SMART WASTE MANAGEMENT SYSTEM FOR METROPOLITAN CITIES

IBM-DOCUMENTATION

UNDER THE GUIDENCE OF

FACULTY MENTOR NAME:**P.SUGANYA**

TEAM ID :**PNT2022TMID46943**

Submitted By

W.STEPHENRAJ 821919104027

J.JUSTINRAJ 821919104007

D.KARAN 821919104008

W.MERUNRAJ 821919104012

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

St. Joseph's College of Engineering & Technology

ANNA UNIVERSITY :: 2019 – 2023

ROJECT 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 KytoÂ¨jokib, Harri Niskaa, Olli BraÂ¨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, Â©2014 IEEE.

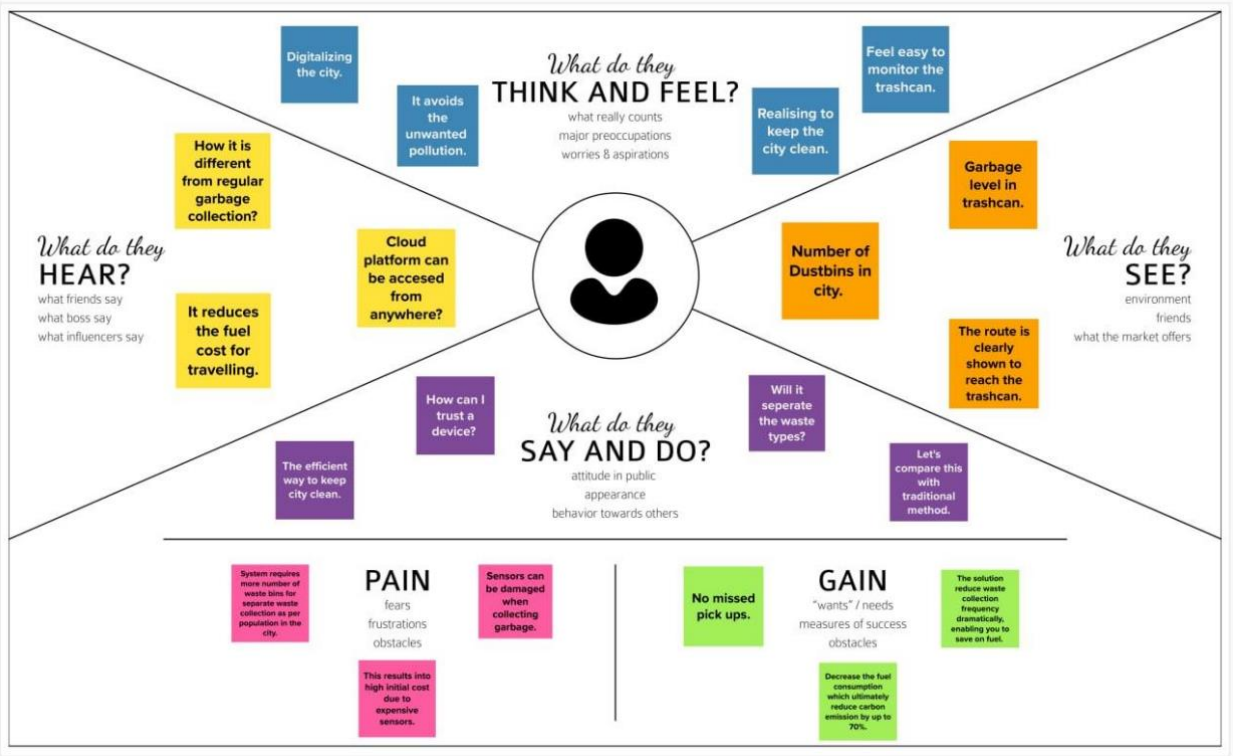
2.3.Problem Statement Definition:

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.

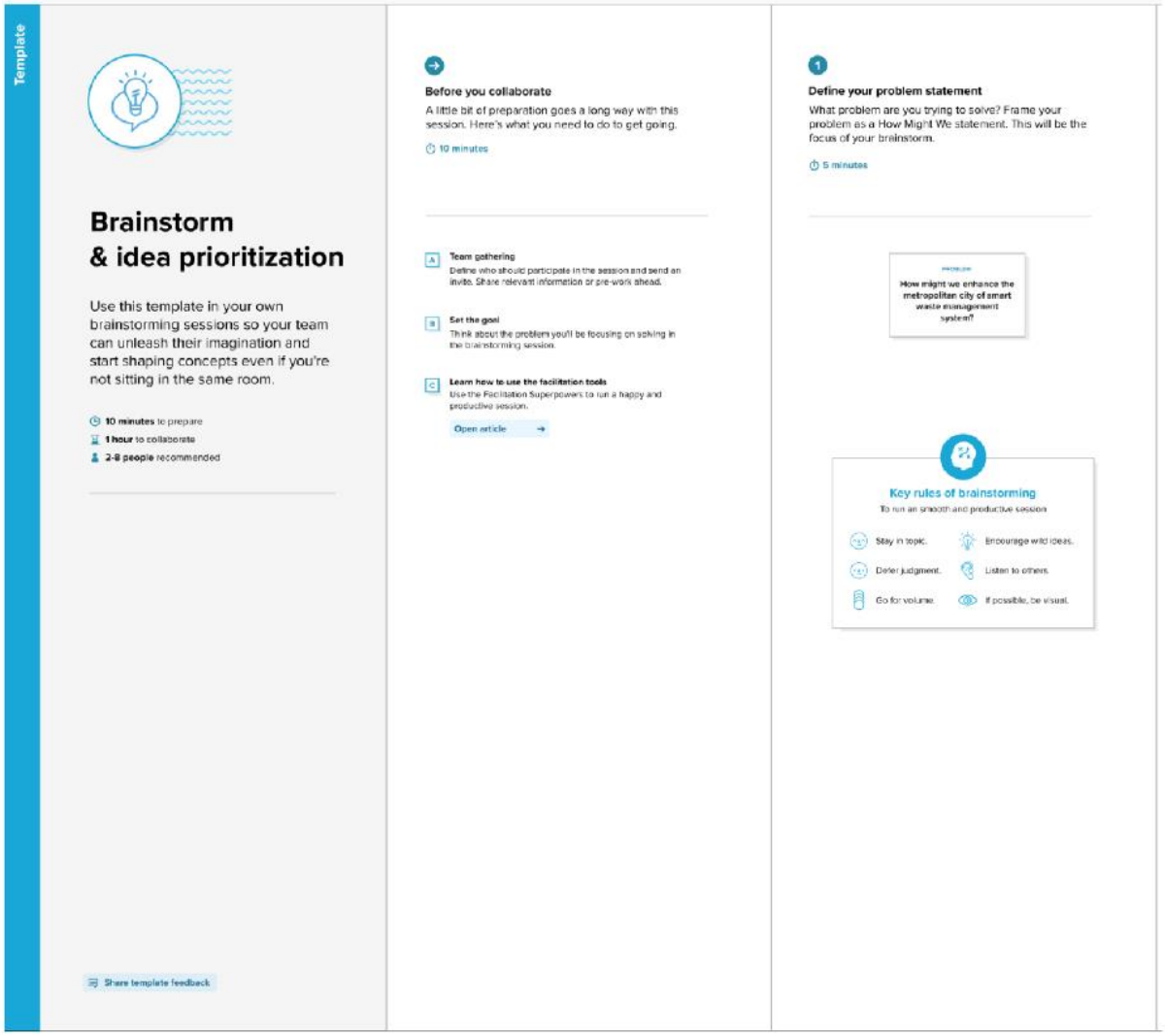
3. IDEATION & PROPOSED SOLUTION:

3.1. Empathy Map Canvas:

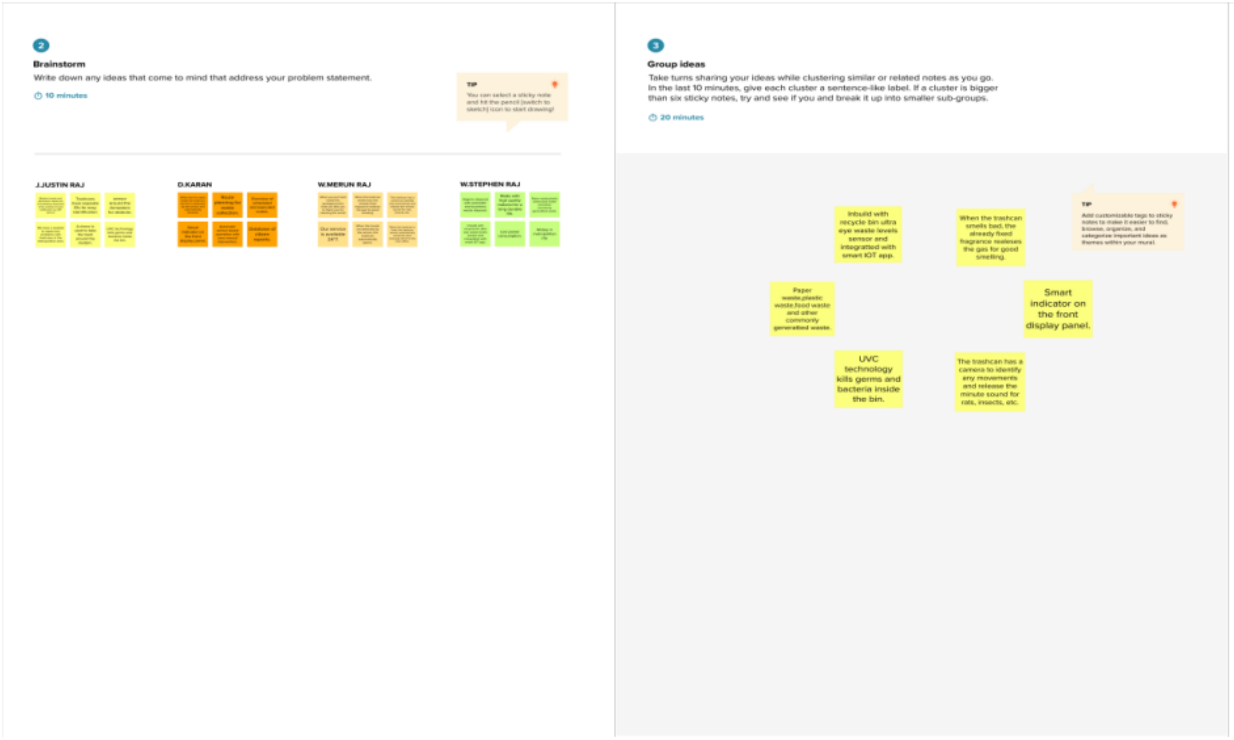


3.2. Ideation & Brainstorming:

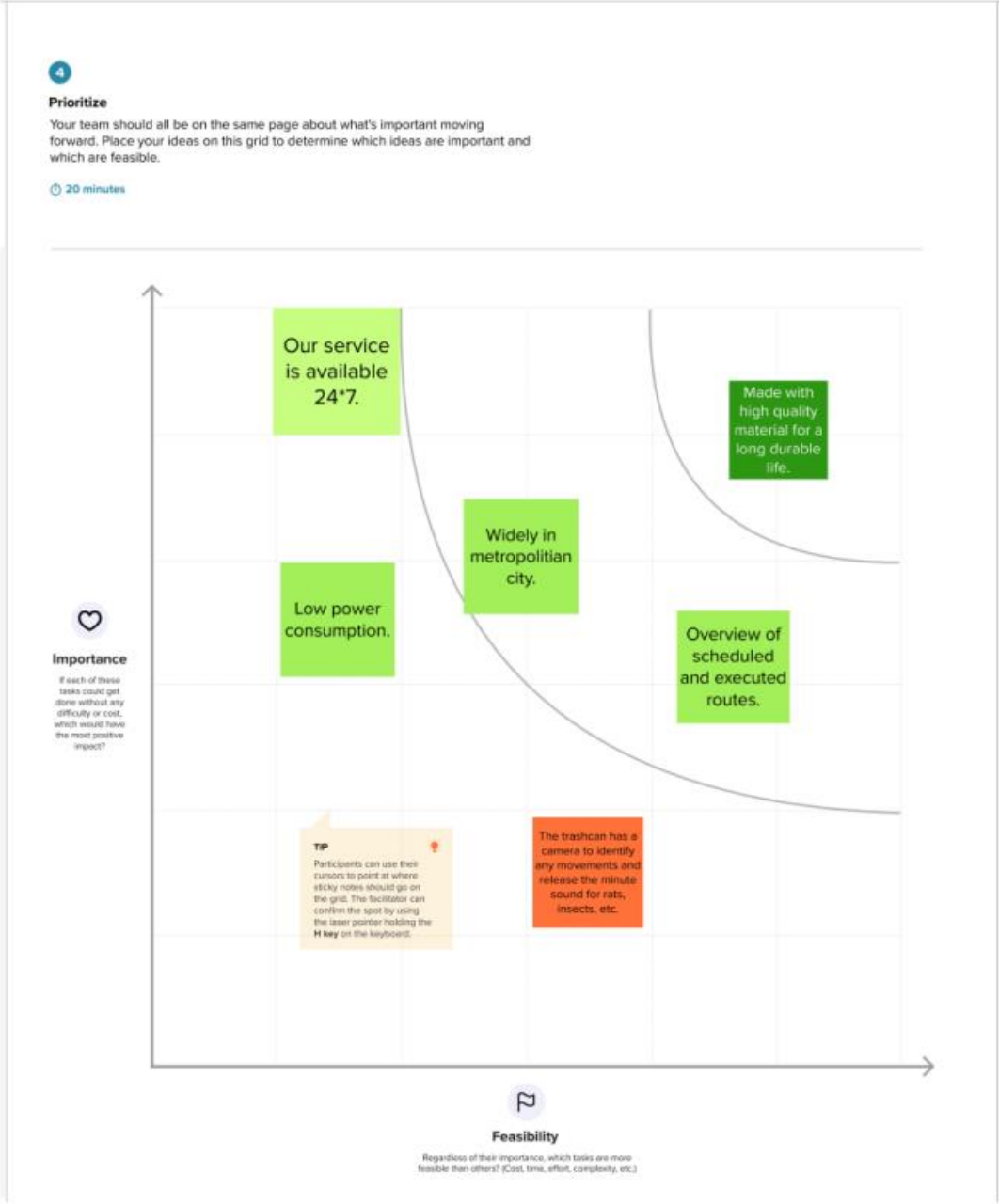
Step-1: Team Gathering, Collaboration and Select the Problem Statement:



Step-2: Brainstorm, Idea Listing and Grouping:



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

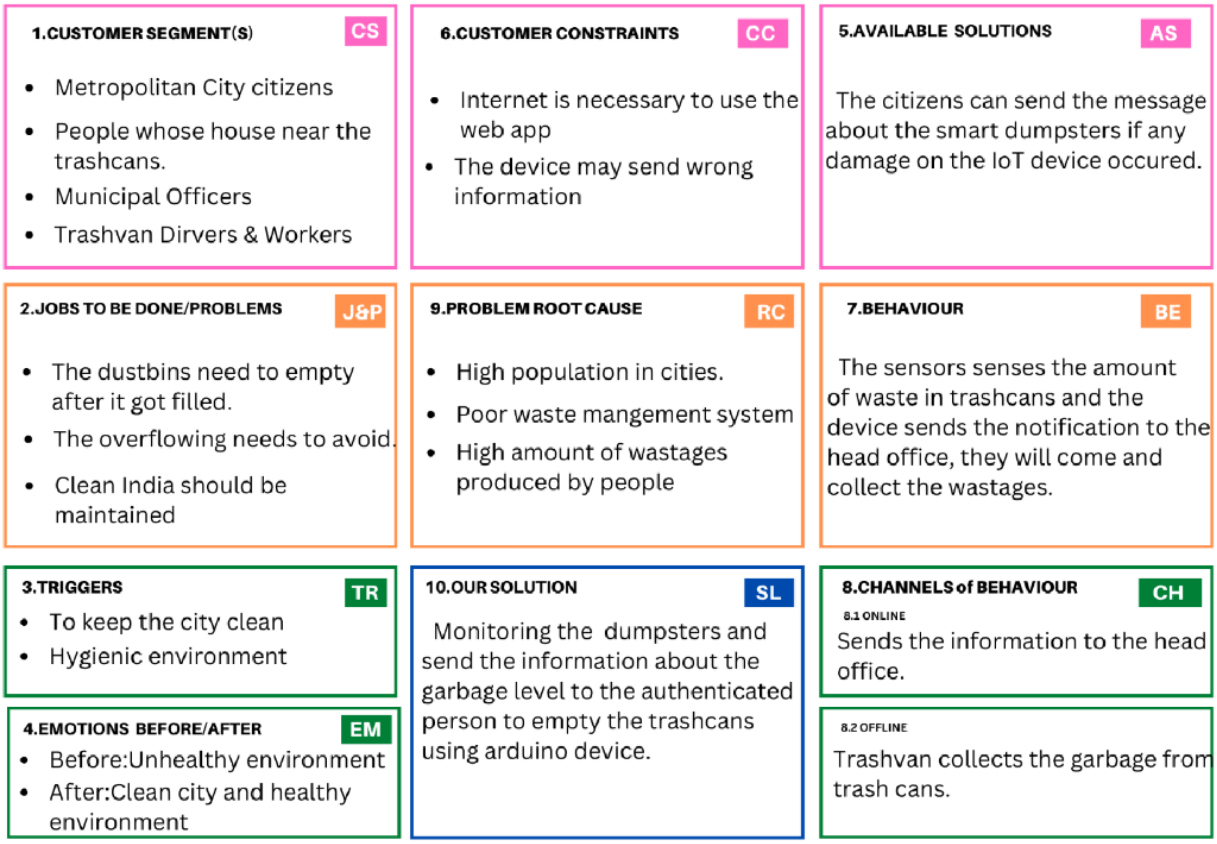
		collection is solved by designing an embedded IoT system that will monitor each dumpster individually for the amount of waste deposited. The IR sensor is used for detecting the presence of any waste the IR sensor used. The device is connected to the cloud. Whenever the bin gets filled, the message will 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.	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 6 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.
5	Business Model (Revenue Model).	• 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

		need to fix it. • The contribution of the municipality is necessary to make the project succeed in the market
6	Scalability of the Solution.	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

		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,

		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
NFR-2	Security	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.
NFR-3	Reliability	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

		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.
NFR-4	Performance	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.
NFR-5	Availability	Another purpose of this project is to make the proposed waste management system as

		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.

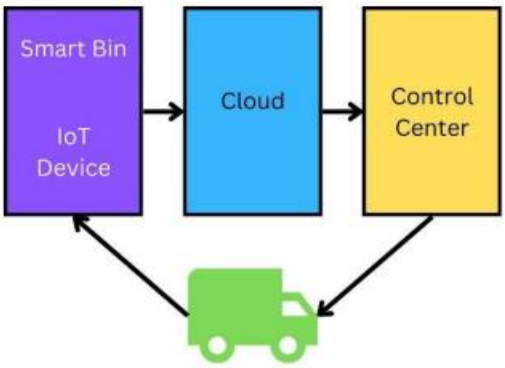
• The more number of bins needed in high populated area.

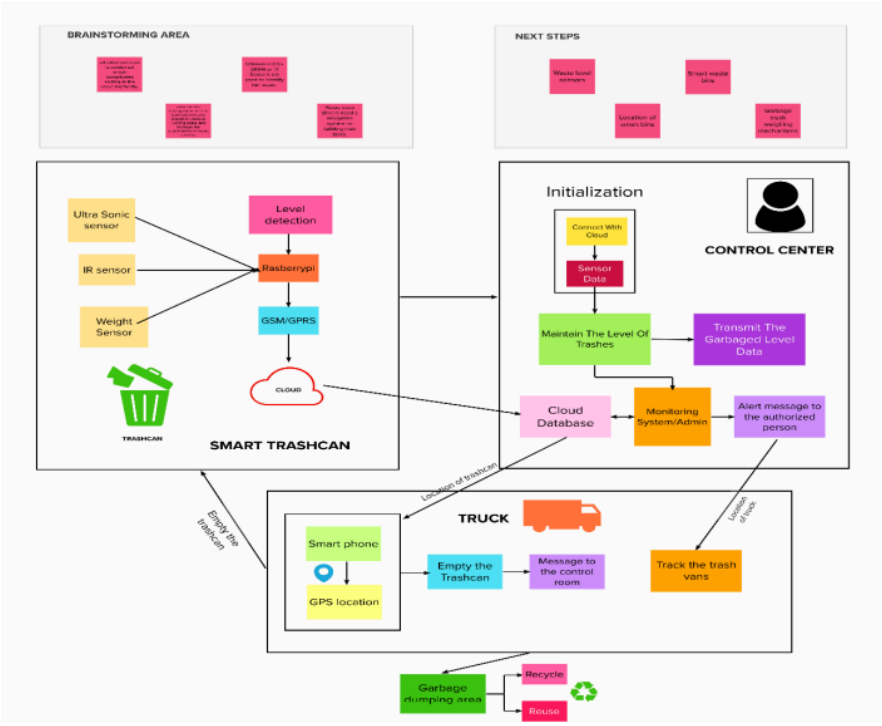
• The overflowing of trashcans can be avoided.

• 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:

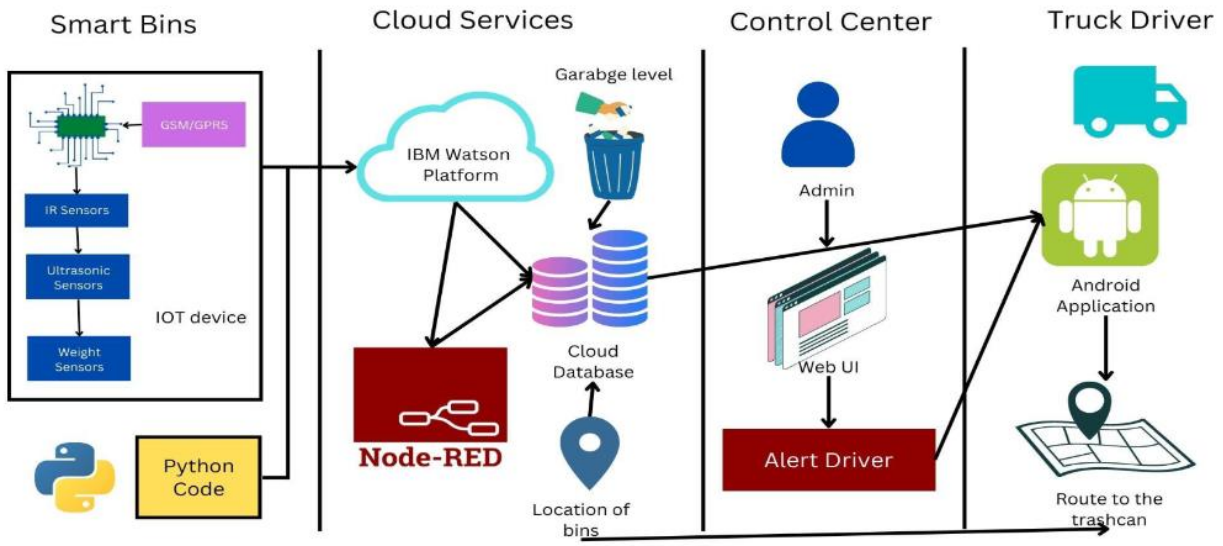


Table-1 : Components & Technologies:

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 municipality 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 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)

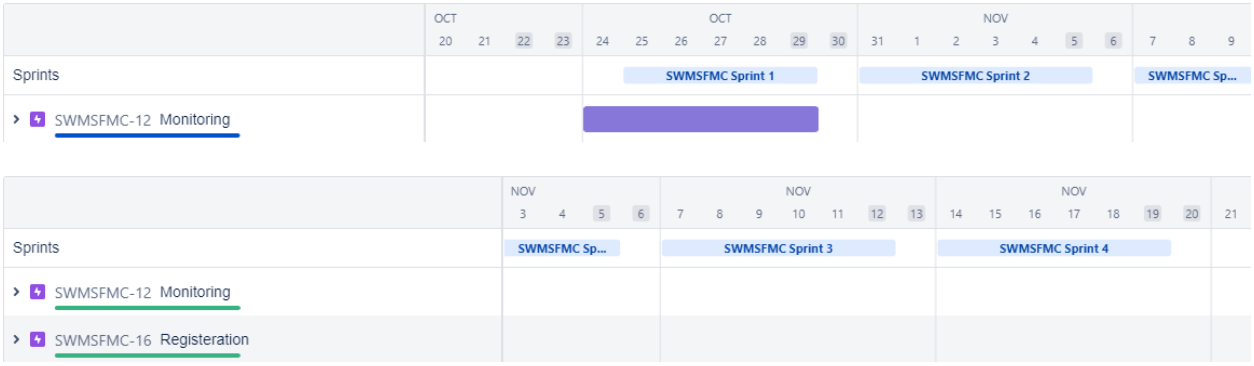
Sprint	**Total Story Points**	**Duration**	**Sprint Start Date**	**Sprint End Date (Planned)**	**Story Points Completed (as on Planned End Date)**	**Sprint Release Date (Actual)**
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct	05 Nov	20	05 Nov

			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).

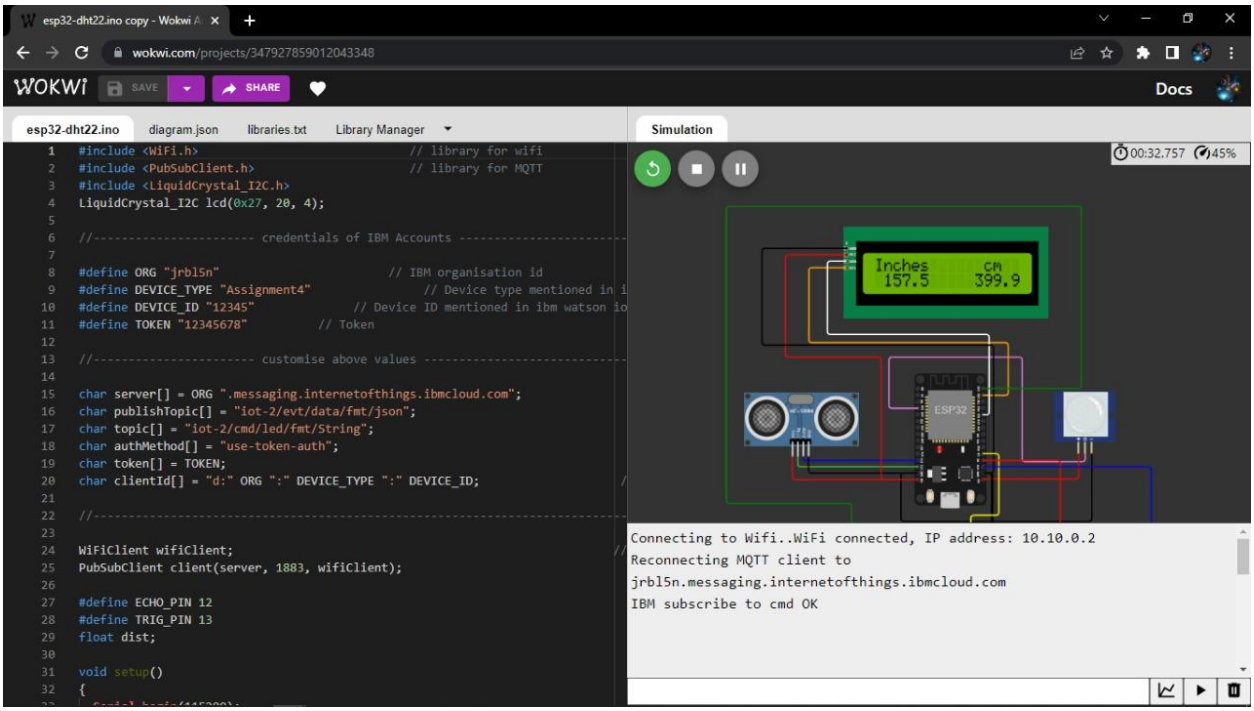
$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

6.3 Reports from JIRA:

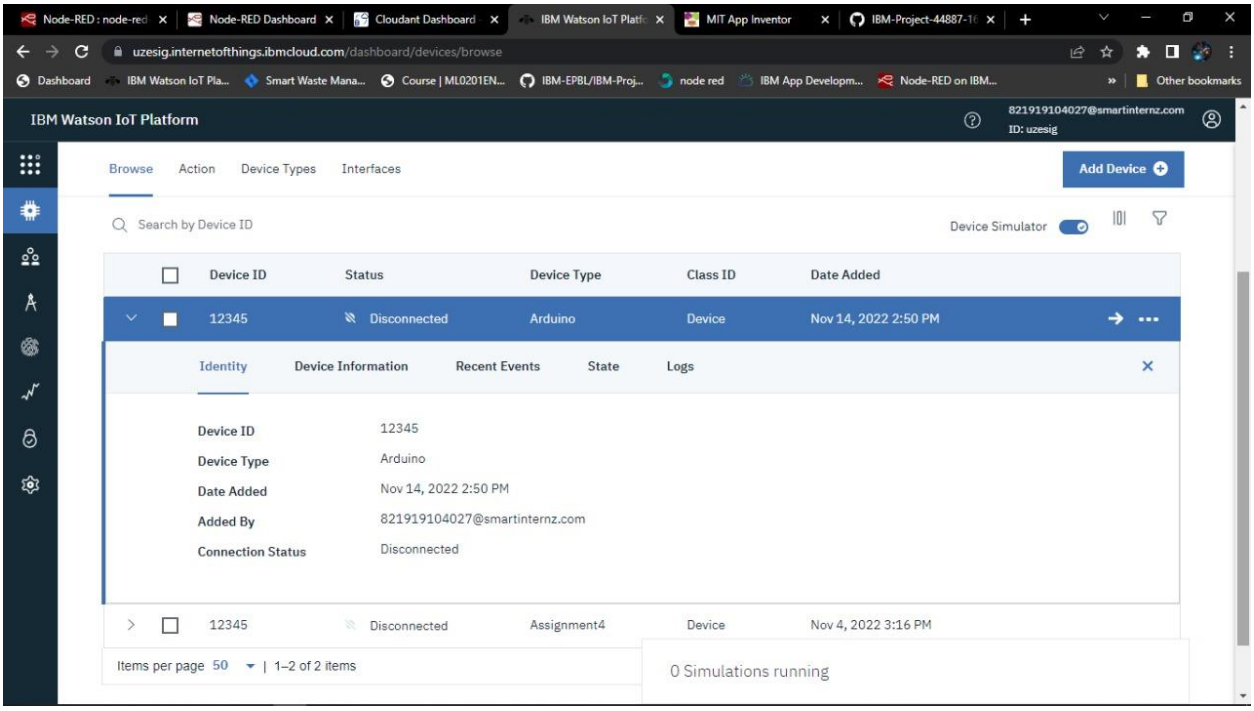


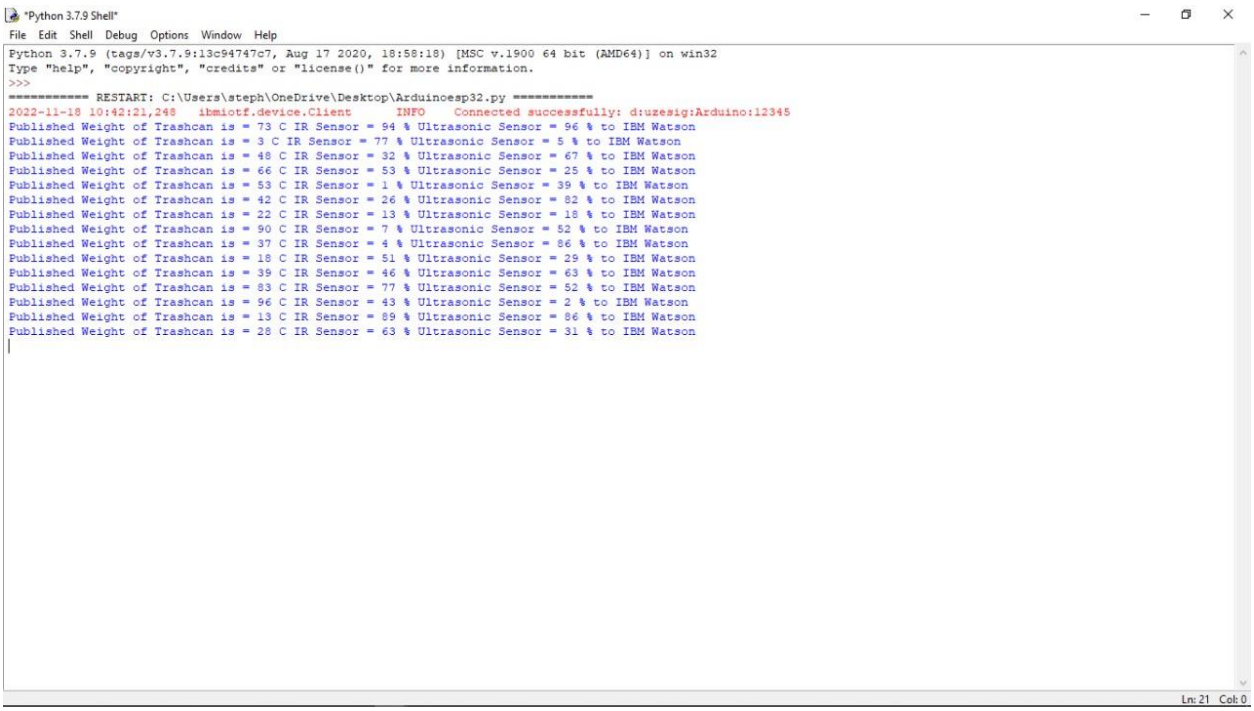
7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1.Wokwi:

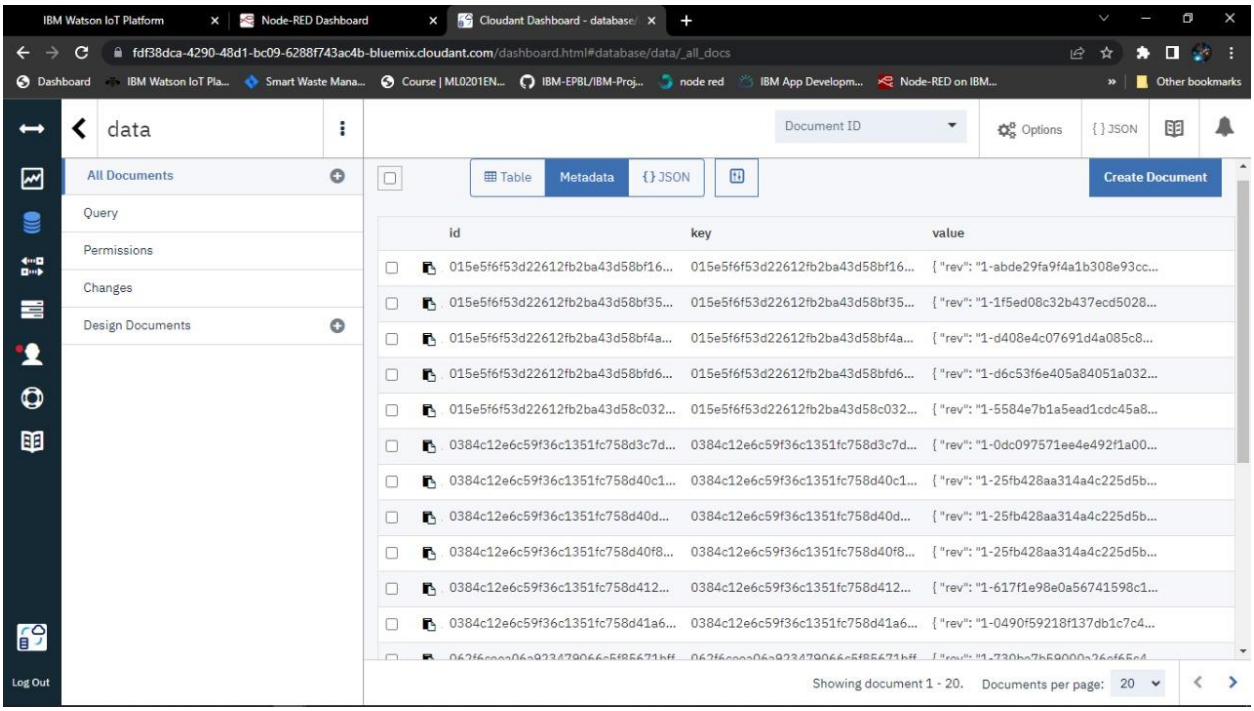

esp32-dht22.ino copy - Wokwi
wokwi.com/projects/347927859012043348
WOKWI
SAVE
SHARE
Docs
esp32-dht22.ino
diagram.json
libraries.txt
Library Manager
Simulation
00:32.757
45%
#include <WiFi.h> // library for wifi
#include <PubSubClient.h> // library for MQTT
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 20, 4);
//---------- credentials of IBM Accounts ----------
#define ORG "jrbl5n" // IBM organisation id
#define DEVICE_TYPE "Assignment4" // Device type mentioned in i
#define DEVICE_ID "12345" // Device ID mentioned in ibm watson io
#define TOKEN "12345678" // Token
//---------- customise above values ----------
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/data/fmt/json";
char topic[] = "iot-2/cmd/led/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
WiFiClient wifiClient;
PubSubClient client(server, 1883, wifiClient);
#define ECHO_PIN 12
#define TRIG_PIN 13
float dist;
void setup()
{
Inches
157.5
cm
399.9
ESP32
Connecting to Wifi..WiFi connected, IP address: 10.10.0.2
Reconnecting MQTT client to
jrbl5n.messaging.internetofthings.ibmcloud.com
IBM subscribe to cmd OK

7.2.Watson:

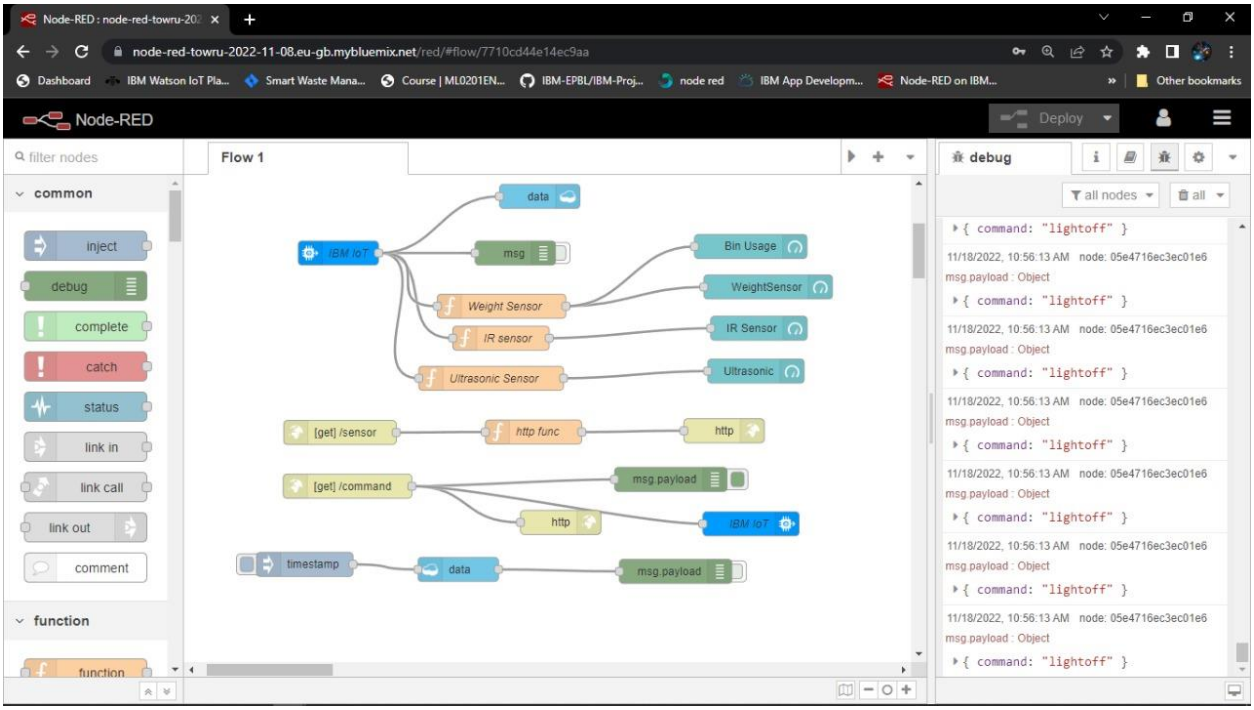

Node-RED : node-red
Node-RED Dashboard
Cloudant Dashboard
IBM Watson IoT Platf
MIT App Inventor
IBM-Project-44887-16
uzesig.internetofthings.ibmcloud.com/dashboard/devices/browse
Dashboard
IBM Watson IoT Pla...
Smart Waste Mana...
Course | ML0201EN...
IBM-EPBL/IBM-Proj...
node red
IBM App Developm...
Node-RED on IBM...
Other bookmarks
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
12345
Disconnected
Assignment4
Device
Nov 4, 2022 3:16 PM
Items per page 50
1–2 of 2 items
0 Simulations running

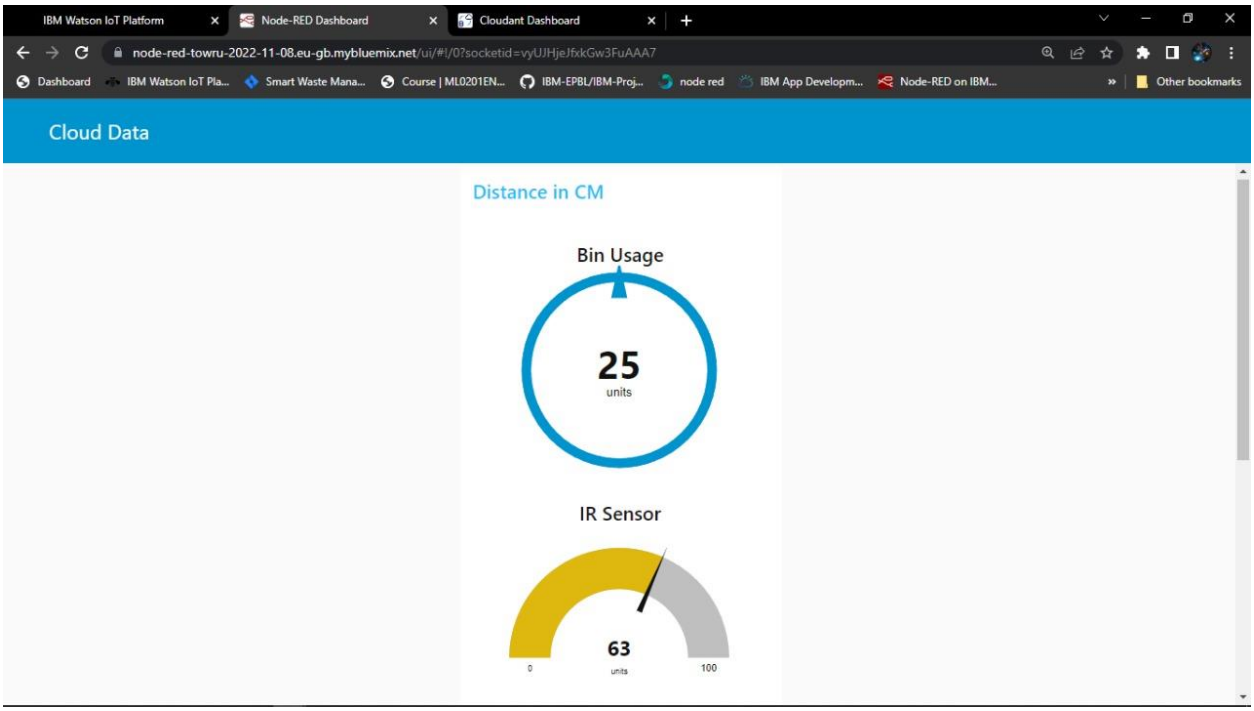

Python 3.7.9 Shell
File Edit Shell Debug Options Window Help
Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 18:58:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:\Users\steph\OneDrive\Desktop\Arduinoesp32.py ==========
2022-11-18 10:42:21,248 ibmiotf.device.Client INFO Connected successfully: d:uzesig: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

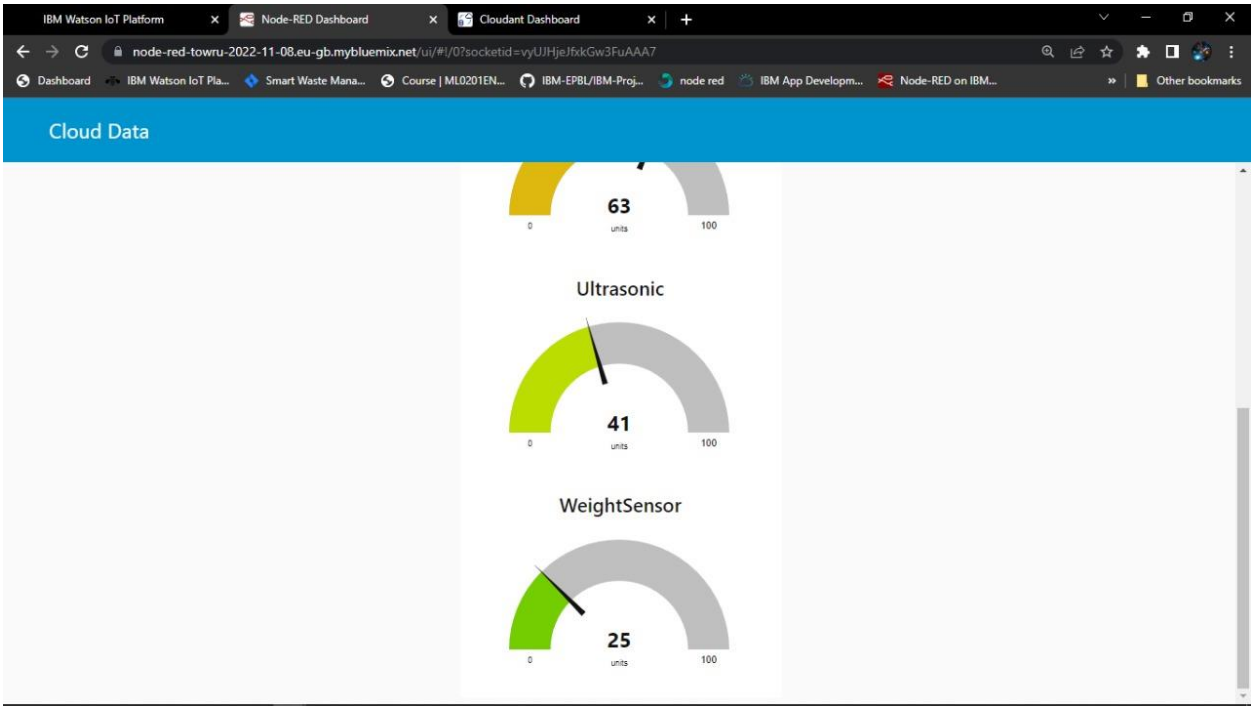
7.3.Cloudant:


IBM Watson IoT Platform
Node-RED Dashboard
Cloudant Dashboard - database
fdf38dca-4290-48d1-bc09-6288f743ac4b-bluemix.cloudant.com/dashboard.html#database/data/_all_docs
data
All Documents
Query
Permissions
Changes
Design Documents
Table
Metadata
{} JSON
Create Document
Document ID
Options
{ } JSON
id
key
value
Showing document 1 - 20.
Documents per page: 20
Log Out

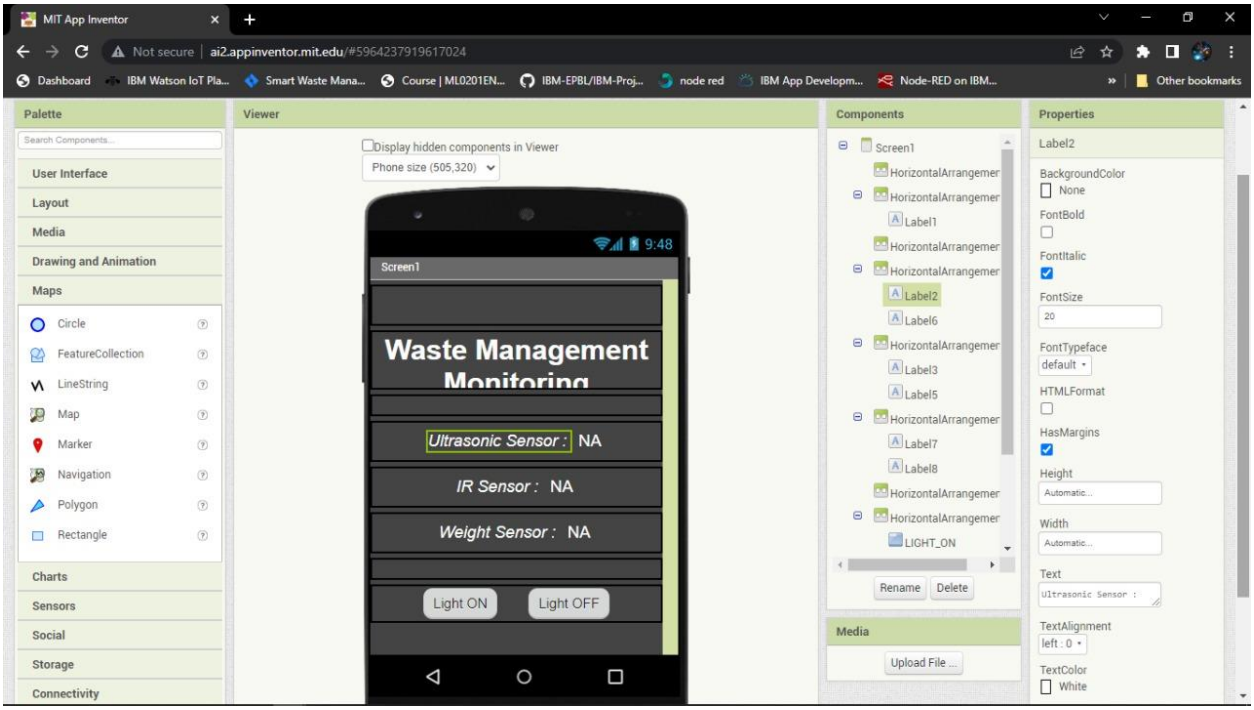
7.4.Nodered:

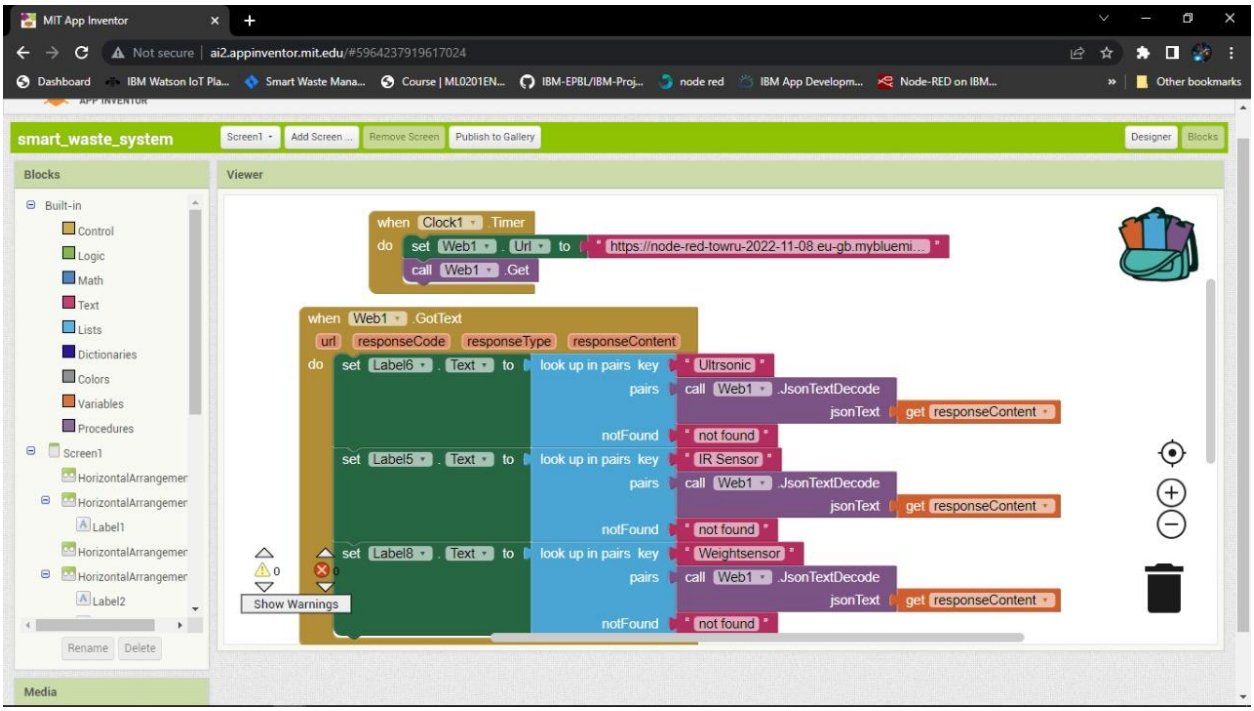

Node-RED : node-red-towru-202
node-red-towru-2022-11-08.eu-gb.mybluemix.net/red/#flow/7710cd44e14ec9aa
Node-RED
Deploy
filter nodes
Flow 1
common
inject
debug
complete
catch
status
link in
link call
link out
comment
function
data
IBM IoT
msg
Bin Usage
Weight Sensor
WeightSensor
IR sensor
IR Sensor
Ultrasonic Sensor
Ultrasonic
[get] /sensor
http func
http
[get] /command
msg.payload
http
IBM IoT
timestamp
data
msg.payload
debug
all nodes
all
{ command: "lightoff" }
11/18/2022, 10:56:13 AM node: 05e4716ec3ec01e6
msg.payload : Object

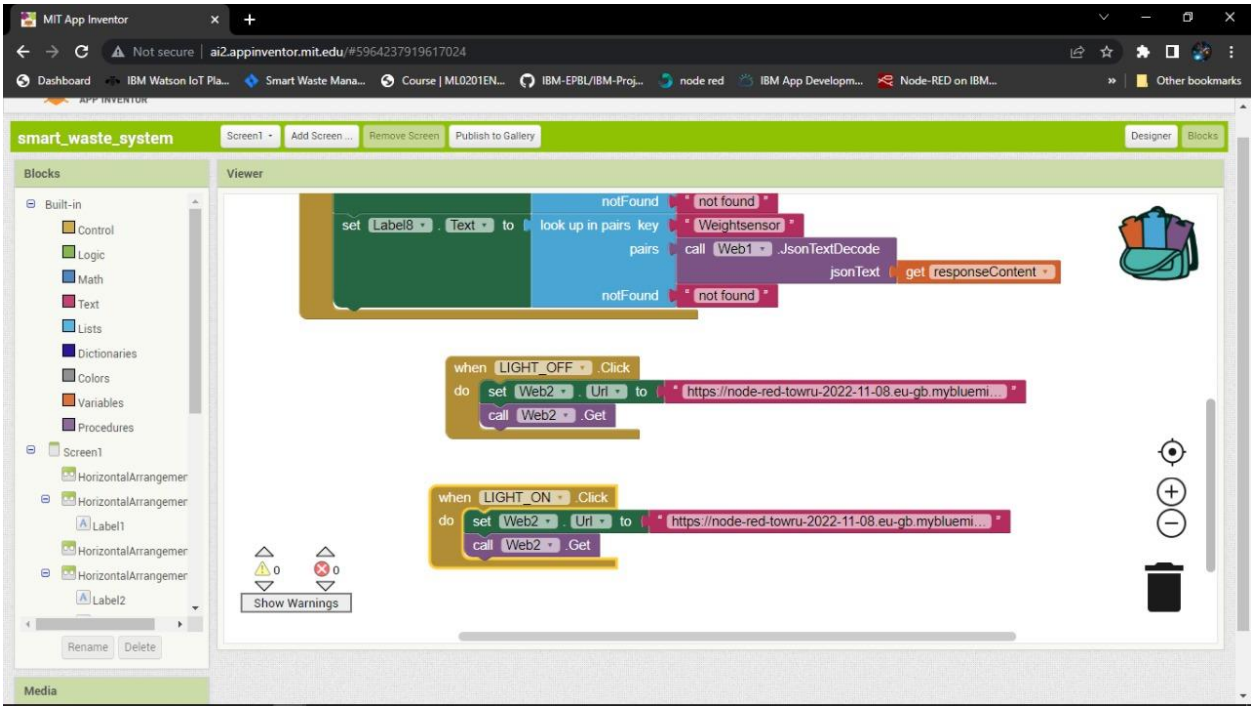

IBM Watson IoT Platform
Node-RED Dashboard
Cloudant Dashboard
node-red-towru-2022-11-08.eu-gb.mybluemix.net/ui/#!/0?socketid=vyUJHjeJfxkGw3FuAAA7
Dashboard
IBM Watson IoT Pla...
Smart Waste Mana...
Course | ML0201EN...
IBM-EPBL/IBM-Proj...
node red
IBM App Developm...
Node-RED on IBM...
Other bookmarks
Cloud Data
Distance in CM
Bin Usage
25
units
IR Sensor
63
units
0
100

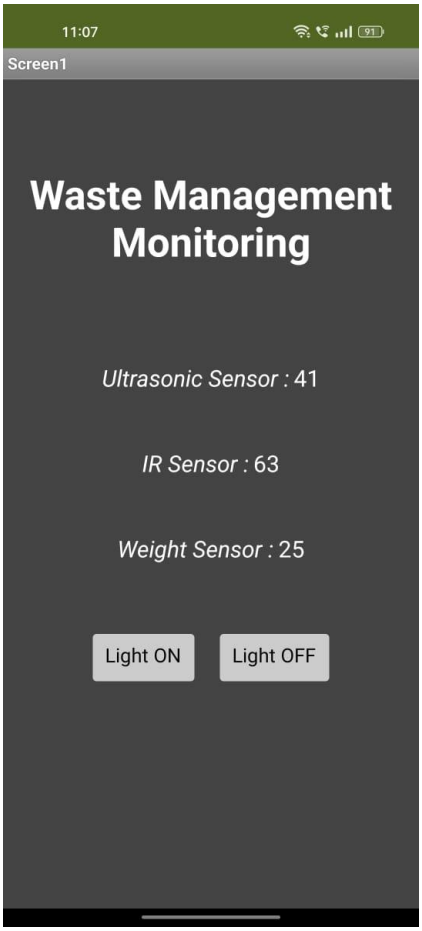

IBM Watson IoT Platform
Node-RED Dashboard
Cloudant Dashboard
node-red-towru-2022-11-08.eu-gb.mybluemix.net/ui/#!/0?socketid=vyUJHjeJfxkGw3FuAAA7
Dashboard
IBM Watson IoT Pla...
Smart Waste Mana...
Course | ML0201EN...
IBM-EPBL/IBM-Proj...
node red
IBM App Developm...
Node-RED on IBM...
Other bookmarks
Cloud Data
63
units
0
100
Ultrasonic
41
units
0
100
WeightSensor
25
units
0
100

7.5.MIT App:


MIT App Inventor
Not secure
ai2.appinventor.mit.edu/#5964237919617024
Palette
Viewer
Components
Properties
Display hidden components in Viewer
Phone size (505,320)
Screen1
Waste Management Monitoring
Ultrasonic Sensor : NA
IR Sensor : NA
Weight Sensor : NA
Light ON
Light OFF
Label2
Rename
Delete
Upload File ...

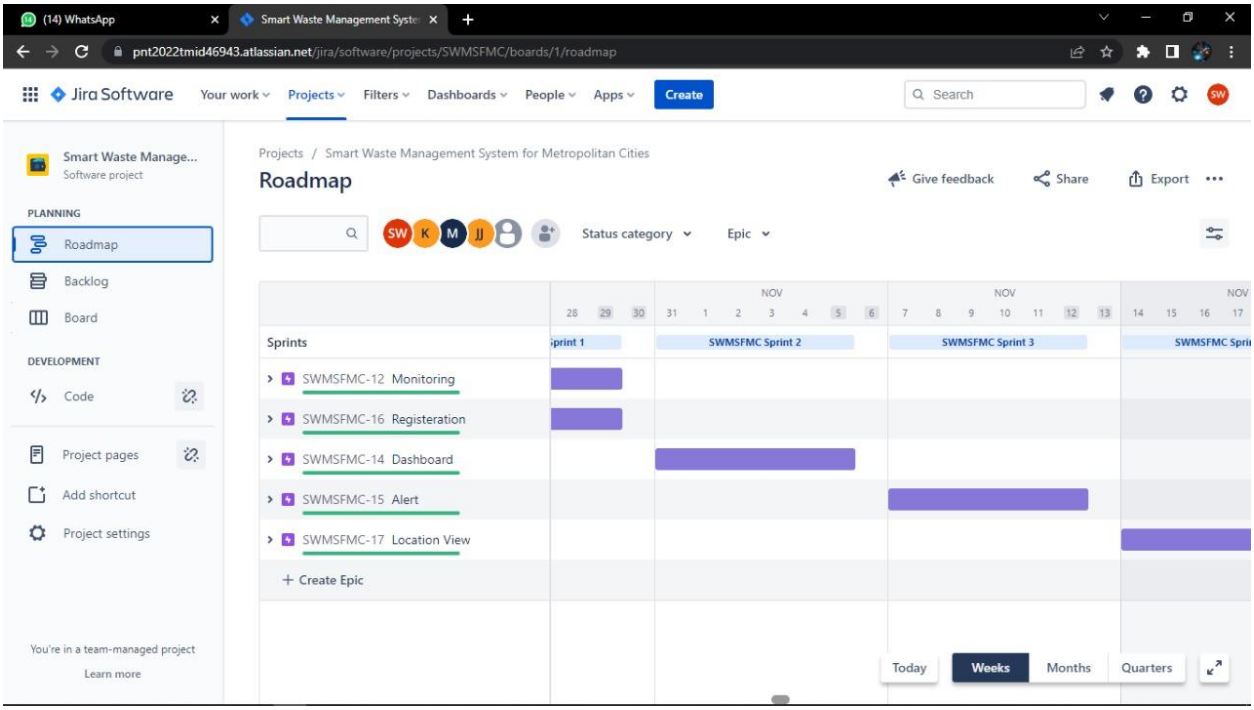

MIT App Inventor
Not secure | ai2.appinventor.mit.edu/#5964237919617024
Dashboard
IBM Watson IoT Pla...
Smart Waste Mana...
Course | ML0201EN...
IBM-EPBL/IBM-Proj...
node red
IBM App Developm...
Node-RED on IBM...
Other bookmarks
smart_waste_system
Screen1
Add Screen ...
Remove Screen
Publish to Gallery
Designer
Blocks
Viewer
Built-in
Control
Logic
Math
Text
Lists
Dictionaries
Colors
Variables
Procedures
Screen1
HorizontalArrangemer
Label1
Label2
Rename
Delete
Media
Show Warnings
when Clock1 .Timer
do set Web1 . Url to " https://node-red-towru-2022-11-08.eu-gb.mybluemi... "
call Web1 .Get
when Web1 .GotText
url responseCode responseType responseContent
do set Label6 . Text to look up in pairs key " Ultrsonic "
pairs call Web1 .JsonTextDecode
jsonText get responseContent
notFound " not found "
set Label5 . Text to look up in pairs key " IR Sensor "
pairs call Web1 .JsonTextDecode
jsonText get responseContent
notFound " not found "
set Label8 . Text to look up in pairs key " Weightsensor "
pairs call Web1 .JsonTextDecode
jsonText get responseContent
notFound " not found "


MIT App Inventor
Not secure | ai2.appinventor.mit.edu/#5964237919617024
Dashboard
IBM Watson IoT Pla...
Smart Waste Mana...
Course | ML0201EN...
IBM-EPBL/IBM-Proj...
node red
IBM App Developm...
Node-RED on IBM...
Other bookmarks
smart_waste_system
Screen1
Add Screen ...
Remove Screen
Publish to Gallery
Designer
Blocks
Viewer
Built-in
Control
Logic
Math
Text
Lists
Dictionaries
Colors
Variables
Procedures
Screen1
HorizontalArrangemer
Label1
Label2
Rename
Delete
Media
Show Warnings
notFound " not found "
set Label8 . Text to look up in pairs key " Weightsensor "
pairs call Web1 .JsonTextDecode
jsonText get responseContent
notFound " not found "
when LIGHT_OFF .Click
do set Web2 . Url to " https://node-red-towru-2022-11-08.eu-gb.mybluemi... "
call Web2 .Get
when LIGHT_ON .Click
do set Web2 . Url to " https://node-red-towru-2022-11-08.eu-gb.mybluemi... "
call Web2 .Get


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)
 {
```

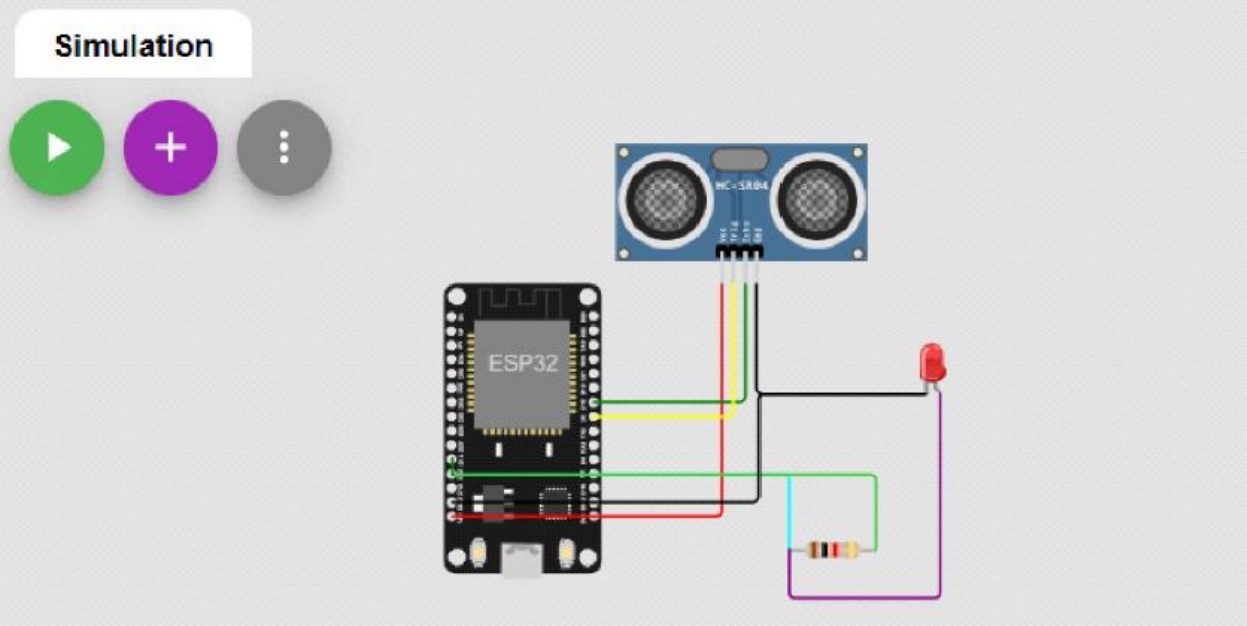
```
   digitalWrite(23, HIGH);
   Serial.println("Bin is available");
   digitalWrite(2,LOW);
   digitalWrite(4, LOW);

 }
   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);
```

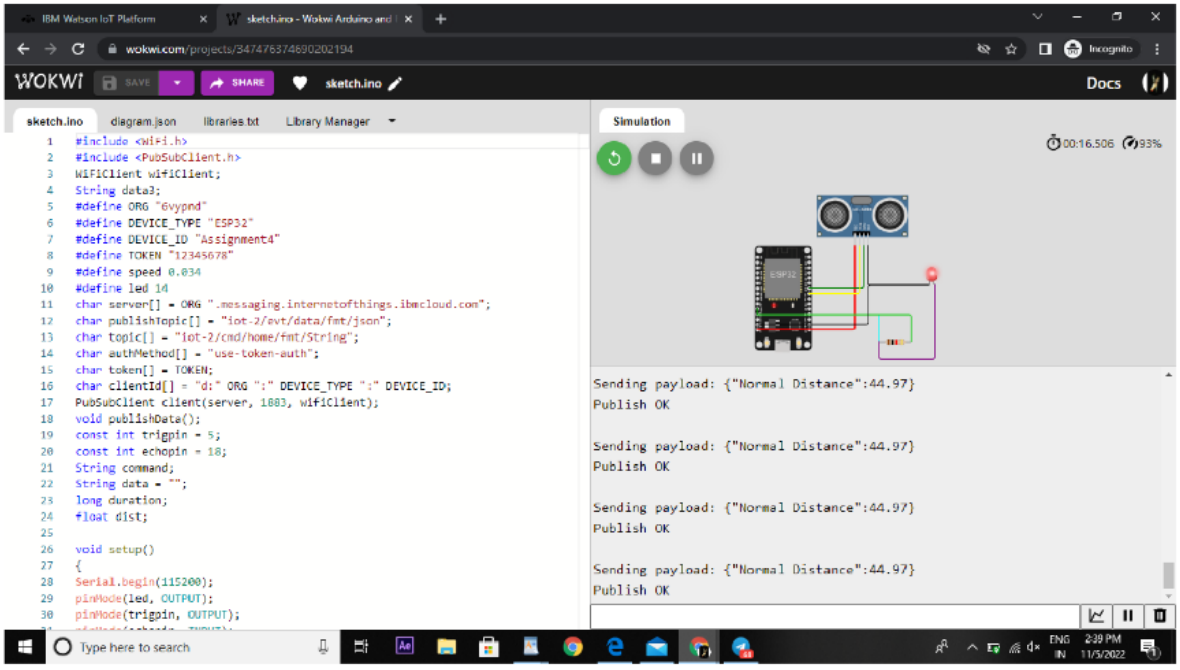
```
  }


  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);
```

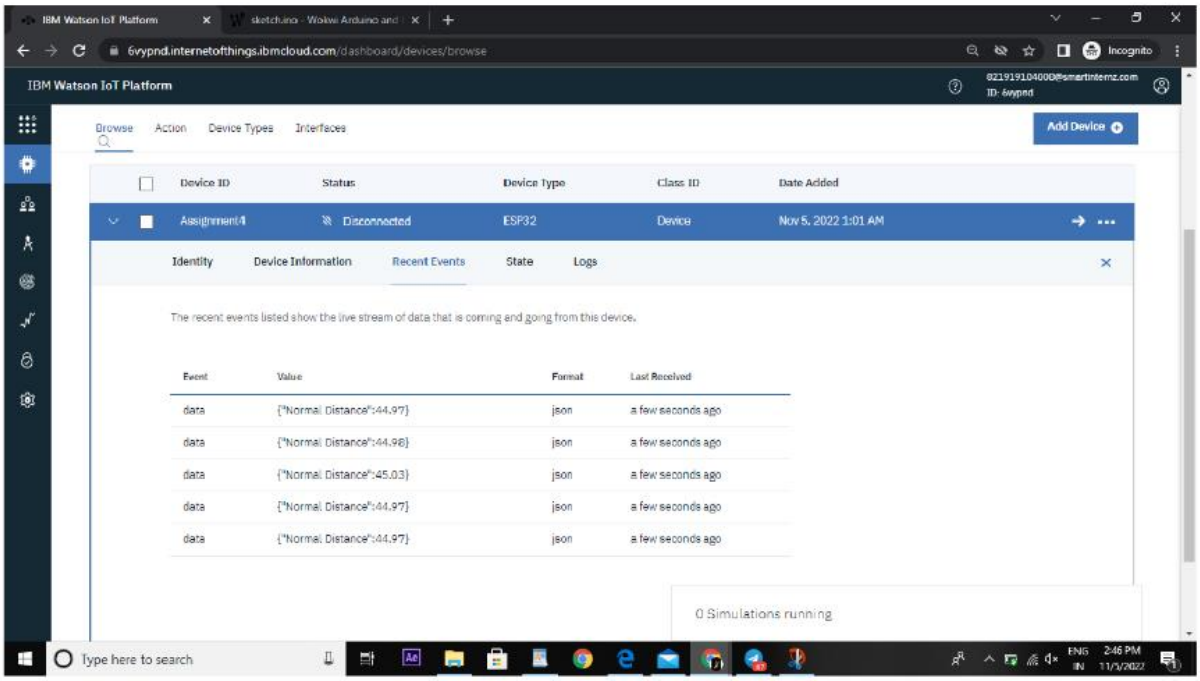
```
String payload = "{\"Warning\":";
payload += cm ;
payload += " }";
Serial.print("\n");
Serial.print("Sending payload: ");
Serial.println(payload);
if(client.publish(publishTopic, (char*) payload.c_str()))
{
Serial.println("Publish OK");
}
else
{
Serial.println("Publish FAILED");
}
}
else
{
Serial.println();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

  float inches = (cm / 2.54);                                  //print on lcd
  lcd.setCursor(0,0);
          lcd.print("Inches");
```

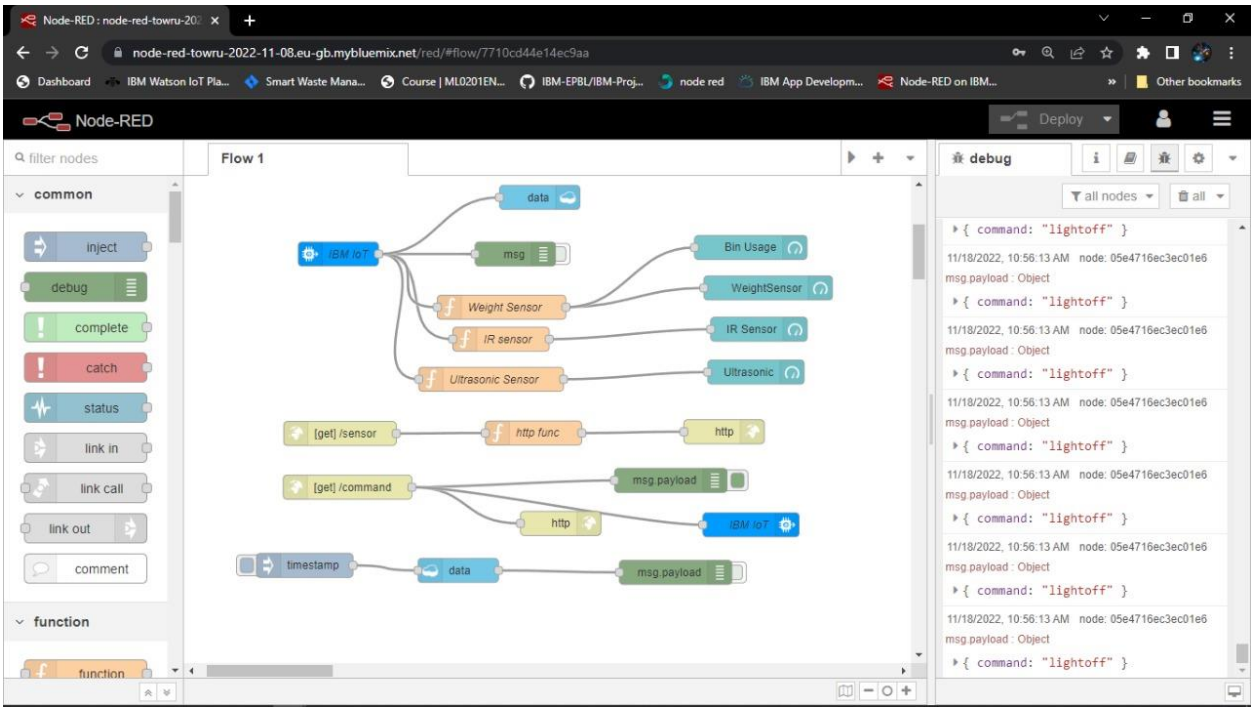
```
        lcd.setCursor(4,0);
        lcd.setCursor(12,0);
        lcd.print("cm");
        lcd.setCursor(1,1);
        lcd.print(inches, 1);
        lcd.setCursor(11,1);
        lcd.print(cm, 1);
        lcd.setCursor(14,1);
        delay(1000);
        lcd.clear();
}
```

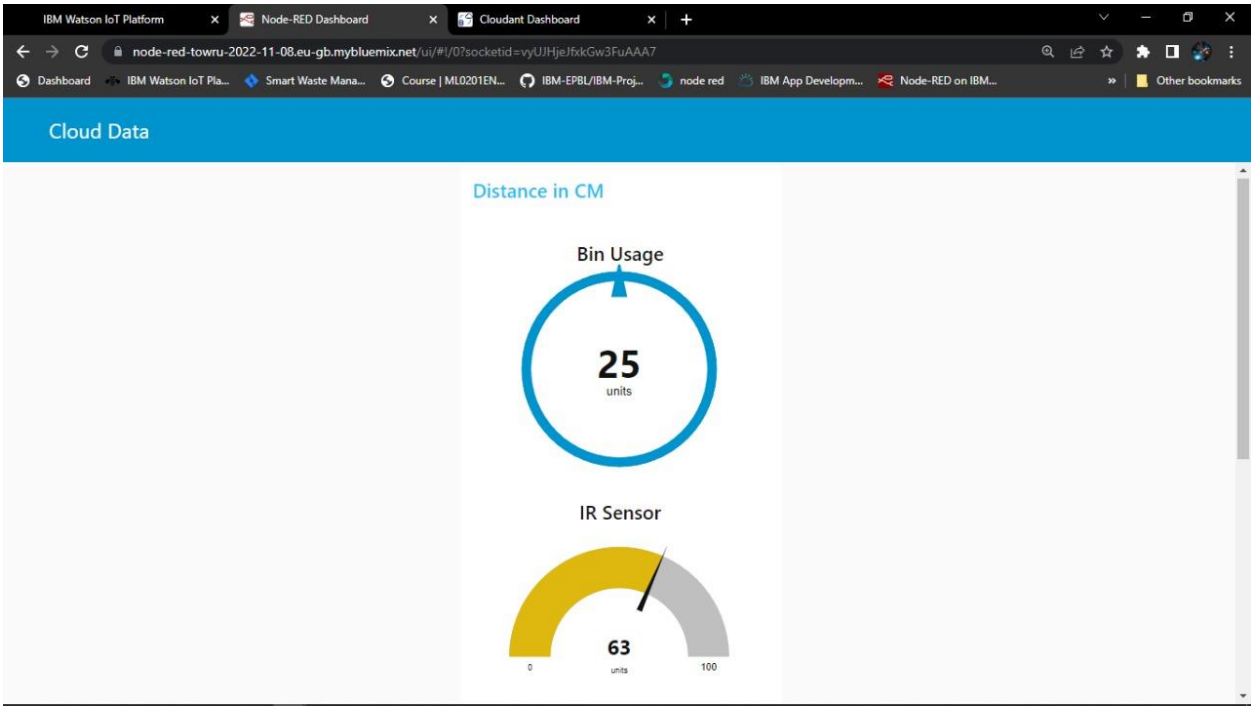
13. OUTPUT PICTURE:

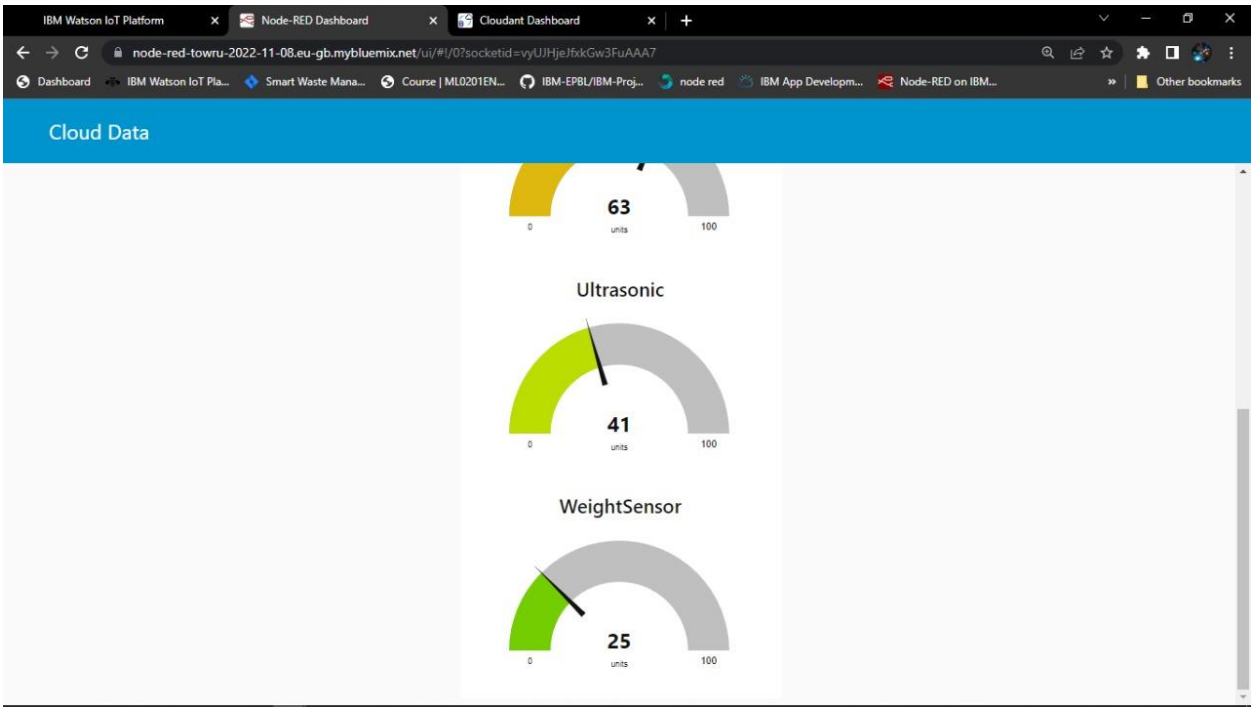


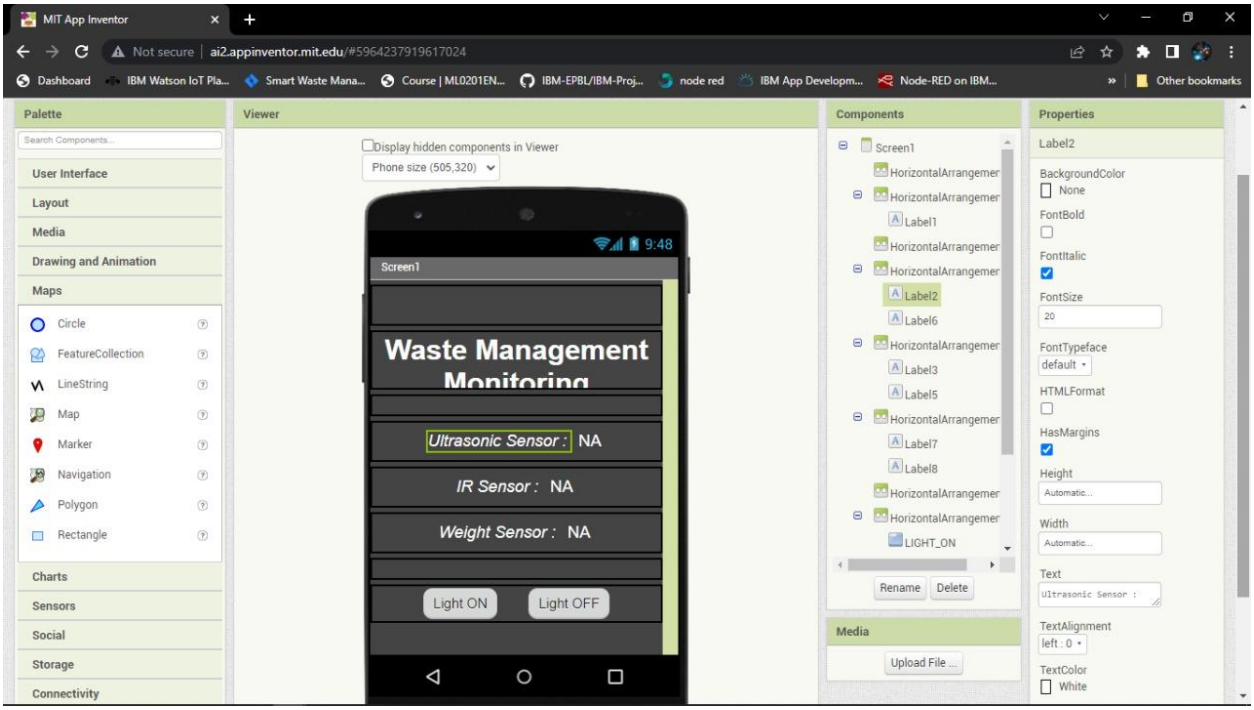

IBM Watson IoT Platform
sketch.ino - Wokwi Arduino and
wokwi.com/projects/347476374690202194
WOKWI
SAVE
SHARE
sketch.ino
Docs
sketch.ino
diagram.json
libraries.txt
Library Manager
#include <WiFi.h>
#include <PubSubClient.h>
WiFiClient wifiClient;
String data3;
#define ORG "6vypnd"
#define DEVICE_TYPE "ESP32"
#define DEVICE_ID "Assignment4"
#define TOKEN "12345678"
#define speed 0.034
#define led 14
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/data/fmt/json";
char topic[] = "iot-2/cmd/home/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
PubSubClient client(server, 1883, wifiClient);
void publishData();
const int trigpin = 5;
const int echopin = 18;
String command;
String data = "";
long duration;
float dist;
void setup()
{
Serial.begin(115200);
pinMode(led, OUTPUT);
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
ENG
IN
2:39 PM
11/5/2022

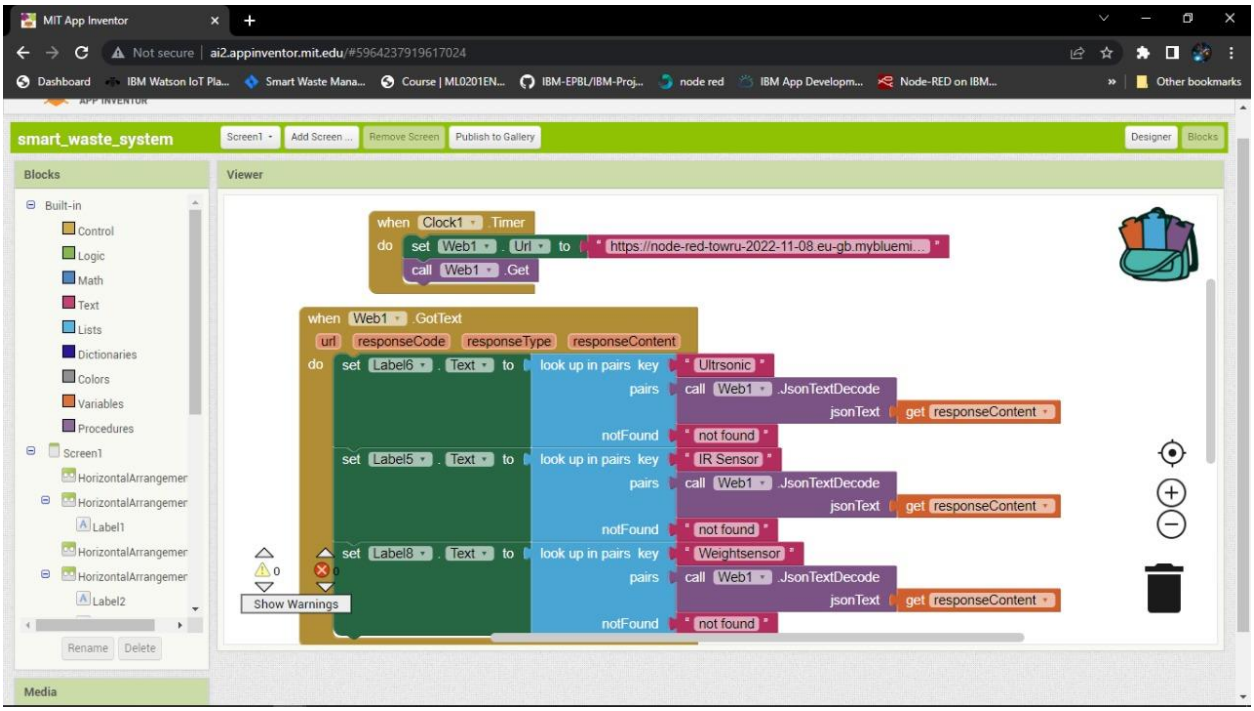

IBM Watson IoT Platform
6vypnd.internetofthings.ibmcloud.com/dashboard/devices/browse
IBM Watson IoT Platform
Browse
Action
Device Types
Interfaces
Add Device
Device ID
Status
Device Type
Class ID
Date Added
Assignment4
Disconnected
ESP32
Device
Nov 5, 2022 1:01 AM
Identity
Device Information
Recent 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
ENG
IN
2:46 PM
11/5/2022

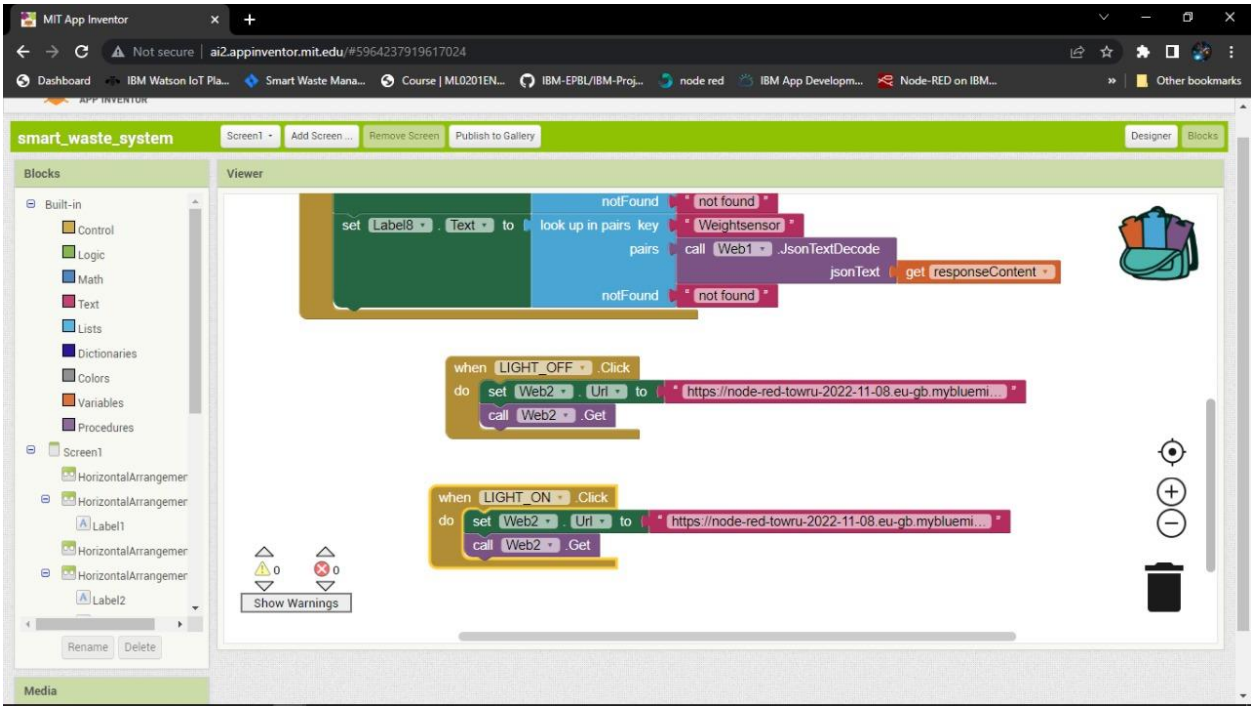

Node-RED
Deploy
Flow 1
debug
common
inject
debug
complete
catch
status
link in
link call
link out
comment
function
data
IBM IoT
msg
Bin Usage
Weight Sensor
WeightSensor
IR sensor
IR Sensor
Ultrasonic Sensor
Ultrasonic
[get] /sensor
http func
http
[get] /command
msg.payload
http
IBM IoT
timestamp
data
msg.payload
{ command: "lightoff" }
11/18/2022, 10:56:13 AM node: 05e4716ec3ec01e6
msg.payload : Object

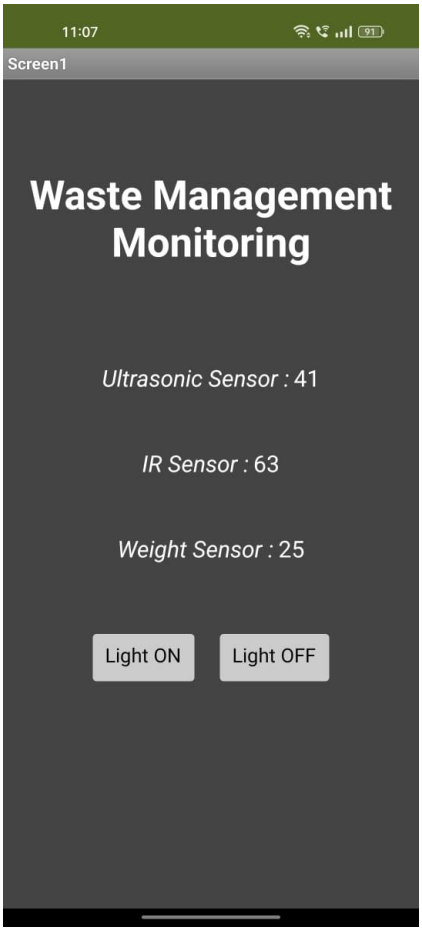

IBM Watson IoT Platform
Node-RED Dashboard
Cloudant Dashboard
Cloud Data
Distance in CM
Bin Usage
25
units
IR Sensor
63
units
0
100


IBM Watson IoT Platform
Node-RED Dashboard
Cloudant Dashboard
node-red-towru-2022-11-08.eu-gb.mybluemix.net/ui/#!/0?socketid=vyUJHjeJfxkGw3FuAAA7
Cloud Data
63
Ultrasonic
41
WeightSensor
25


MIT App Inventor
ai2.appinventor.mit.edu/#5964237919617024
Palette
Viewer
Components
Properties
Screen1
Waste Management Monitoring
Ultrasonic Sensor : NA
IR Sensor : NA
Weight Sensor : NA
Light ON
Light OFF


MIT App Inventor
Not secure | ai2.appinventor.mit.edu/#5964237919617024
Dashboard
IBM Watson IoT Pla...
Smart Waste Mana...
Course | ML0201EN...
IBM-EPBL/IBM-Proj...
node red
IBM App Developm...
Node-RED on IBM...
Other bookmarks
smart_waste_system
Screen1
Add Screen ...
Remove Screen
Publish to Gallery
Designer
Blocks
Viewer
Built-in
Control
Logic
Math
Text
Lists
Dictionaries
Colors
Variables
Procedures
Screen1
HorizontalArrangemer
Label1
Label2
Rename
Delete
Media
Show Warnings
when Clock1 .Timer
do set Web1 . Url to " https://node-red-towru-2022-11-08.eu-gb.mybluemi... "
call Web1 .Get
when Web1 .GotText
url responseCode responseType responseContent
do set Label6 . Text to look up in pairs key " Ultrsonic "
pairs call Web1 .JsonTextDecode
jsonText get responseContent
notFound " not found "
set Label5 . Text to look up in pairs key " IR Sensor "
pairs call Web1 .JsonTextDecode
jsonText get responseContent
notFound " not found "
set Label8 . Text to look up in pairs key " Weightsensor "
pairs call Web1 .JsonTextDecode
jsonText get responseContent
notFound " not found "


MIT App Inventor
Not secure | ai2.appinventor.mit.edu/#5964237919617024
Dashboard
IBM Watson IoT Pla...
Smart Waste Mana...
Course | ML0201EN...
IBM-EPBL/IBM-Proj...
node red
IBM App Developm...
Node-RED on IBM...
Other bookmarks
smart_waste_system
Screen1
Add Screen ...
Remove Screen
Publish to Gallery
Designer
Blocks
Viewer
Built-in
Control
Logic
Math
Text
Lists
Dictionaries
Colors
Variables
Procedures
Screen1
HorizontalArrangemer
Label1
Label2
Rename
Delete
Media
Show Warnings
notFound " not found "
set Label8 . Text to look up in pairs key " Weightsensor "
pairs call Web1 .JsonTextDecode
jsonText get responseContent
notFound " not found "
when LIGHT_OFF .Click
do set Web2 . Url to " https://node-red-towru-2022-11-08.eu-gb.mybluemi... "
call Web2 .Get
when LIGHT_ON .Click
do set Web2 . Url to " https://node-red-towru-2022-11-08.eu-gb.mybluemi... "
call Web2 .Get



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