-Project-44887-1660727265>

Wokwi Link:

<https://wokwi.com/projects/347927859012043348>

MIT App Link:

<http://ai2.appinventor.mit.edu/#5964237919617024>

Node Link:

https://node-red-towru-2022-11-08.eu-gb.mybluemix.net/ui/#!/0?socketid=4K9d90PmJD_0qgwwAABF

Video Demo Link:

<https://youtu.be/eDVDmruqYoA>