SMART WASTE MANAGEMENT SYSTEM FOR

METROPOLITAN CITIESDOMAIN – INTERNET OF THINGS

(IoT)

A PROJECT REPORT

Team Members	Sathya priya B Sheeba G Snekha N Sabarivasan B Sir Rajam M
Team ID	PNT2022TMID36201

CONTENT :

10. PROJECT PLANNING PHASE

Prepare Milestone & Activity List
Sprint Delivery Plan

11. PROJECT DEVELOPMENT PHASE

Project Development-Delivery Of Sprint-1
Project Development-Delivery Of Sprint-2
Project Development-Delivery Of Sprint-3
Project Development-Delivery Of Sprint-4

12. FINAL PROJECT REPORT

13. RESULTS

14. ADVANTAGES & DISADVANTAGES

15. CONCLUSION

16. APPENDIX

Source Code

GitHub & Project Demo Link

1. INTRODUCTION

Internet of Things is nothing but the applications performing with the help of internet access. IoT Communication over the internet has grown from user - user interaction to device – device interactions these days. The IoT concepts were proposed years back but still it's in the initial stage of commercial deployment. Home automation industry and transportation industries are seeing rapid growth with IoT. The basic project idea is to design a smart waste detection system which would automatically notify the officials about the current status of various garbage bins in the city, would have realtime monitoring capabilities, which would be remotely controlled using IoT techniques. This paper introduces you to the use of IoT on one such area, that is, Garbage Detection in smart ways using IoT and see how this can also be a major part of developing a city into a smart city.

Project Overview

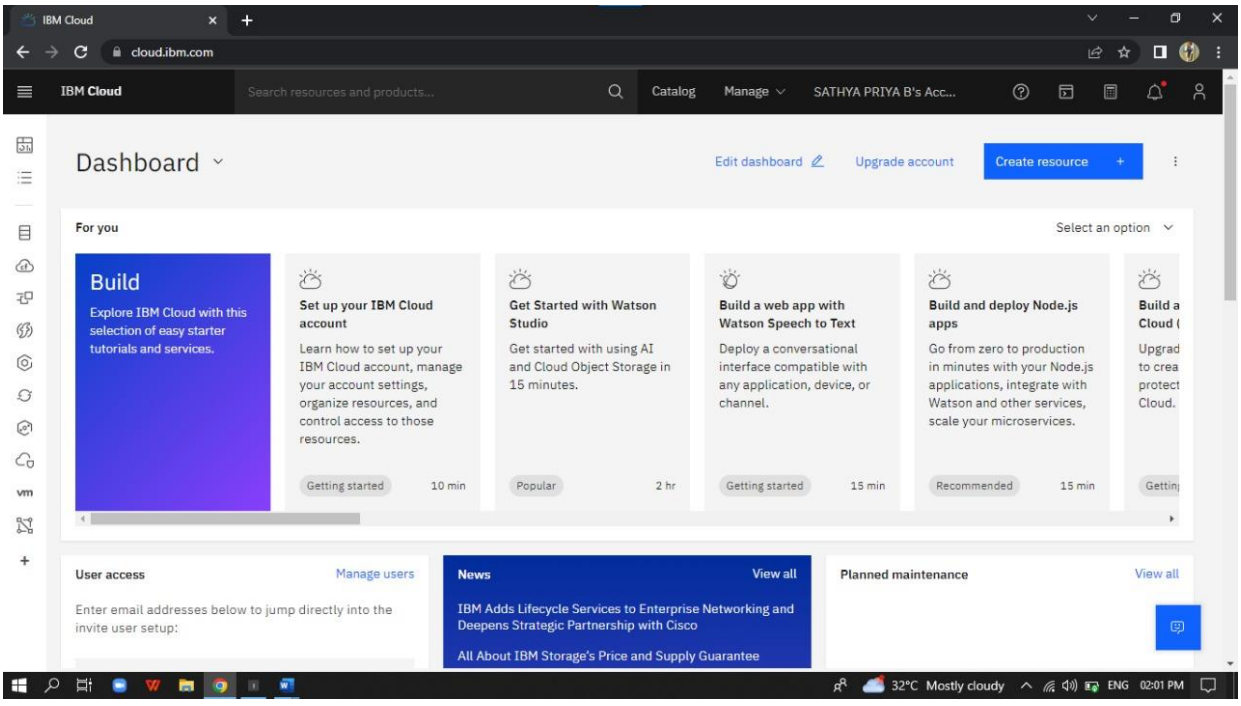
A big challenge in the urban cities is that of waste management as there is a rapid growth in the rate of urbanization and thus there is a need of sustainable urban development plans. As the concept of smart cities is very much trending these days and the smart cities cannot be complete without smart waste management system. There needs to be system that gives prior information of the filling of the bin that alerts the municipality so that they can clean the bin on time and safeguard the environment. To avoid all such situations we intend to propose a solution for this problem "Smart Garbage Bin", which will alarm and inform the authorized person when the garbage bin is about to fill. Then message will be send to the authorized person to collect the garbage from the particular area. The authorized person will sends the message from his web application to the garbage collectors by sending a SMS .This system maintain a dry waste and a wet waste separately. This will help to reduce the overflow of the garbage bin and thus keeping the environment clean.

Purpose

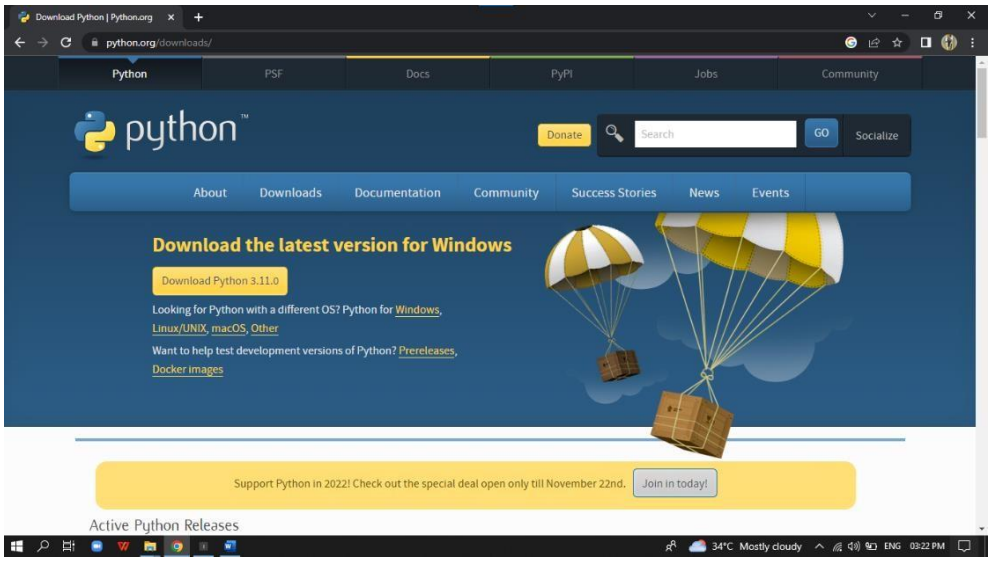
This project helps the citizens to make their surroundings and environment clean , pollution free and lead a healthy life throughout. It avoids the possibility garbage overflow, unhygienic environment, air-borne and water-borne disease , etc...

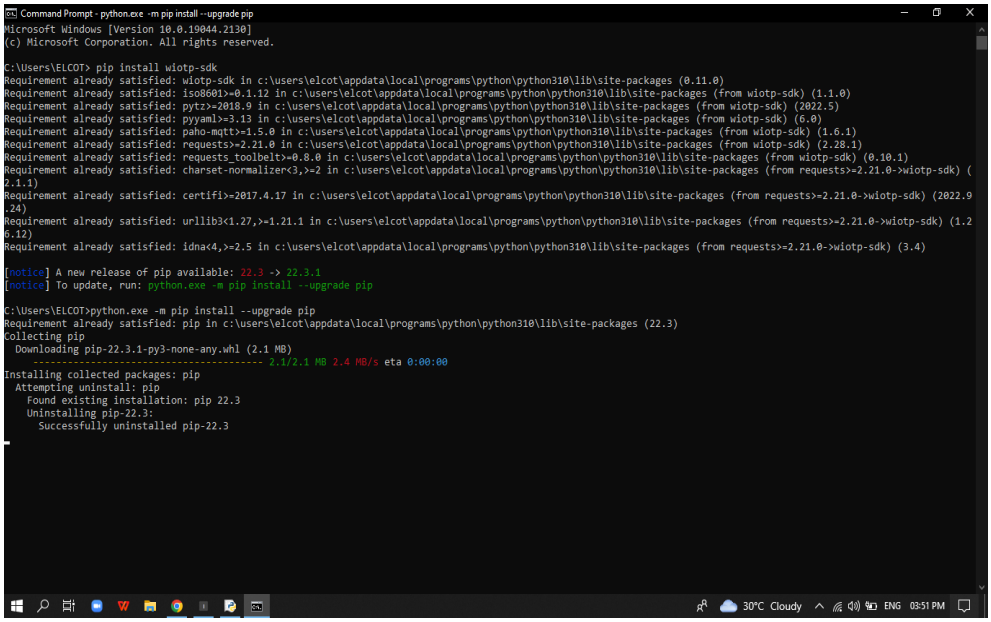
2. PREREQUISITES

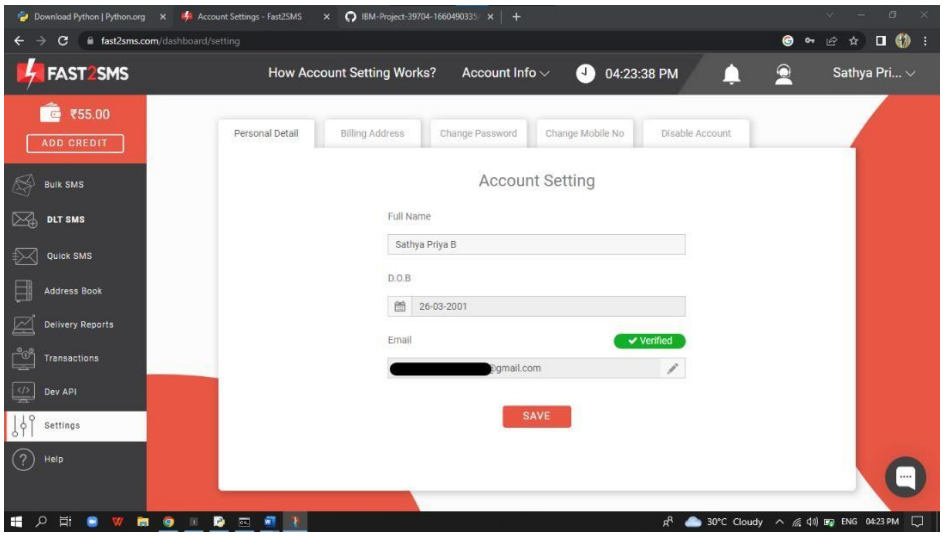
IBM cloud services



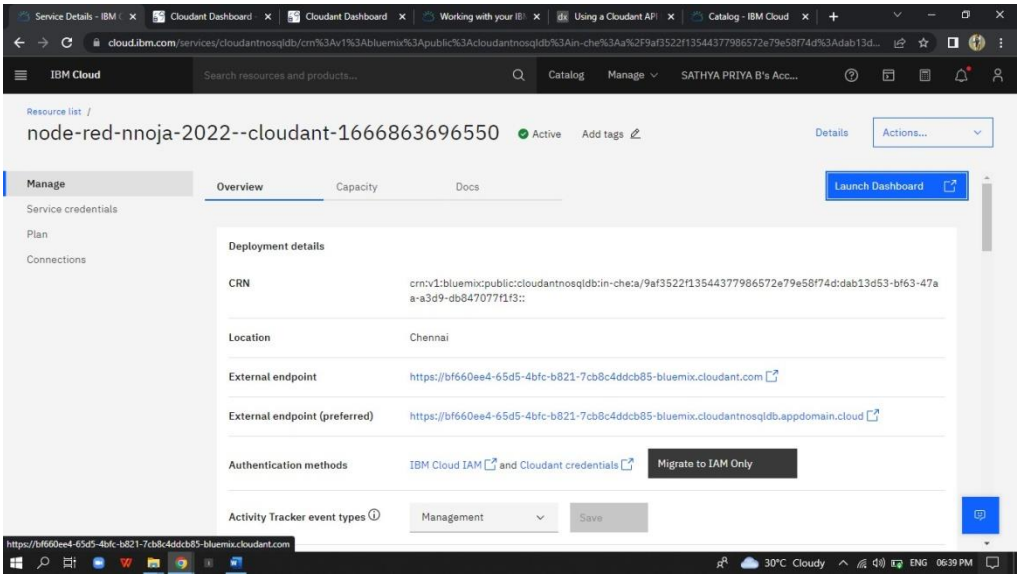
Software


Download Python | Python.org
python.org/downloads/
Python
PSF
Docs
PyPI
Jobs
Community
python
Donate
Search
GO
Socialize
About
Downloads
Documentation
Community
Success Stories
News
Events
Download the latest version for Windows
Download Python 3.11.0
Looking for Python with a different OS? Python for Windows,
Linux/UNIX, macOS, Other
Want to help test development versions of Python? Prereleases,
Docker images
Support Python in 2022! Check out the special deal open only till November 22nd.
Join in today!
Active Python Releases
34°C Mostly cloudy
ENG
03:22 PM


Command Prompt - python.exe -m pip install --upgrade pip
Microsoft Windows [Version 10.0.19044.2130]
(c) Microsoft Corporation. All rights reserved.
C:\Users\ELCOT> pip install wiotp-sdk
Requirement already satisfied: wiotp-sdk in c:\users\elcot\appdata\local\programs\python\python310\lib\site-packages (0.11.0)
Requirement already satisfied: iso8601>=0.1.12 in c:\users\elcot\appdata\local\programs\python\python310\lib\site-packages (from wiotp-sdk) (1.1.0)
Requirement already satisfied: pytz>=2018.9 in c:\users\elcot\appdata\local\programs\python\python310\lib\site-packages (from wiotp-sdk) (2022.5)
Requirement already satisfied: pyyaml>=3.13 in c:\users\elcot\appdata\local\programs\python\python310\lib\site-packages (from wiotp-sdk) (6.0)
Requirement already satisfied: paho-mqtt>=1.5.0 in c:\users\elcot\appdata\local\programs\python\python310\lib\site-packages (from wiotp-sdk) (1.6.1)
Requirement already satisfied: requests>=2.21.0 in c:\users\elcot\appdata\local\programs\python\python310\lib\site-packages (from wiotp-sdk) (2.28.1)
Requirement already satisfied: requests_toolbelt>=0.8.0 in c:\users\elcot\appdata\local\programs\python\python310\lib\site-packages (from wiotp-sdk) (0.10.1)
Requirement already satisfied: charset-normalizer<3,>=2 in c:\users\elcot\appdata\local\programs\python\python310\lib\site-packages (from requests>=2.21.0->wiotp-sdk) (2.1.1)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\elcot\appdata\local\programs\python\python310\lib\site-packages (from requests>=2.21.0->wiotp-sdk) (2022.9.24)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\elcot\appdata\local\programs\python\python310\lib\site-packages (from requests>=2.21.0->wiotp-sdk) (1.26.12)
Requirement already satisfied: idna<4,>=2.5 in c:\users\elcot\appdata\local\programs\python\python310\lib\site-packages (from requests>=2.21.0->wiotp-sdk) (3.4)
[notice] A new release of pip available: 22.3 -> 22.3.1
[notice] To update, run: python.exe -m pip install --upgrade pip
C:\Users\ELCOT>python.exe -m pip install --upgrade pip
Requirement already satisfied: pip in c:\users\elcot\appdata\local\programs\python\python310\lib\site-packages (22.3)
Collecting pip
Downloading pip-22.3.1-py3-none-any.whl (2.1 MB)
2.1/2.1 MB 2.4 MB/s eta 0:00:00
Installing collected packages: pip
Attempting uninstall: pip
Found existing installation: pip 22.3
Uninstalling pip-22.3:
Successfully uninstalled pip-22.3
30°C Cloudy
ENG
03:51 PM

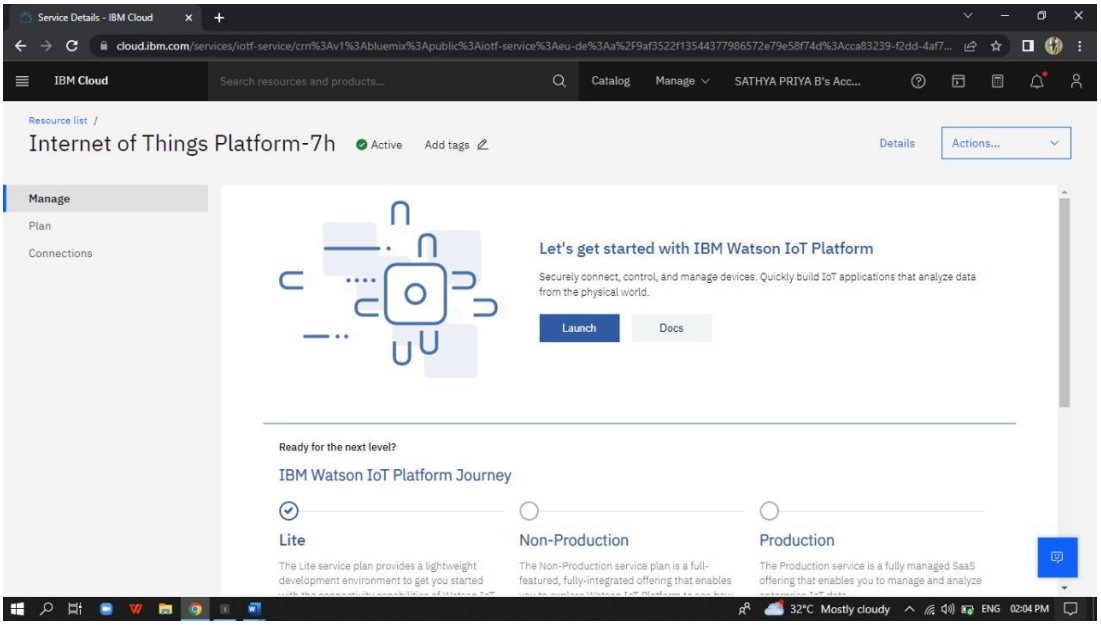


3. PROJECT OBJECTIVES

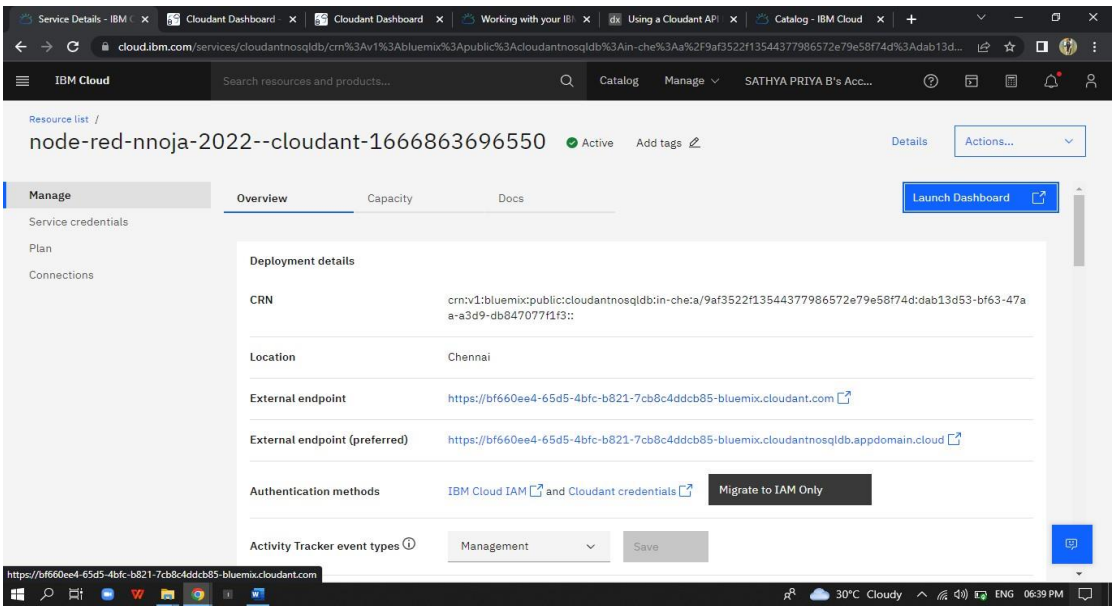


4.7 CRTATE AND CONFIGURE IBM CLOUD SERVICES

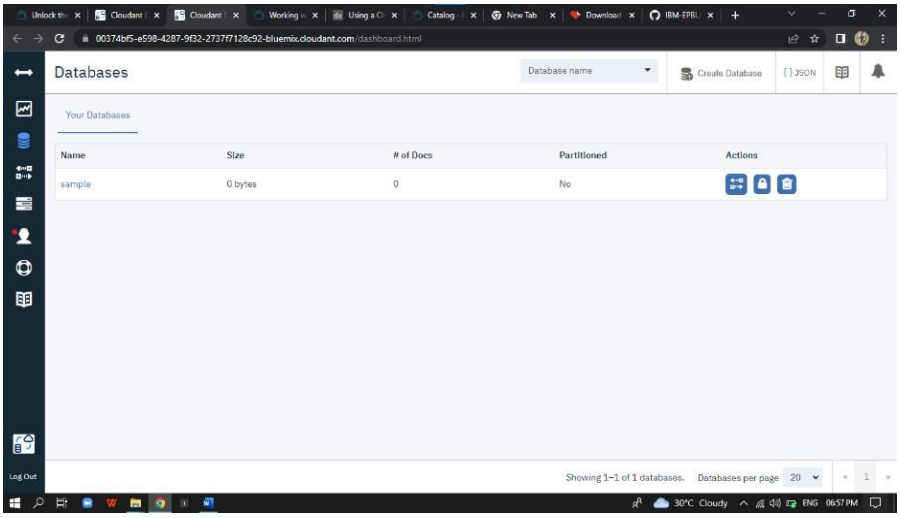
Create IBM Watson IoT Platform And Device

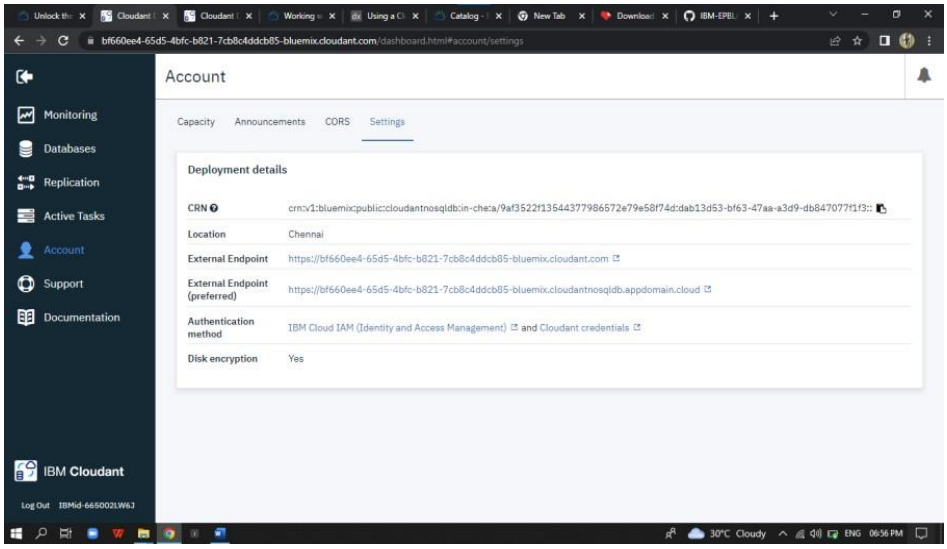


Create Node-RED Service



Create A Datebase In Cloudant DB


00374bf5-e598-4287-9f32-2737f7128c92-bluemix.cloudant.com/dashboard.html
Databases
Database name
Create Database
{ } JSON
Your Databases
Name
Size
# of Docs
Partitioned
Actions
sample
0 bytes
0
No
Log Out
Showing 1–1 of 1 databases.
Databases per page
20
30°C Cloudy
ENG
06:57 PM


bf660ee4-65d5-4bfc-b821-7cb8c4ddcb85-bluemix.cloudant.com/dashboard.html#account/settings
Account
Monitoring
Databases
Replication
Active Tasks
Account
Support
Documentation
Capacity
Announcements
CORS
Settings
Deployment details
CRN
Location
Chennai
External Endpoint
https://bf660ee4-65d5-4bfc-b821-7cb8c4ddcb85-bluemix.cloudant.com
External Endpoint (preferred)
https://bf660ee4-65d5-4bfc-b821-7cb8c4ddcb85-bluemix.cloudantnosqldb.appdomain.cloud
Authentication method
IBM Cloud IAM (Identity and Access Management) and Cloudant credentials
Disk encryption
Yes
IBM Cloudant
Log Out
30°C Cloudy
ENG
06:56 PM

5. DEVELOP THE PYTHON SCRIPT

Develop A Python Script

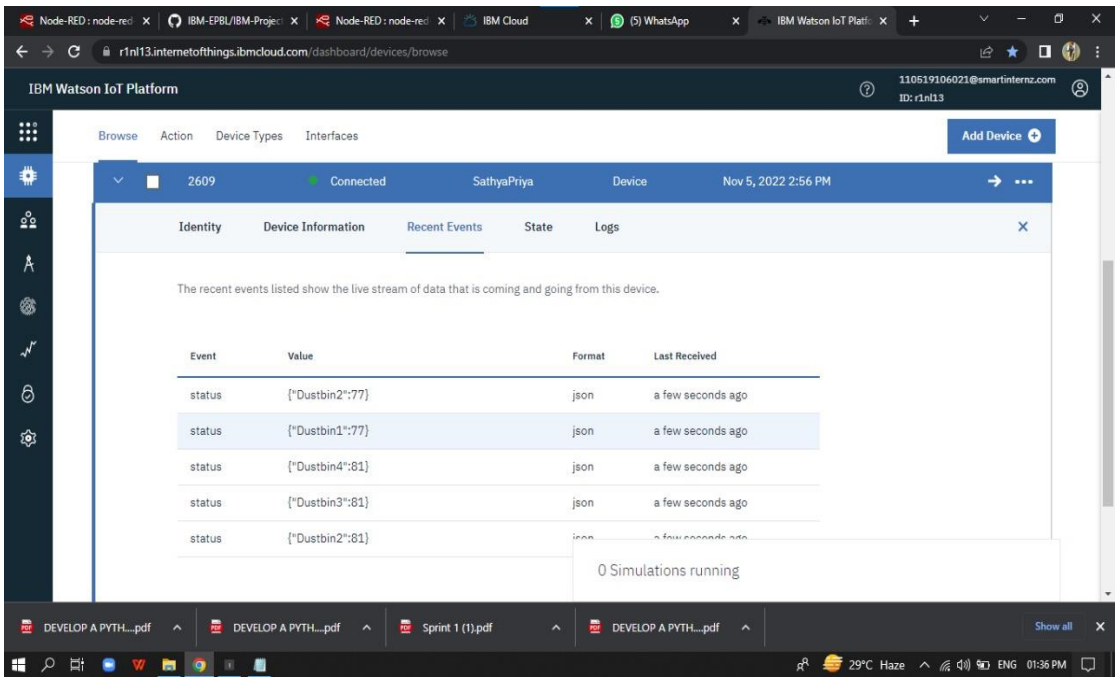
PYTHON CODE:

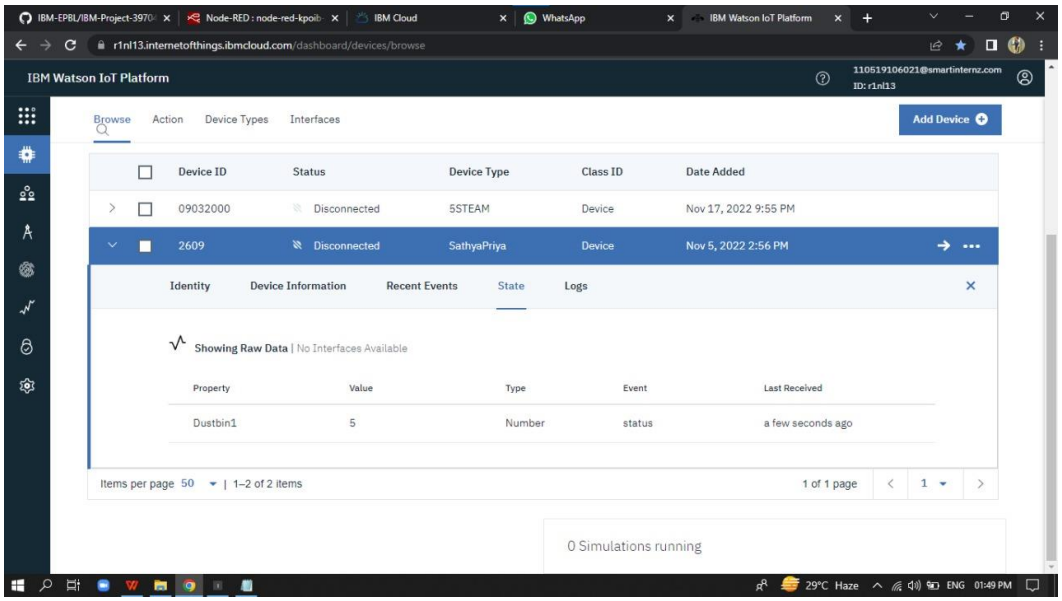
```
#IBM Watson IOT Platform
#pip install wiotp-sdk
import wiotp.sdk.device
import time
import random
myConfig = {
  "identity": {
    "orgId":
    "r1nl13",
    "typeId":
    "SathyaPriya",
    "deviceId":"2609"
  },
  "auth": {
    "token": "i5QqQIPGeqAES5iqgy"
  }
}
def myCommandCallback(cmd):
  print("Message received from IBM IoT Platform: %s" % cmd.data['command']) m=cmd.data['command']


client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
```

```
client.connect()


while True:
  Dust =
  random.randint(0,100) if
  (Dust > 90):
    print("Dustbin Over
  Flow")
  myData={'Dustbin1':Dust}
  client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
  onPublish=None) print(myData)
  client.commandCallback = myCommandCallback
  time.sleep(2)
##client.disconnect()
  if(Dust>80):
    print("")
  myData={'Dustbin2':Dust}
  client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
  onPublish=None) print(myData)
  client.commandCallback = myCommandCallback
  time.sleep(2)
##client.disconnect()
  if(Dust>78):
    print("")
  myData={'Dustbin3':Dust}
  client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
  onPublish=None) print(myData)
```

```
  client.commandCallback = myCommandCallback time.sleep(2)
##client.disconnect() if(Dust>70):
    print("")
  myData={'Dustbin4':Dust}
  client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
  print(myData)
  client.commandCallback = myCommandCallback time.sleep(2)
client.disconnect()
```

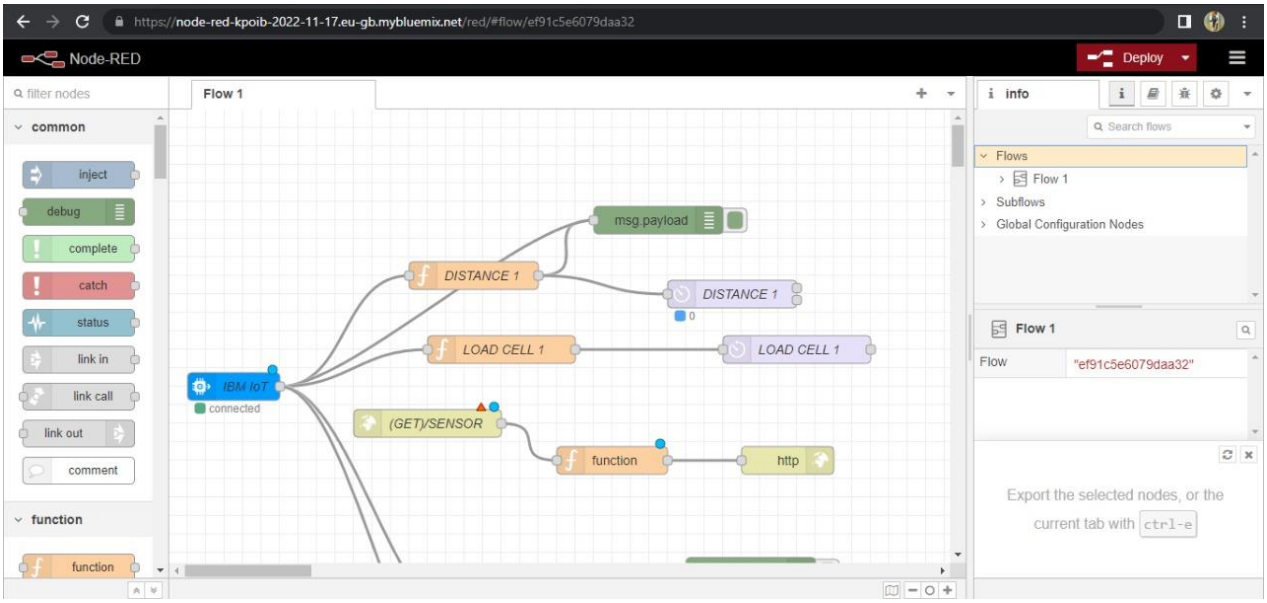
Publish Data To The IBM Cloud

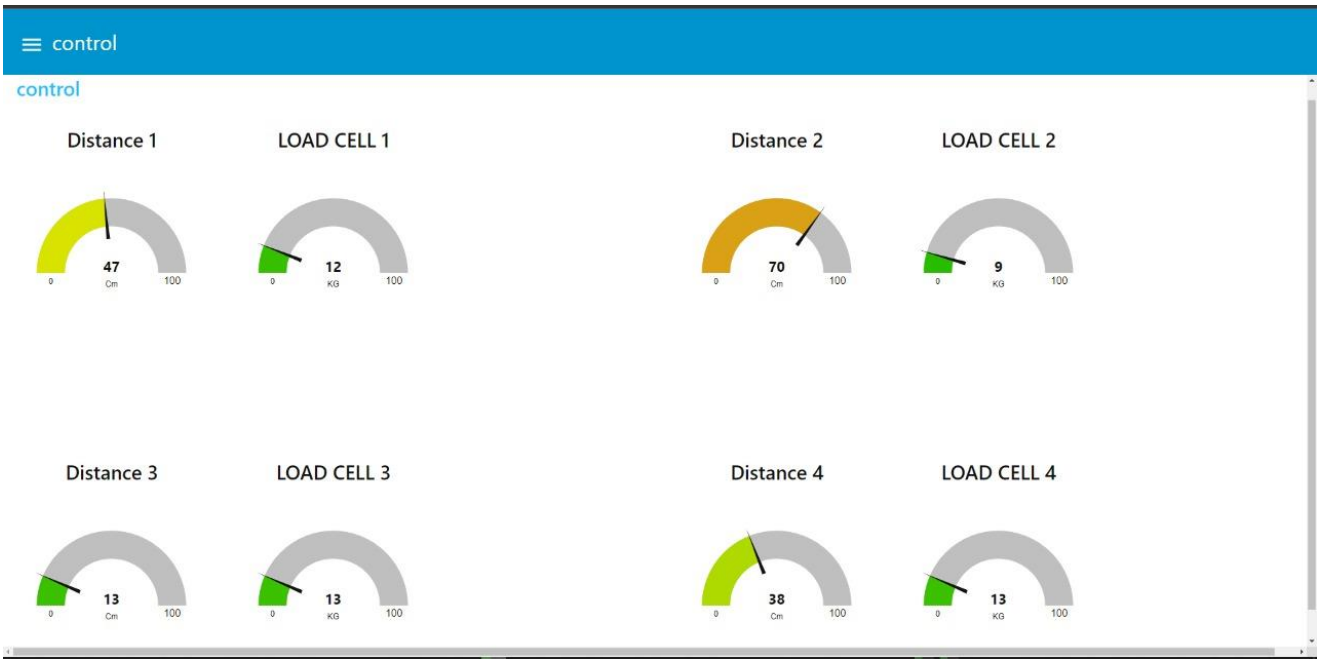




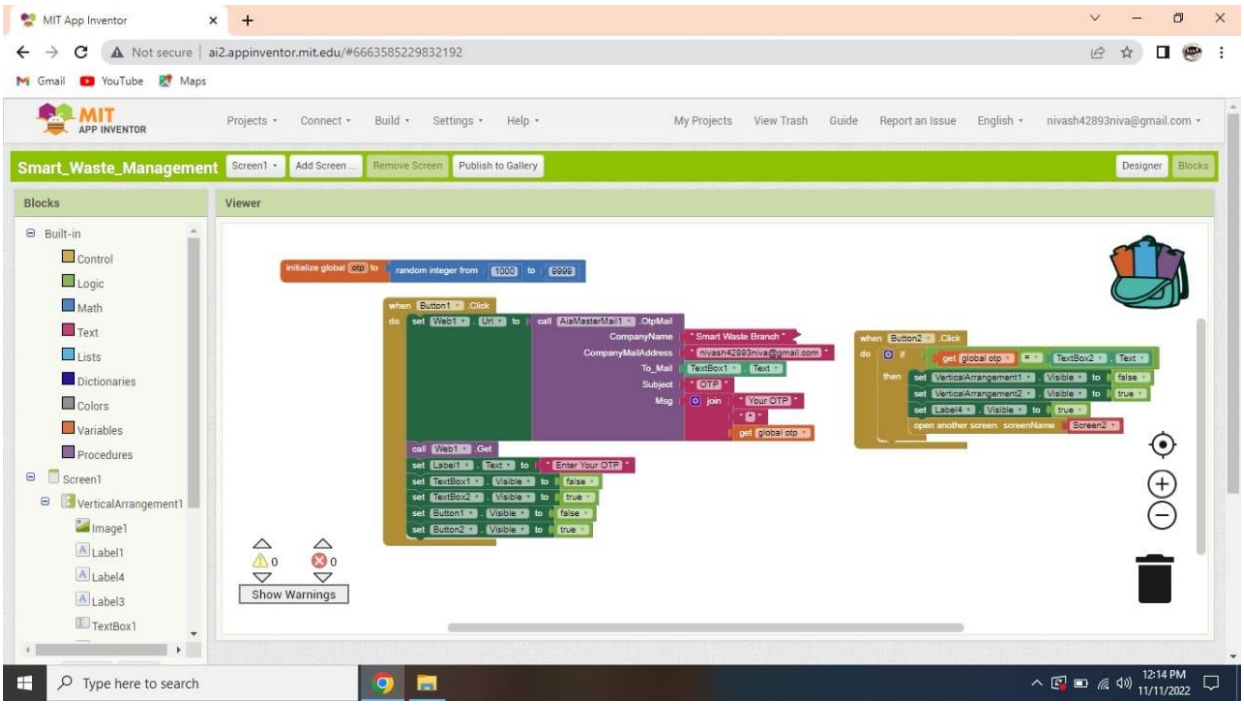
6. DEVELOP A WEB APPLICATION USING NODE-RED SERVICE

Develop The Web Application Using Node-RED





Use Dashboard Node For Creating UI(Web APP)



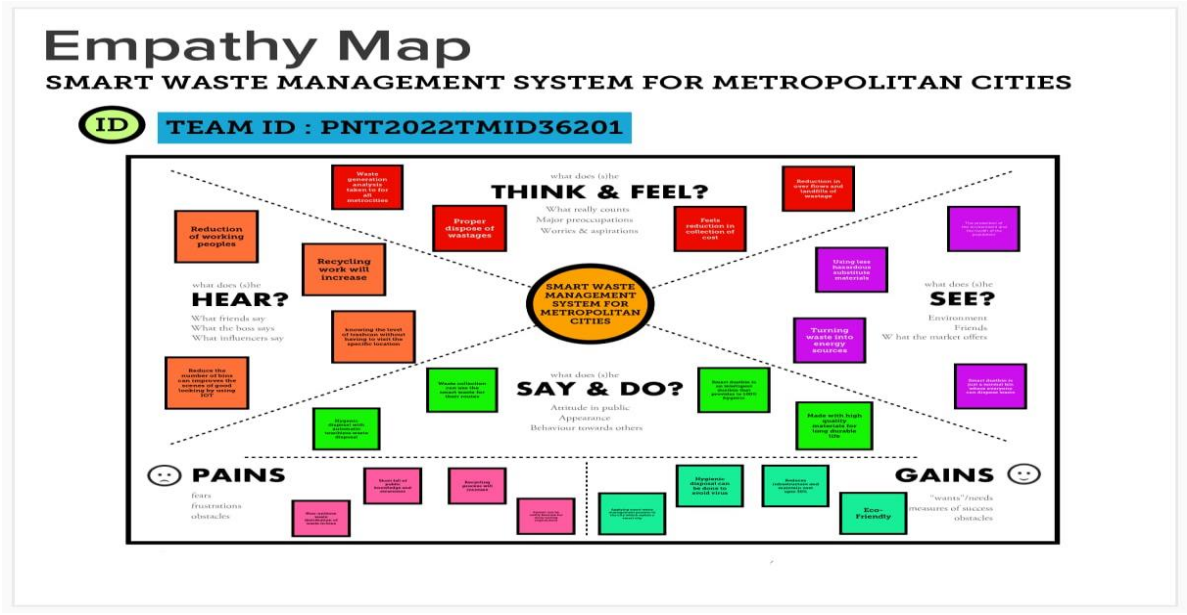
7. IDEATION PHASE

Literature Survey On The Selected Project & Information Gathering

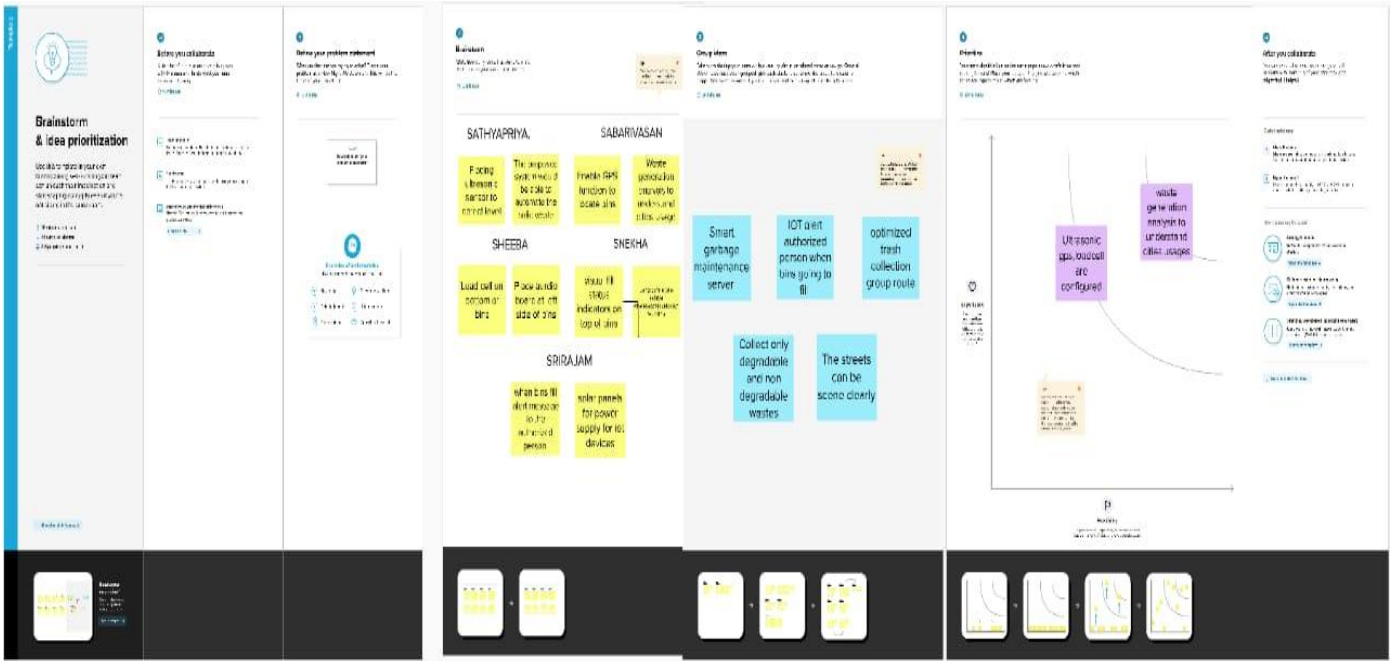
PAPER TITLE	AUTHOR	OUTCOME
Actuator Networks Sensor and IoT-Based Solid Waste Management Solutions: A Survey	1) Kellow Pardini 2) Joel Rodrigues	The scientists also suggested the technique for management and disposal of garbage, in which the garbage bin had been made to interface and connect with a system which was based on microcontroller and also had IR systems along with a main system that clearly displayed the current weight and level of the waste in the bin.
Waste Management by a Robot-A Smart and Autonomous Technique	1)Shikha Parashar 2)Pankaj Tomar	Smart waste management or more fully, smart municipal waste collection and management refers to the process of municipal waste collection, disposal, recycling and landfill .

Automatic Waste Segregator as an integral part of Smart Bin for waste management system in a Smart City	1)Chander Partap Singh 2)Manisha	For this reason, IoT infrastructures enable to manage the waste collection efficiently. Recent studies utilize the IoT devices placed in the garbage which are connected to the server through Low-Power Wide Area Networks (LPWANs) like SigFox, LoRa, and NB-IoT
IoT-Enabled Intelligent Solid Waste Management System for Smart City: A Survey	Swati Dewangan	Smart waste bins have using sensors to monitor the empty space in bins. Then, with the aim of Internet of Things, they are employed to efficiently organize collection routes .
Waste management in urban environments: insights of the citizens' views in a densely populated municipality in Greece	1)Paraskevi Karanikola 2)Stilianos Tampakis	In a literature review for smart waste management and a comparison of the different methodologies is given. The authors focus on the IoT, considering its elements (identification, sensing, communication, computation, semantics, and services) and how the IoT can be used effectively to manage solid waste.

Prepare Empathy Map



Ideation



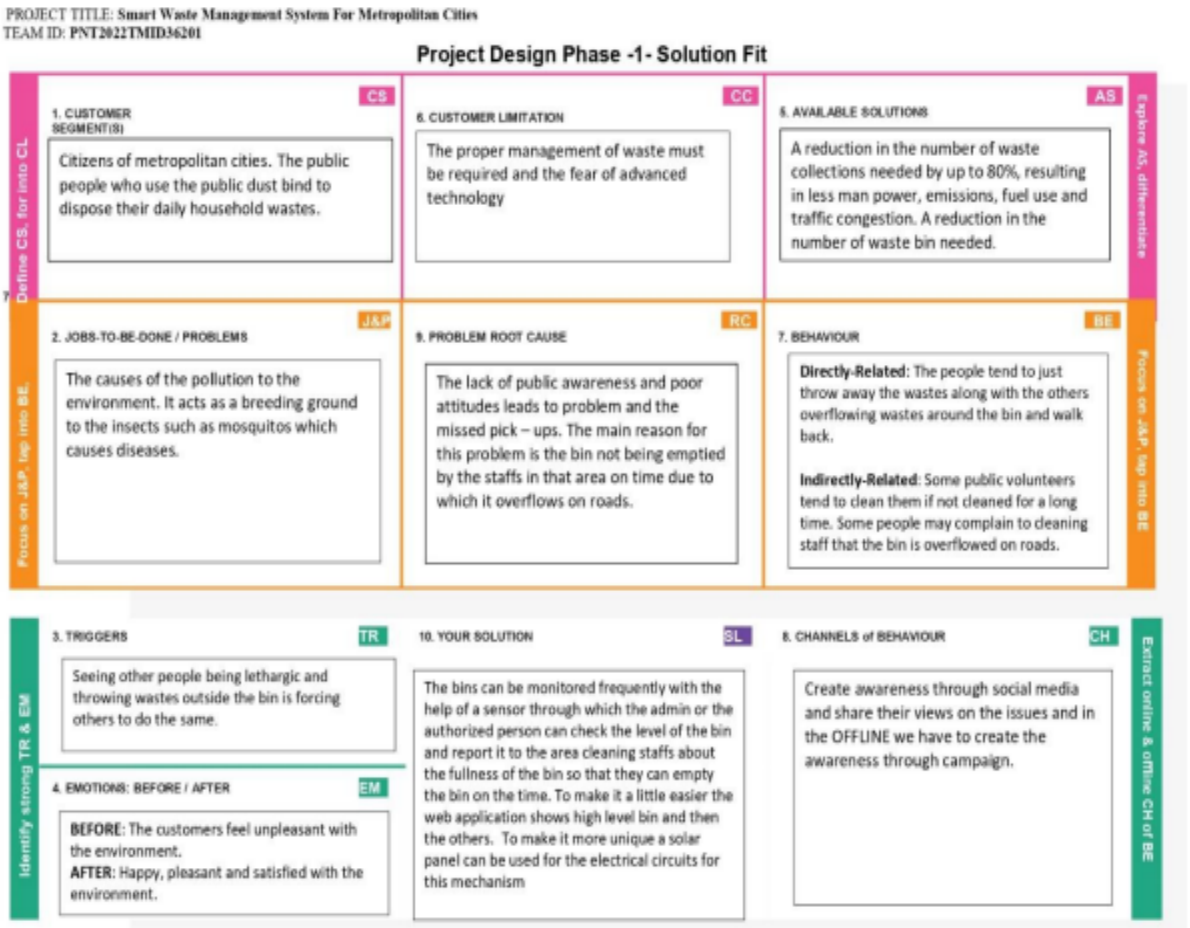
8. PROJECT DESIGN PHASE-I
Proposed Solution

S. No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	This project deals with the problem of waste management in smart cities, where the garbage collection system is not optimized. This project enables the organizations to meet their needs of smart garbage management systems. This system allows the authorized person to know the fill level of each garbage bin in a locality or city at all times, to give a cost-effective and time-saving route to the truck drivers.

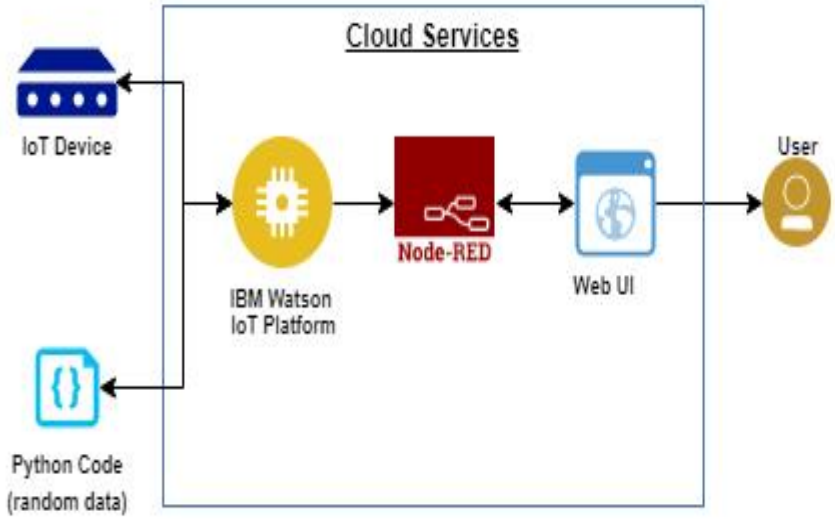
2.	Idea / Solution description	The key research objectives are as follows: • The proposed system would be able to automate the solid waste monitoring process and management of the overall collection process using IOT (Internet of Things). • The Proposed system consists of main subsystems namely Smart Trash System (STS) and Smart Monitoring and Controlling Hut (SMCH). • In the proposed system, whenever the waste bin gets filled this is acknowledged by placing the circuit at the waste bin, which transmits it to the receiver at the desired place in the area or spot. • In the proposed system, the received signal indicates the waste bin status at the monitoring and controlling system.
3.	Novelty / Uniqueness	We are going to establish SWM in our college but the real hard thing is that janitor (cleaner)do not know to operate these thing practically so here our team planned to build a wrist band to them, that indicate via light blinking when the dustbin fill and this is Uniqueness we made here beside from project constrain.
4.	Social Impact / Customer Satisfaction	From the public perception as worst impacts of present solid waste disposal practices are seen direct social impacts such as neighborhood of landfills to communities, breeding of pests and loss in property values
5.	Business Model (Revenue Model)	Waste Management organize its operations into two reportable business segments: Solid Waste, comprising the Company's waste collection, transfer, recycling and resource recovery, and disposal services, which are operated and managed locally by the Company's various subsidiaries, which focus on distinct geographic areas; and Corporate and Other, comprising the Company's other activities, including its development and operation of landfill gas-to-energy facilities in the INDIA, and its recycling brokerage services, as well as various corporate functions.

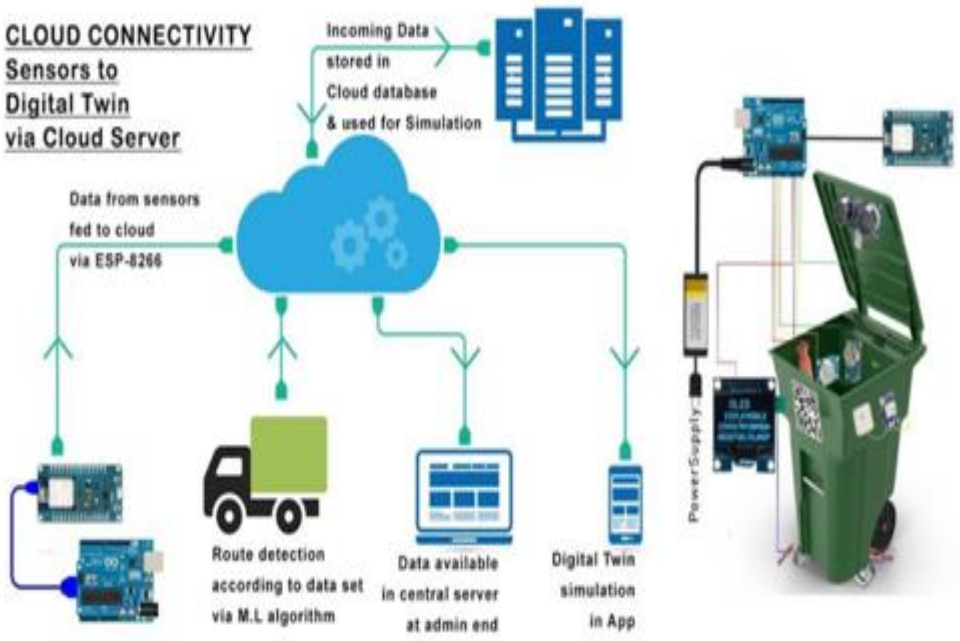
6.	Scalability of the Solution	In this regard, smart city design has been increasingly studied and discussed around the world to solve this problem. Following this approach, this paper presented an efficient IoT based and real-time waste management model for improving the living environment in cities, focused on a citizen perspective. The proposed system uses sensor and communication technologies where waste data is collected from the smart bin, in real-time, and then transmitted to an online platform where citizens can access and check the availability of the compartments scattered around a city.

Problem Solution Fit

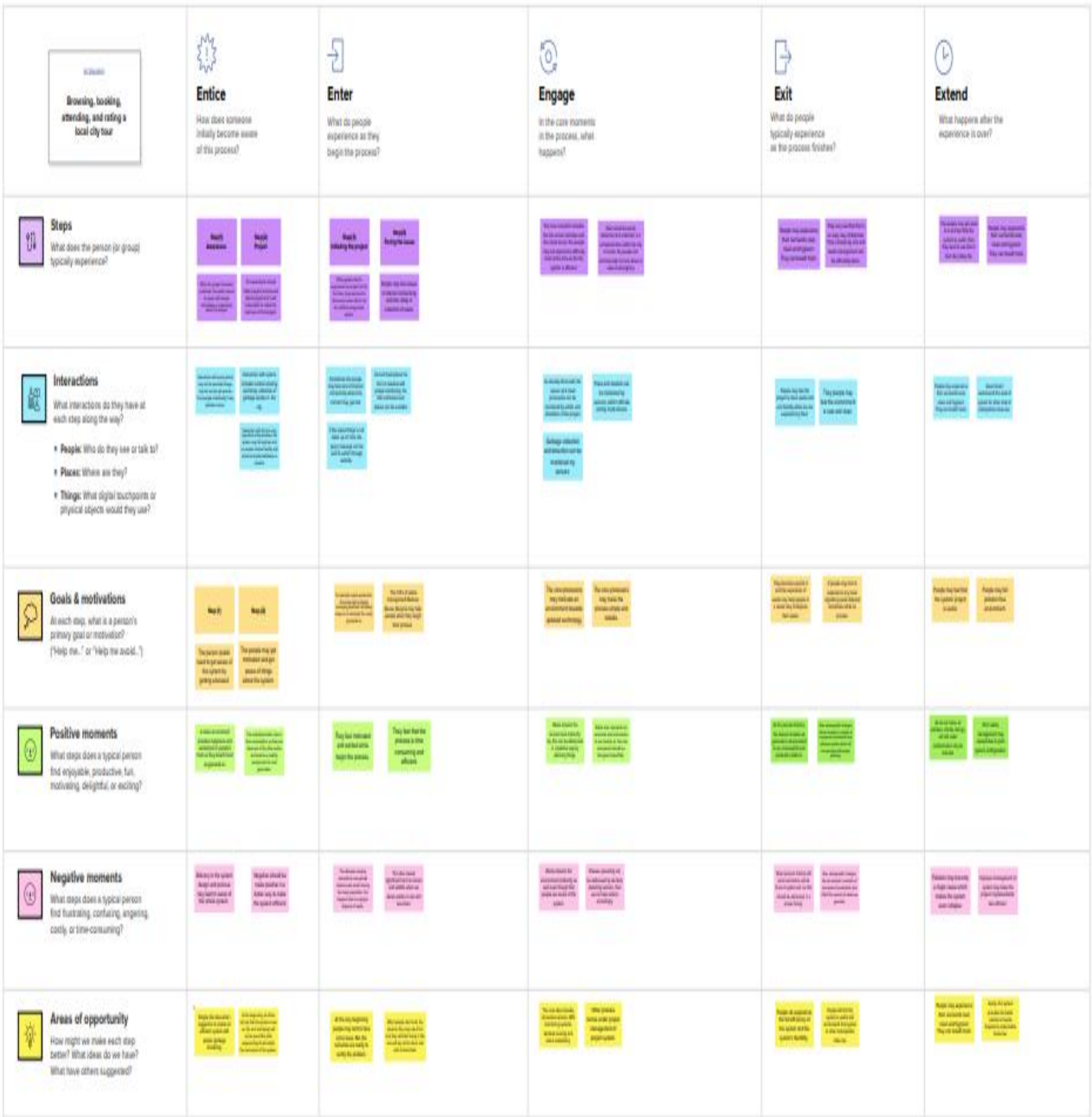


Solution Architecture





9. PROJECT DESIGN PHASE-II
Customer Journey



Functional Requirement

Functional Requirements:

Following are the functional requirements of the proposed solution.

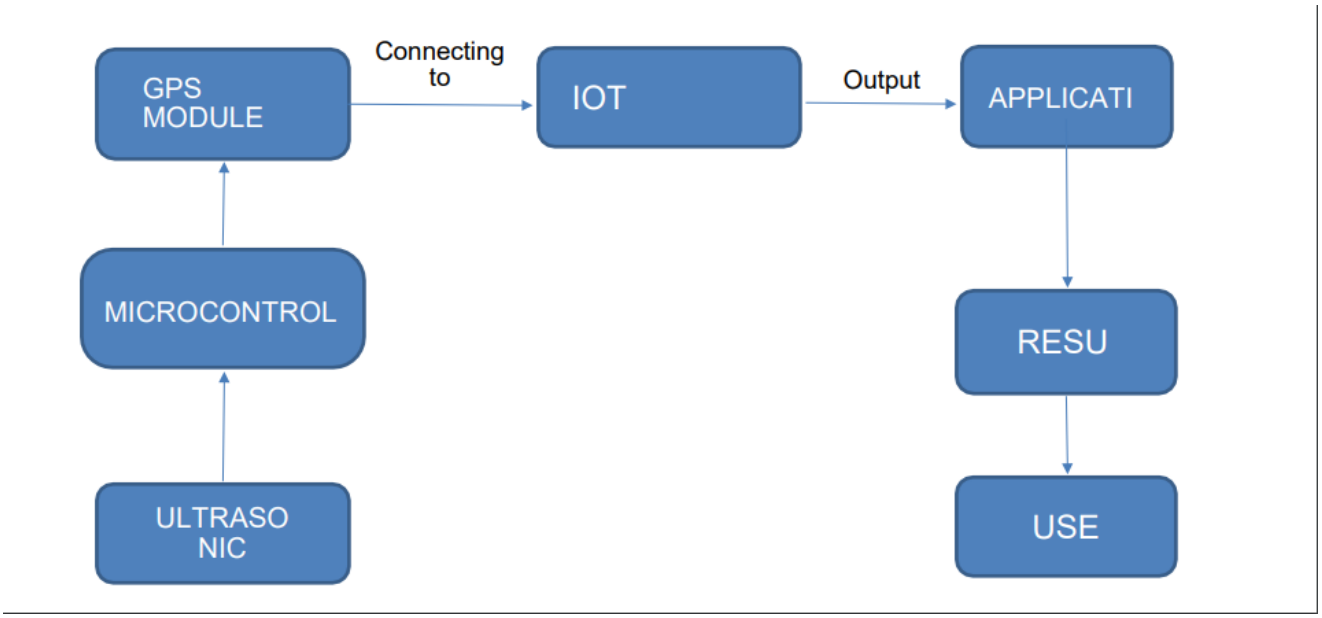
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Detailed bin inventory	All monitored bins and stands can be seen on the map, and you can visit them at any time via the Street View feature from Google. Bins or stands are visible on the map as green, orange or red circles. You can see bin details in the Dashboard – capacity, waste type, last measurement, GPS location and collection schedule or pick recognition
FR-4	Real time bin monitoring	The Dashboard displays real-time data on fill-levels of bins monitored by smart sensors. In addition to the % of fill-level, based on the historical data, the tool predicts when the bin will become full, one of the functionalities that are not included even in the best waste management software.. Sensors recognize picks as well; so you can check when the bin was last collected. With real-time data and predictions, you can eliminate the overflowing bins and stop collecting half-empty ones
FR-5	Eliminate inefficient picks	Eliminate the collection of half-empty bins. The sensors recognize picks. By using real-time data on fill-levels and pick recognition, we can show you how full the bins you collect are. The report shows

Non-functional Requirements:

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

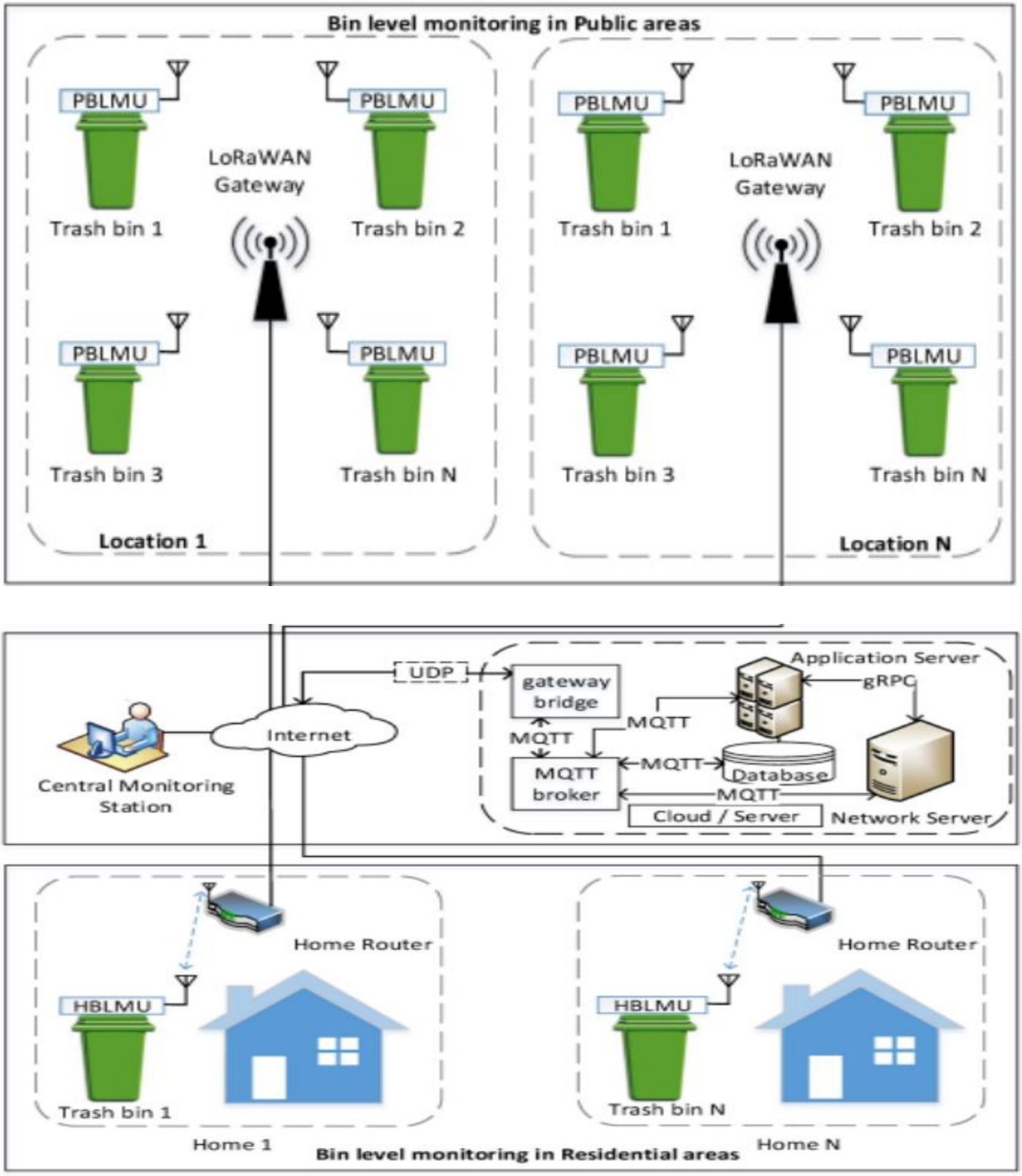
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	IoT device verifies that usability is a special and important perspective to analyze user requirements, which can further improve the design quality. In the design process with user experience as the core, the analysis of users' product usability can indeed help designers better understand users' potential needs in waste management, behaviour and experience
NFR-2	**Security**	Use a reusable bottles Use reusable grocery bags Purchase wisely and recycle Avoid single use food and drink containers
NFR-3	**Reliability**	Smart waste management is also about creating better working conditions for waste collectors and drivers. Instead of driving the same collection routes and servicing empty bins, waste collectors will spend their time more efficiently, taking care of bins that need servicing.
NFR-4	**Performance**	The Smart Sensors use ultrasound technology to measure the fill levels (along with other data) in bins several times a day. Using a variety of IoT networks ((NB-Io T, GPRS), the sensors send the data to Smart Waste Management

Date Flow Diagrams



User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Web user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can able to access the status the status every bin in the city	High	Sprint-1
Customer (Cloud user)	Access	USN-2	As a user, I can access database of the garbage bin	I can receive confirmation email & cl ick confirm	High	Sprint-2
Customer Care Executive	Gmail account	USN 3	As a user, I can register for the application through Gmail	I can register and access the model	Medium	Sprint-1
Administrator	Login	USN 4	As a Admin, I can log into the application by entering email & password	I can access the garbage database directly	High	Sprint-1
Customer (User)	Internet Facility	USN 5	As a user I can give input to the model through the website	I can get location and status of the bin	High	Sprint-2
Customer (User)	Laptop or Computer or Mobile	USN 6	As a user I can view the pictorial garbage status and able to view the location of the bin in a maps	I can insights on garbage status	High	Sprint-2

Technology Architecture



10.PROJECT PLANNING PHASE

Prepare Milestone & Activity List

TITLE	DESCRIPTION	DATE
Literature Survey & Information Gathering	Literature survey on the selected project & gathering information by referring the, technical papers, research publications etc.	30 SEPTEMBER 2022
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem statements	26 SEPTEMBER 2022
Ideation	List the by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.	28 SEPTEMBER 2022
Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.	05 OCTOBER 2022
Problem Solution Fit	Prepare problem - solution fit document.	09 OCTOBER 2022
Solution Architecture	Prepare solution architecture document.	10 OCTOBER 2022

Customer Journey	Prepare the customer journey maps to understand the user interactions & experiences with the application (entry to exit).	20 OCTOBER 2022
Functional Requirement	Prepare the functional requirement document.	15 OCTOBER 2022
Data Flow Diagrams	Draw the data flow diagrams and submit for review.	19 OCTOBER 2022
Technology Architecture	Prepare the technology architecture diagram.	20 OCTOBER 2022
Prepare Milestone & Activity List	Prepare the milestones & activity list of the project.	09 NOVEMBER 2022
Project Development - Delivery of Sprint-1, 2, 3 & 4	Develop & submit the developed code by testing it.	IN PROGRESS.

Sprint Delivery Plan

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Login	USN-1	As an Administrator, I need to give user id and password for ever workers over there in metropolis to work further	10	High	Sathya Priya
Sprint-1	Login	USN-2	As a Co-Admin, I will control the waste level by monitoring them via real time web portal. Once the wastage is filled , I will notify to the trash truck with location of bin with bin ID or number	10	High	Sabarivasan
Sprint-2	Dashboard	USN-3	As a Truck Driver, I will follow Co-Admin's Instruction to reach the filled bin in short roots and save time	20	Low	Sheeba
Sprint-3	Dashboard	USN-4	As a Local Garbage Collector, I will gather all the waste from the garbage, load it onto a garbage truck, and deliver it to Landfills	20	Medium	Snekha
Sprint-4	Dashboard	USN-5	As a Metropolis officer, I will make sure everything is proceeding as planned and without any problems	20	High	Sri Rajam

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	30 Oct 2022	20	01 Nov 2022
Sprint-2	20	6 Days	31 Oct 2022	07 Nov 2022	20	08 Nov 2022
Sprint-3	20	6 Days	09 Nov 2022	11 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	18 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)

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

11.PROJECT DEVELOPMENT PHASE

Project Development-Delivery Of Sprint-1

```
#IBM Watson IOT Platform
#pip install wiotp-sdk
import wiotp.sdk.device
import time
import random
myConfig = {
  "identity": {
    "orgId": "r1nl13",
    "typeId": "SathyaPriya",
    "deviceId":"2609"
  },
  "auth": {
    "token": "i5QqQIPGeqAES5iqgy"
  }
}

def myCommandCallback(cmd):
```

```
  print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
  m=cmd.data['command']

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

while True:
  Dust = random.randint(0,100)
  if (Dust > 90):
    print("Dustbin Over Flow")
  myData={'Dustbin1':Dust}
  client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
onPublish=None)
  print(myData)
  client.commandCallback = myCommandCallback
  time.sleep(2)
##client.disconnect()
  if(Dust>80):
    print("")
  myData={'Dustbin2':Dust}
  client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
onPublish=None)
```

```
    print(myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)
##client.disconnect()
    if(Dust>78):
        print("")
    myData={'Dustbin3':Dust}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
onPublish=None)
    print(myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)
##client.disconnect()
    if(Dust>70):
        print("")
    myData={'Dustbin4':Dust}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
onPublish=None)
    print(myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)
client.disconnect()
```

PYTHON SCRIPT:

File Edit Format Run Options Window Help

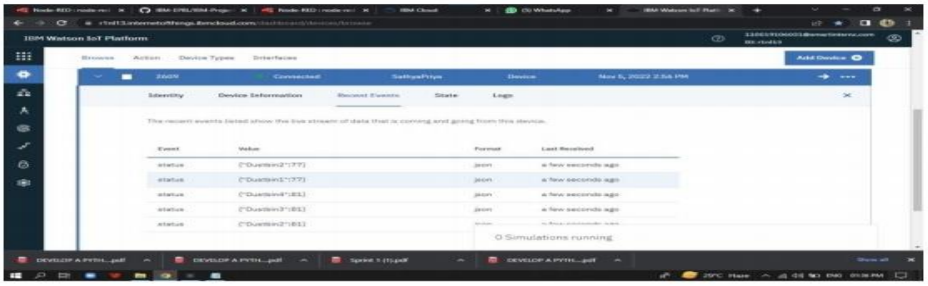
```
#IBM Watson IOT Platform
#pip install wiotp-sdk
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity": {
        "orgId": "r1n113",
        "typeId": "SathyaPriya",
        "deviceId":"2609"
    },
    "auth": {
        "token": "i5QqQIPGeqAE55iqgy"
    }
}

def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
    m=cmd.data['command']

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

while True:
    Dust = random.randint(0,100)
    if (Dust > 90):
        print("Dustbin Over Flow")
    myData={'Dustbin1':Dust}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
    print(myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)
##client.disconnect()
    if(Dust>80):
        print("")
    myData={'Dustbin2':Dust}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
    print(myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)
```

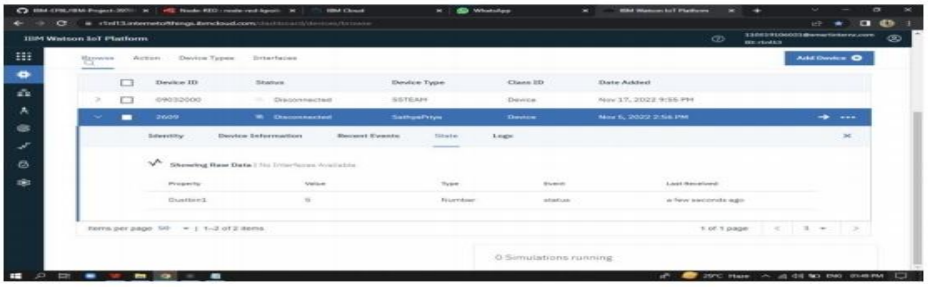
```
##client.disconnect()
    if(Dust>78):
        print("")
    myData={'Dustbin3':Dust}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
    print(myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)
##client.disconnect()
    if(Dust>70):
        print("")
    myData={'Dustbin4':Dust}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
    print(myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)
client.disconnect()
```

OUTPUT:

```
"Python 3.7.4 Shell"
File  Edit  Shell  Debug  Options  Window  Help
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul  8 2019, 20:34:20) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:\Users\SRI RAJAN\Documents\Demo.py ==========
2022-11-19 12:32:42,119    wiotp.sdk.device.client.DeviceClient   INFO     Connected successfully: d:rinli3:SathyaPriya:2609
{'Dustbin1': 41}
2022-11-19 12:32:44,025    wiotp.sdk.device.client.DeviceClient   ERROR    Unexpected disconnect from IBM Watson IoT Platform: 7
2022-11-19 12:32:45,947    wiotp.sdk.device.client.DeviceClient   INFO     Connected successfully: d:rinli3:SathyaPriya:2609
{'Dustbin2': 41}
{'Dustbin3': 41}
{'Dustbin4': 41}
{'Dustbin1': 80}
{'Dustbin2': 80}
2022-11-19 12:32:54,634    wiotp.sdk.device.client.DeviceClient   ERROR    Unexpected disconnect from IBM Watson IoT Platform: 7

2022-11-19 12:32:56,743    wiotp.sdk.device.client.DeviceClient   INFO     Connected successfully: d:rinli3:SathyaPriya:2609
{'Dustbin3': 80}
2022-11-19 12:32:58,651    wiotp.sdk.device.client.DeviceClient   ERROR    Unexpected disconnect from IBM Watson IoT Platform: 7

2022-11-19 12:33:00,540    wiotp.sdk.device.client.DeviceClient   INFO     Connected successfully: d:rinli3:SathyaPriya:2609
{'Dustbin4': 80}
{'Dustbin1': 51}
{'Dustbin2': 51}
{'Dustbin3': 51}
{'Dustbin4': 51}
2022-11-19 12:33:09,930    wiotp.sdk.device.client.DeviceClient   ERROR    Unexpected disconnect from IBM Watson IoT Platform: 7
2022-11-19 12:33:12,599    wiotp.sdk.device.client.DeviceClient   INFO     Connected successfully: d:rinli3:SathyaPriya:2609
{'Dustbin1': 69}
{'Dustbin2': 69}
{'Dustbin3': 69}
{'Dustbin4': 69}
2022-11-19 12:33:20,614    wiotp.sdk.device.client.DeviceClient   ERROR    Unexpected disconnect from IBM Watson IoT Platform: 7
2022-11-19 12:33:27,756    wiotp.sdk.device.client.DeviceClient   INFO     Connected successfully: d:rinli3:SathyaPriya:2609
{'Dustbin1': 44}
{'Dustbin2': 44}
2022-11-19 12:33:30,520    wiotp.sdk.device.client.DeviceClient   ERROR    Unexpected disconnect from IBM Watson IoT Platform: 7
2022-11-19 12:33:32,552    wiotp.sdk.device.client.DeviceClient   INFO     Connected successfully: d:rinli3:SathyaPriya:2609
{'Dustbin3': 44}
```

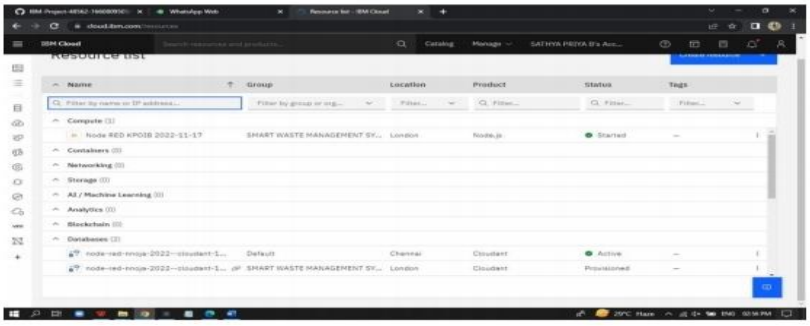
Data in IBM CLOUD Platform:



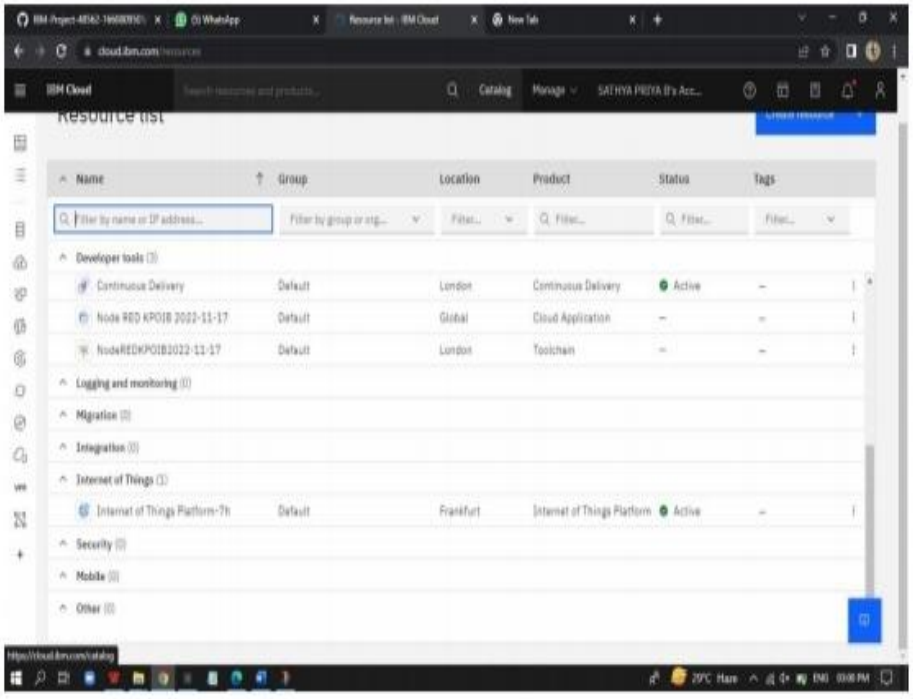


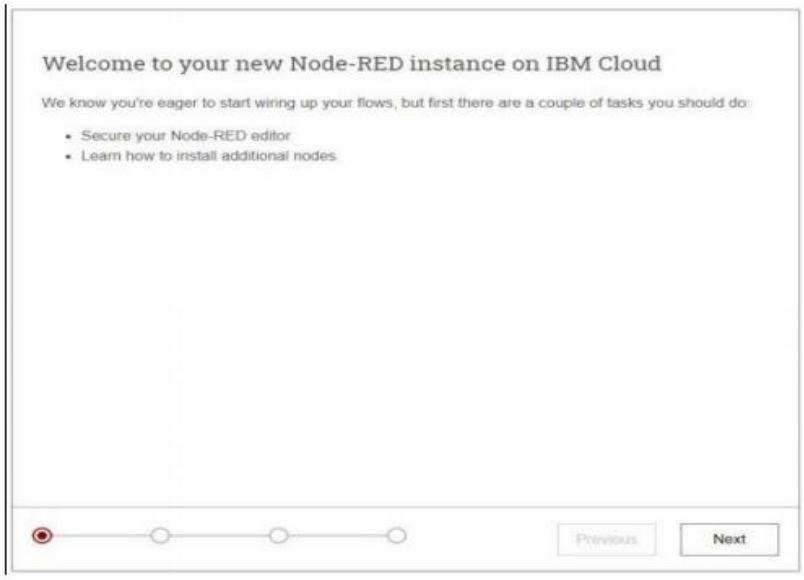
Project Development-Delivery Of Sprint-2

CREATION OF NODE RED:

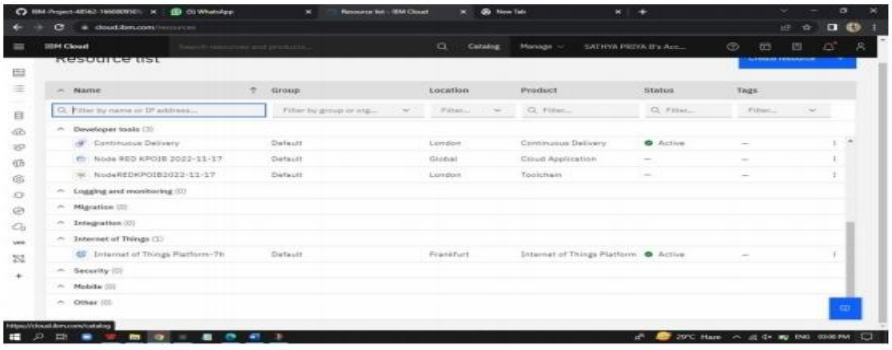


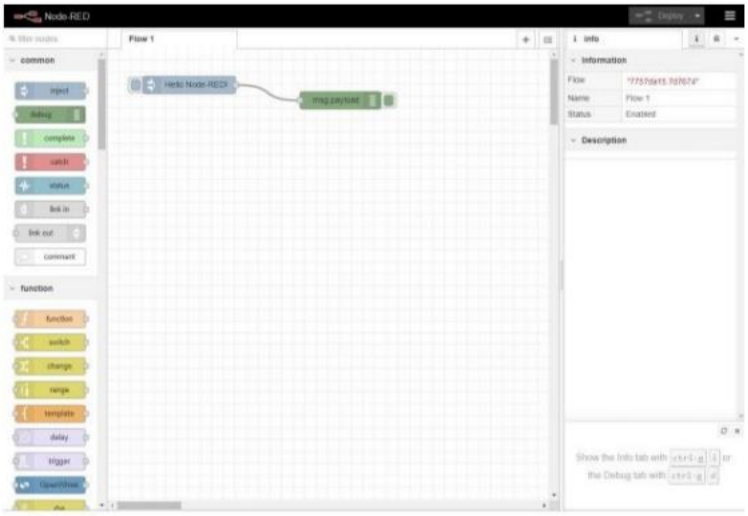
Create the resources list in IBM CLOUD:

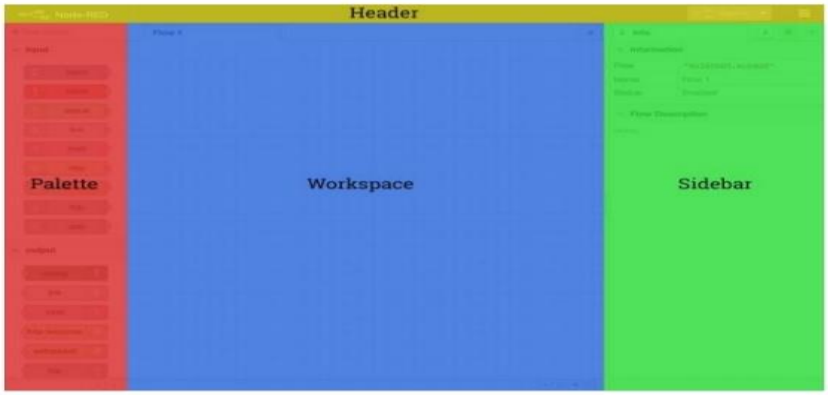




Create the resources list in IBM CLOUD:







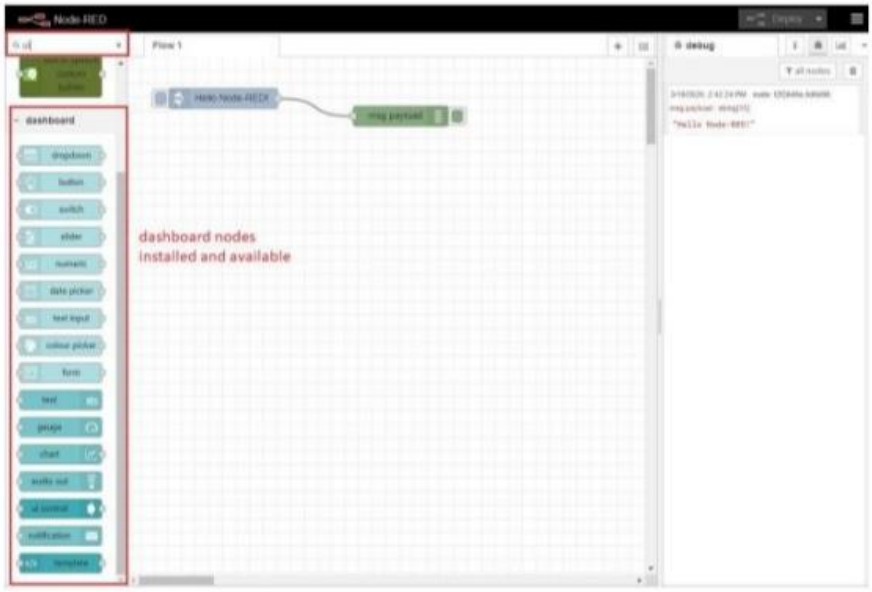
● From the palette, one can drag the nodes and drop them on the editor.

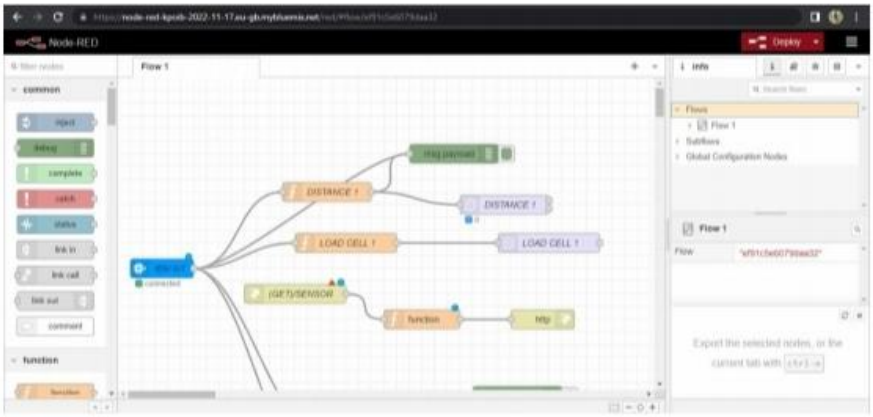
● The sidebar will be useful to know the info about nodes by selecting them and debug console helps us to debug the errors or the output is display on the console.

● The default node-red after launching the node-red application for a new user will contain two nodes connected to each other.

● The first node is the input used to inject the input for output.

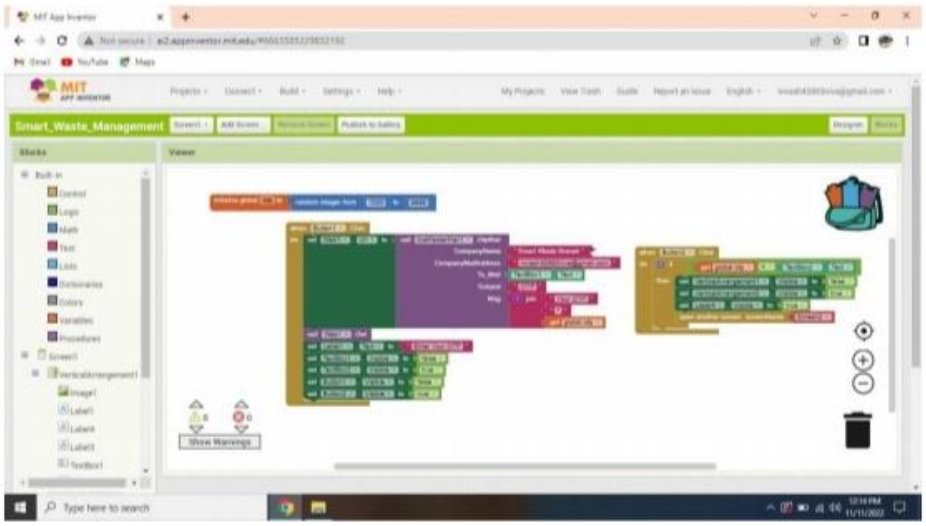
● The debug node to display the output


Node-RED
Flow 1
debug
dashboard
dashboard nodes
installed and available


Node-RED
Deploy
Flow 1
info
common
function
DISTANCE 1
DISTANCE 1
LOAD CELL 1
LOAD CELL 1
GETSENSOR
function
http


control
control
Distance 1
LOAD CELL 1
Distance 2
LOAD CELL 2
47
12
70
9
Distance 3
LOAD CELL 3
Distance 4
LOAD CELL 4
13
13
38
13

MITAPP:



Project Development-Delivery Of Sprint-4

SENDING DATA FROM RASPERRYPI TO IBM CLOUD:

```
#IBM Watson IOT Platform
#pip install wiotp-sdk
import wiotp.sdk.device
import time
import random
myConfig = {
   "identity": {
      "orgId": "r1nl13",
      "typeId": "SathyaPriya",
      "deviceId":"2609"
   },
   "auth": {
      "token": "i5QqQIPGeqAES5iqgy"
   }
}
```

```
def myCommandCallback(cmd):
   print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
   m=cmd.data['command']

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

while True:
   Dust = random.randint(0,100)
   if (Dust > 90):
      print("Dustbin Over Flow")
   myData={'Dustbin1':Dust}
   client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
onPublish=None)
   print(myData)
   client.commandCallback = myCommandCallback
   time.sleep(2)
##client.disconnect()
   if(Dust>80):
      print("")
   myData={'Dustbin2':Dust}
   client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
onPublish=None)
   print(myData)
```

```
    client.commandCallback = myCommandCallback
    time.sleep(2)
##client.disconnect()
    if(Dust>78):
      print("")
    myData={'Dustbin3':Dust}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
onPublish=None)
    print(myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)
##client.disconnect()
    if(Dust>70):
      print("")
    myData={'Dustbin4':Dust}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
onPublish=None)
    print(myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)
client.disconnect()
```

PYTHON CODE:

Demo.py - C:\Users\SRI RAJAM\Desktop\Demo.py (3.7.4)

File Edit Format Run Options Window Help

```
#IBM Watson IOT Platform
#pip install wiotp-sdk
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity": {
        "orgId": "rinl13",
        "typeId": "SathyaPriya",
        "deviceId":"2609"
    },
    "auth": {
        "token": "i5QqQIPGeqAE55iqgy"
    }
}

def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
    m=cmd.data['command']

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

while True:
    temp=random.randint(-20,125)
    hum=random.randint(0,100)
    myData={'temperature':temp, 'humidity':hum}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
    print("Published data Successfully: %s", myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)
client.disconnect()
```

OUTPUT:

12.ADVANTAGES AND DISADVANTAGES

ADVANTAGES

- Reduction in Collection Cost
- No Missed Pickups
- Reduced Overflows
- Waste Generation Analysis
- CO2 Emission Reduction

DISADVANTAGES

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

• This results into high initial cost due to expensive smart dustbins compare to other methods. Sensor nodes used in the dustbins have limited memory size.

13. CONCLUSION :

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

14 . GITHUB LINK

LINK : https://github.com/IBM-EPBL/IBM-Project-39704-1660490335

VIDEO DEMO LINK: https://youtu.be/ikiuOeAsQLM