SMART WASTE MANAGEMENT SYSTEM FOR METEROPOLITIAN CITIES

1.INTRODUCTION:

OVERVIEW:

Smart waste management is the process of collecting, transporting, and disposing the waste material in an efficient manner with minimum impact on the environment. The use of smart bins or smart trashes for collection of waste material and garbage monitoring system helps waste management authorities to manage waste material effectively.

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 system. This system allows the user to know the fill level of each garbage bin in a locality or city at all time, to give a cost effective and time saving route to the truck drivers.

PURPOSE:

This project IoT Garbage Monitoring system is a very innovative system which will help to keep the cities clean. This system monitors the garbage bins and informs about the level of garbage collected in the garbage bins via a web page**.**

2.LITERATURE SURVEY:

EXISTING PROBLEM:

This project IoT Garbage Monitoring system is a very innovative system which will help to keep the cities clean. This system monitors the garbage bins and informs about the level of garbage collected in the garbage bins via a web page.

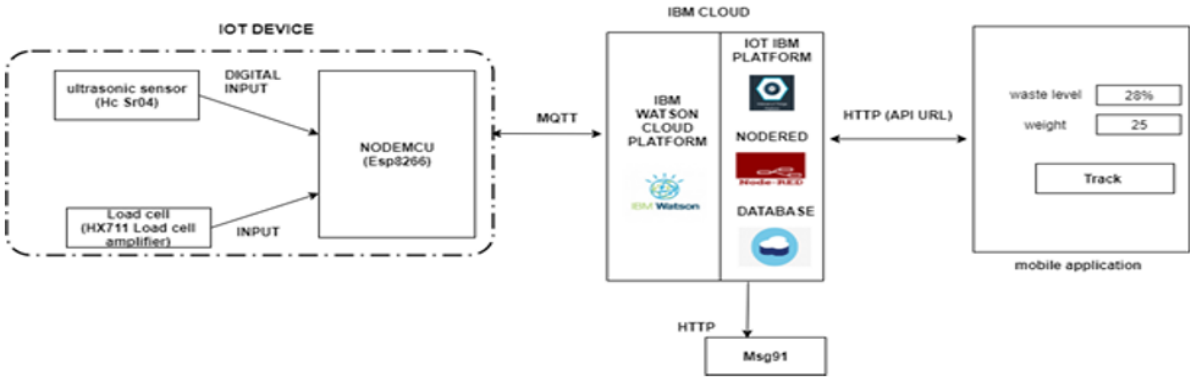
PROPOSED SOLUTION:

This project IoT Garbage Monitoring system is a very innovative system which will help to keep the cities clean. This system monitors the garbage bins and

informs about the level of garbage collected in the garbage bins via a web page.

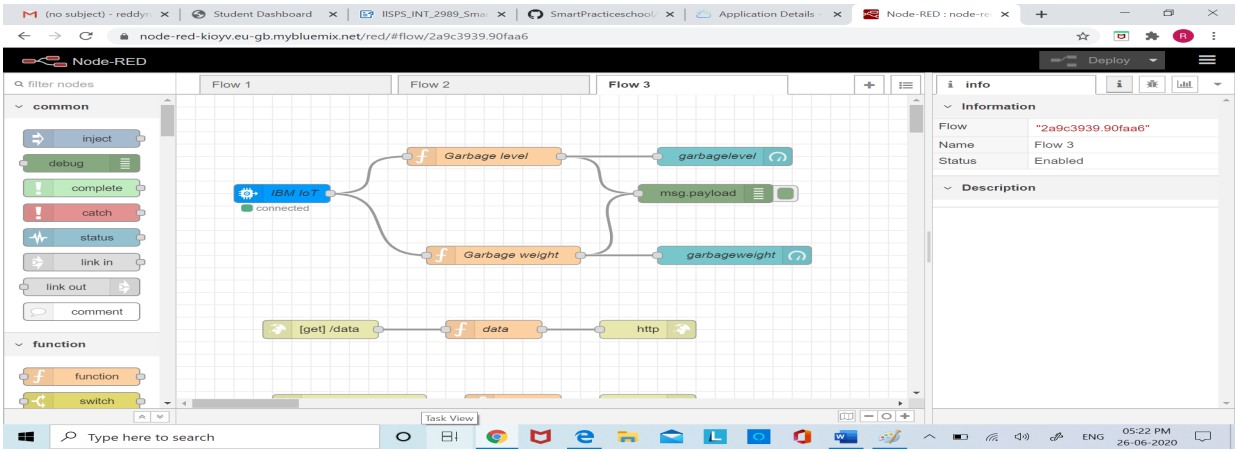
3.THEORITICAL ANALYSIS:

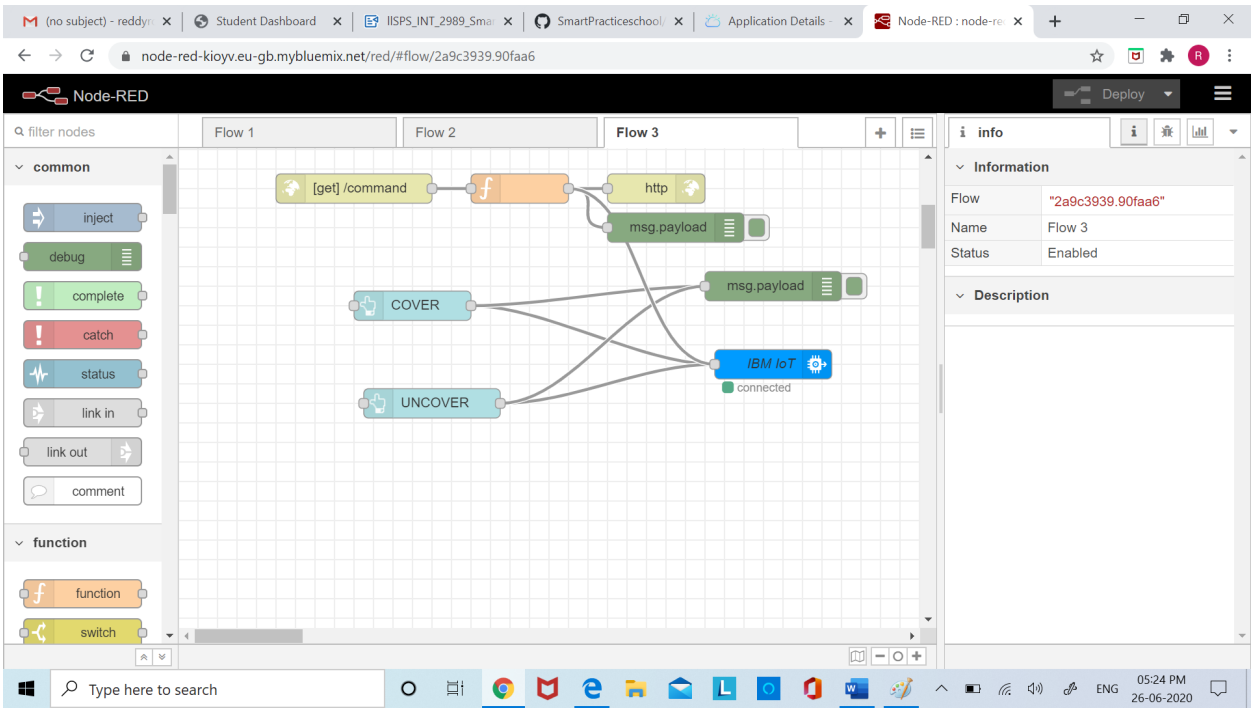
BLOCK DIAGRAM:



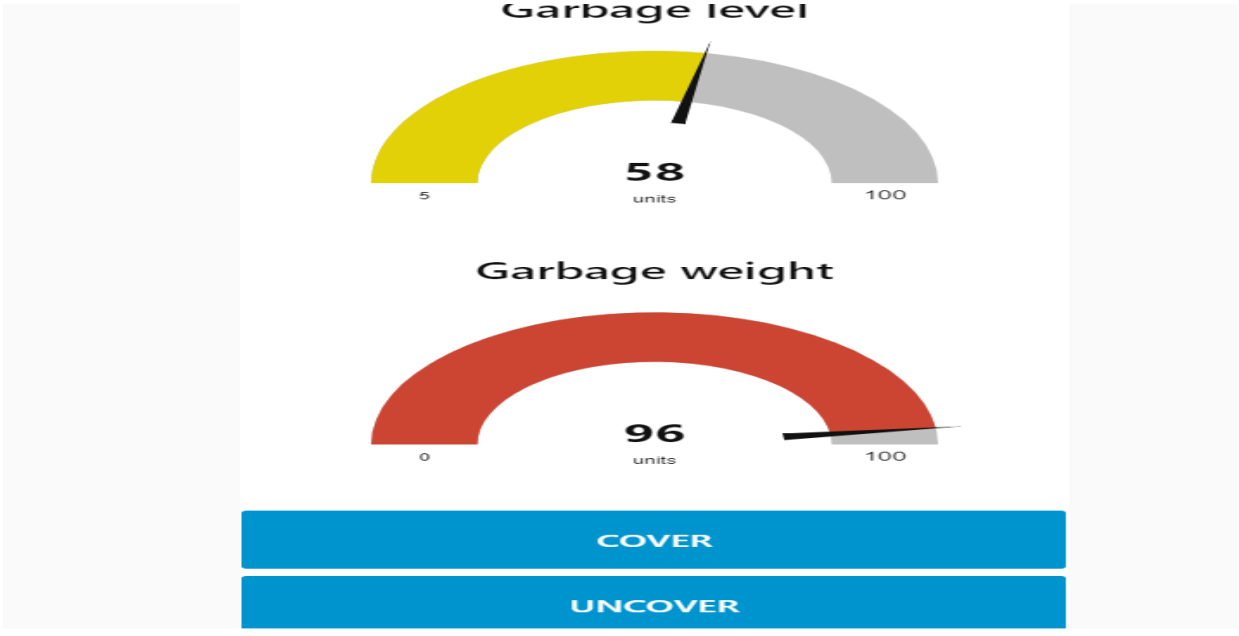
HARDWARE/SOFTWARE DESIGNING:

- Create a IBM account and in that create node red, cloudant, Watson IoT platform .
- After creating the above services go to IBM cloud dashboard, click on Cloud Foundary apps
- A new window appears where we need to NODE-RED app created before.
- After opening the node red service, take the nodes which are required for representing Garbage level, Garbage weight , debug nodes, function nodes etc.
- Connect the nodes to IBM IoT nodes to get the values to Garbage level, Garbage weight etc.



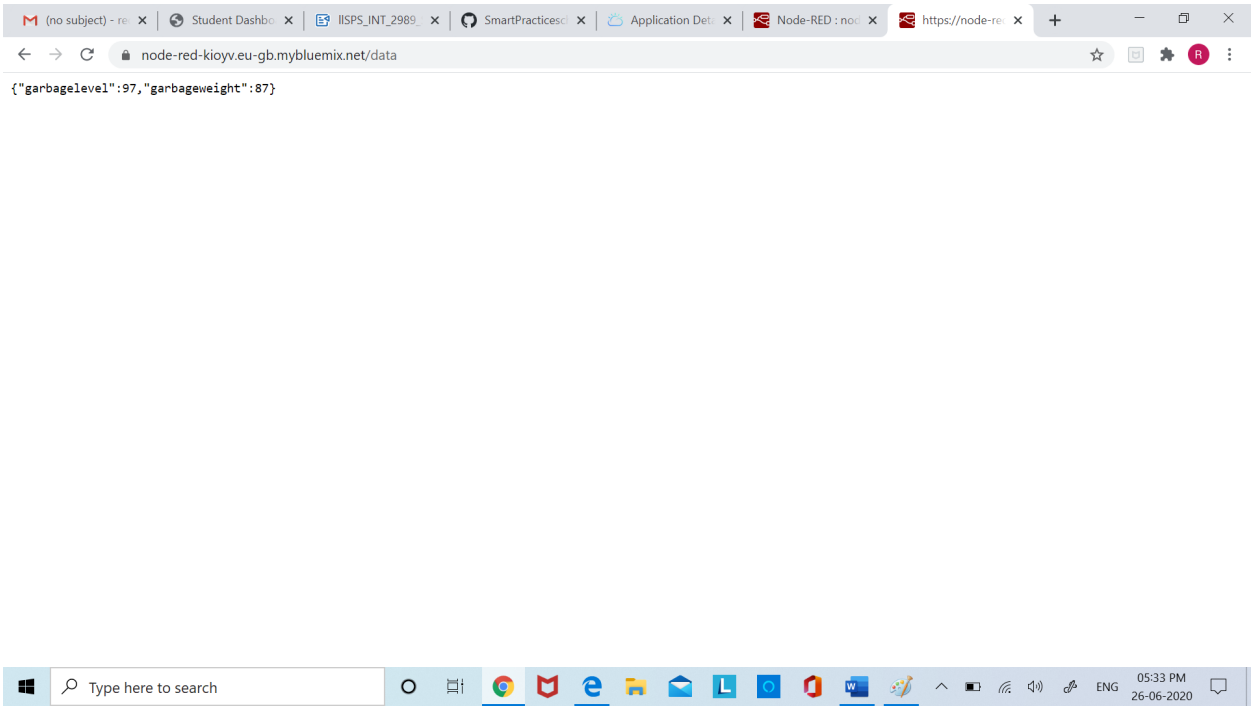


- After connecting all the nodes , Copy the Node Red URL till .net and paste in t he new tab by typing /ui along with the Node Red URL and press ENTER which will display the UI.

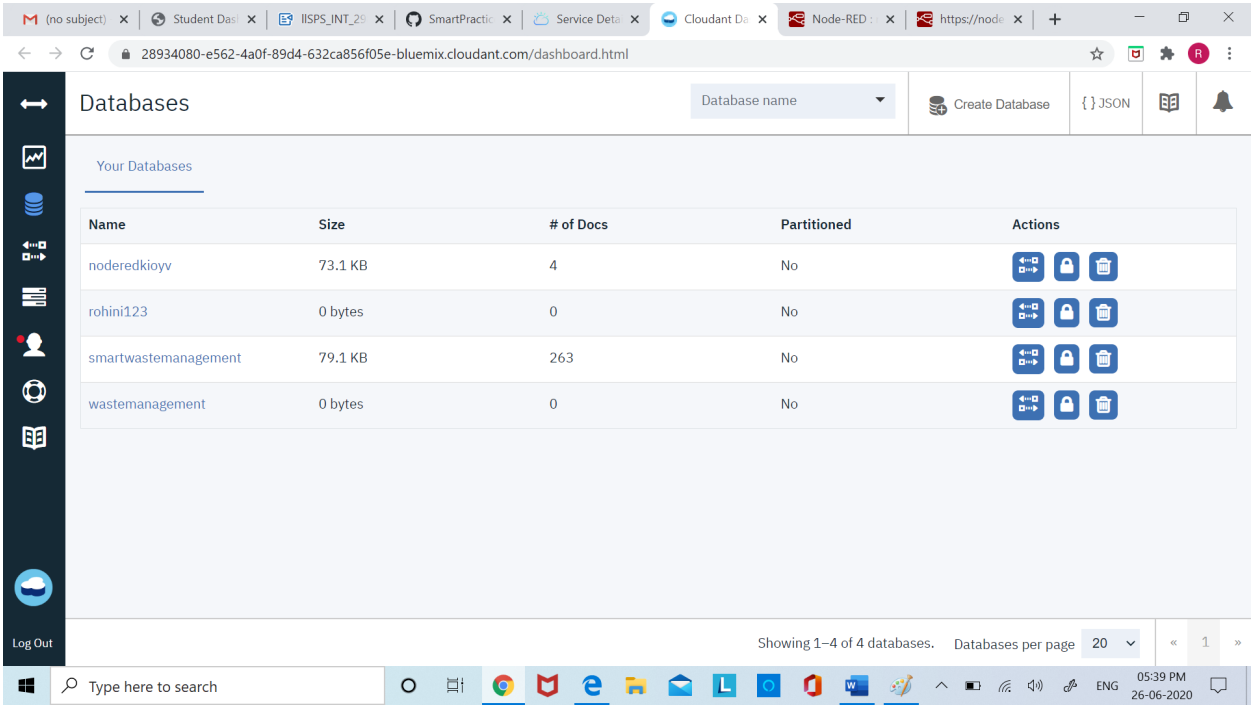


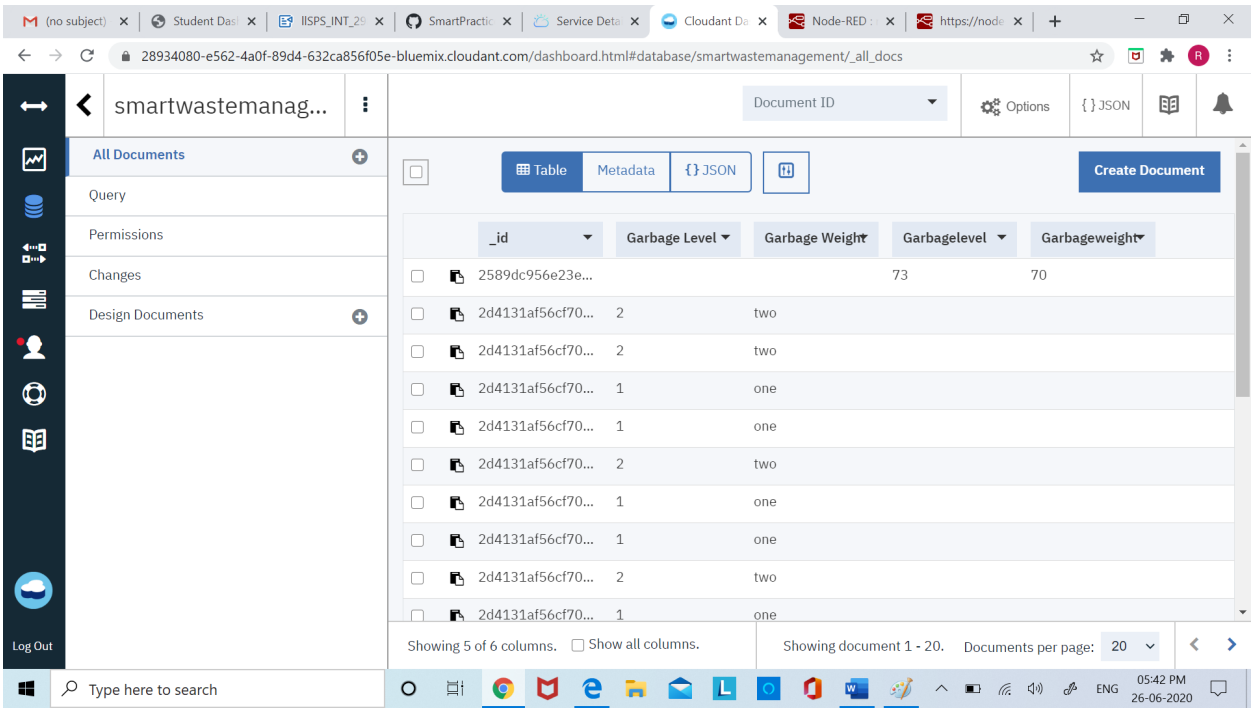
- Copy the URL in the Node Red flow till .net and paste in the new tab by a ppending

“/data” along with the URL and press Enter. Both the tank level and f low rate values will be displayed on the webpage

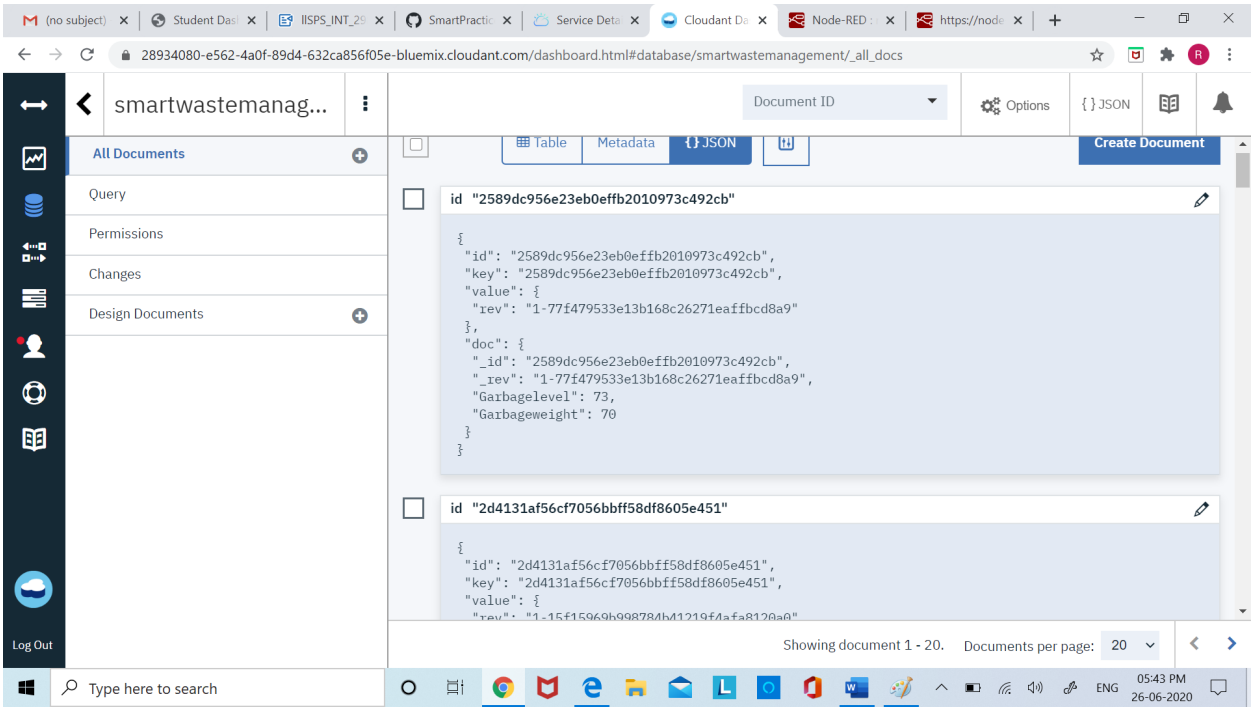


- As the values are sent to cloudant with the given file name





- The data is stored in the json format

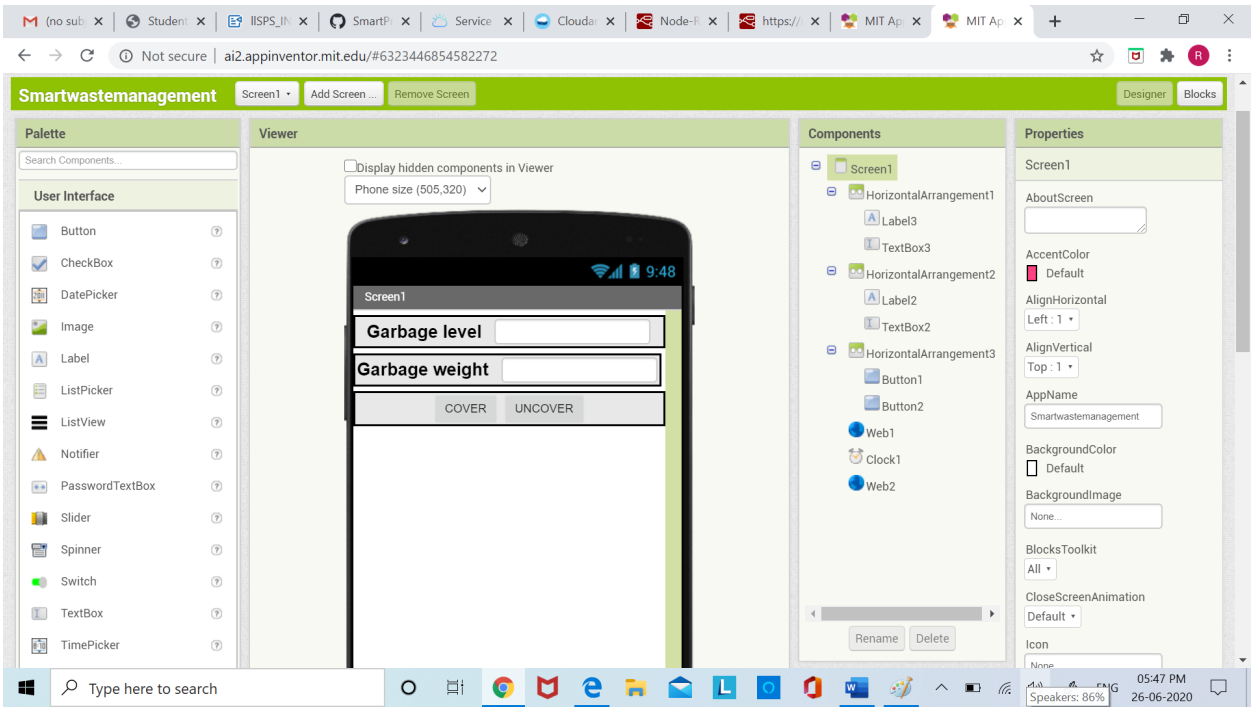


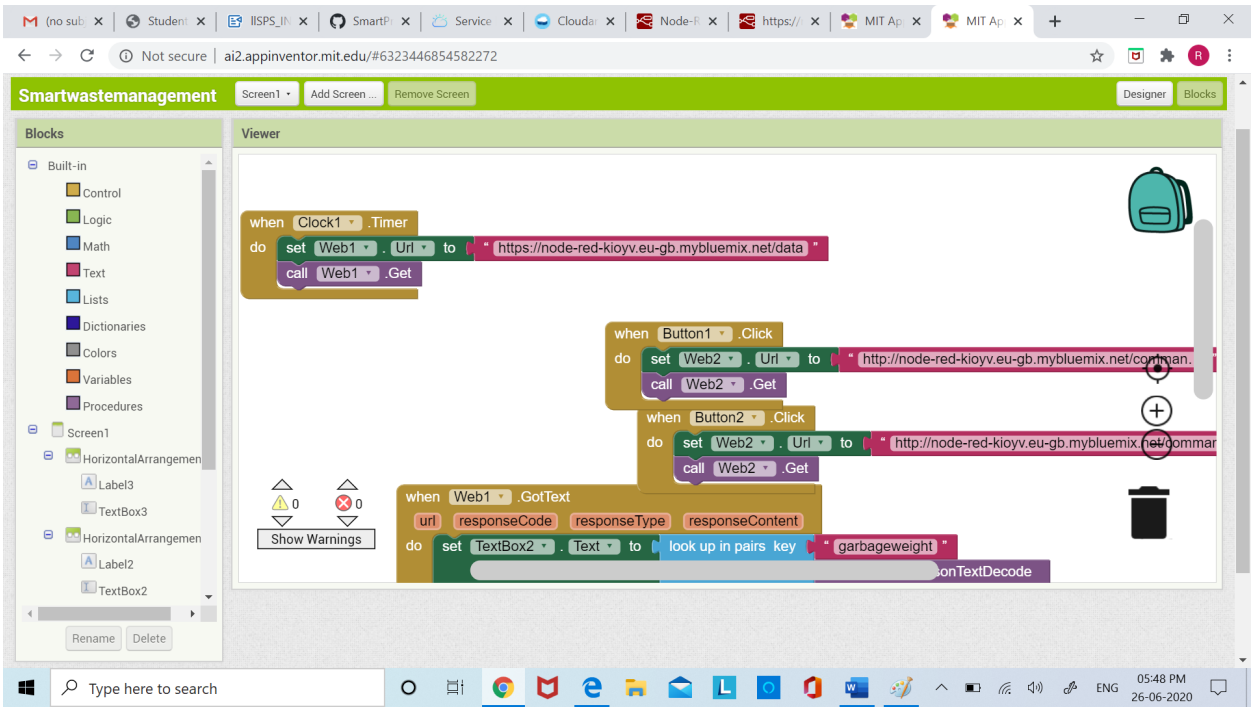
-

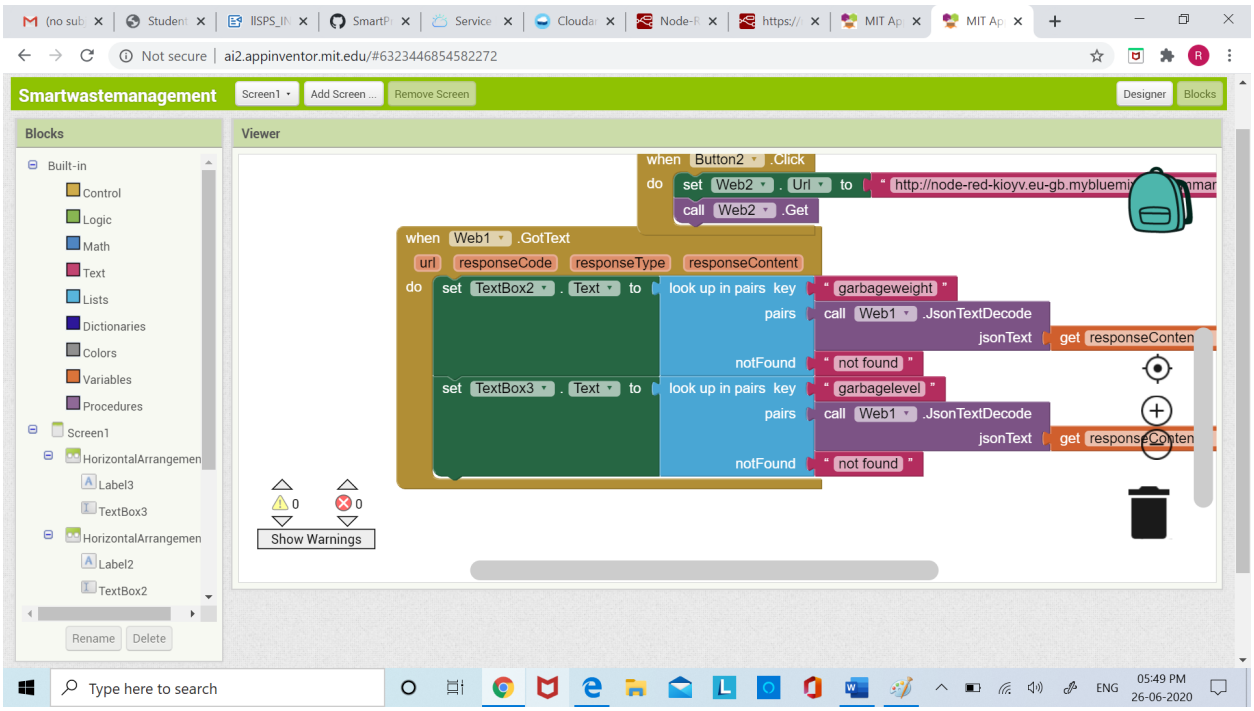
n

- Type MIT App inventor in google search and press Enter, select the first link in the search engine
- Click on the first link you will be redirected to MIT App Inventor dashboard.

- Click on Create Apps! It will redirect to the Gmail login page. Through Gmail account by typing your Username and Password, you can log in to the MIT App flow editor .
- Agree with terms and conditions. By agreeing with the terms and conditions you will be redirected to the Dashboard and click on Start new project.



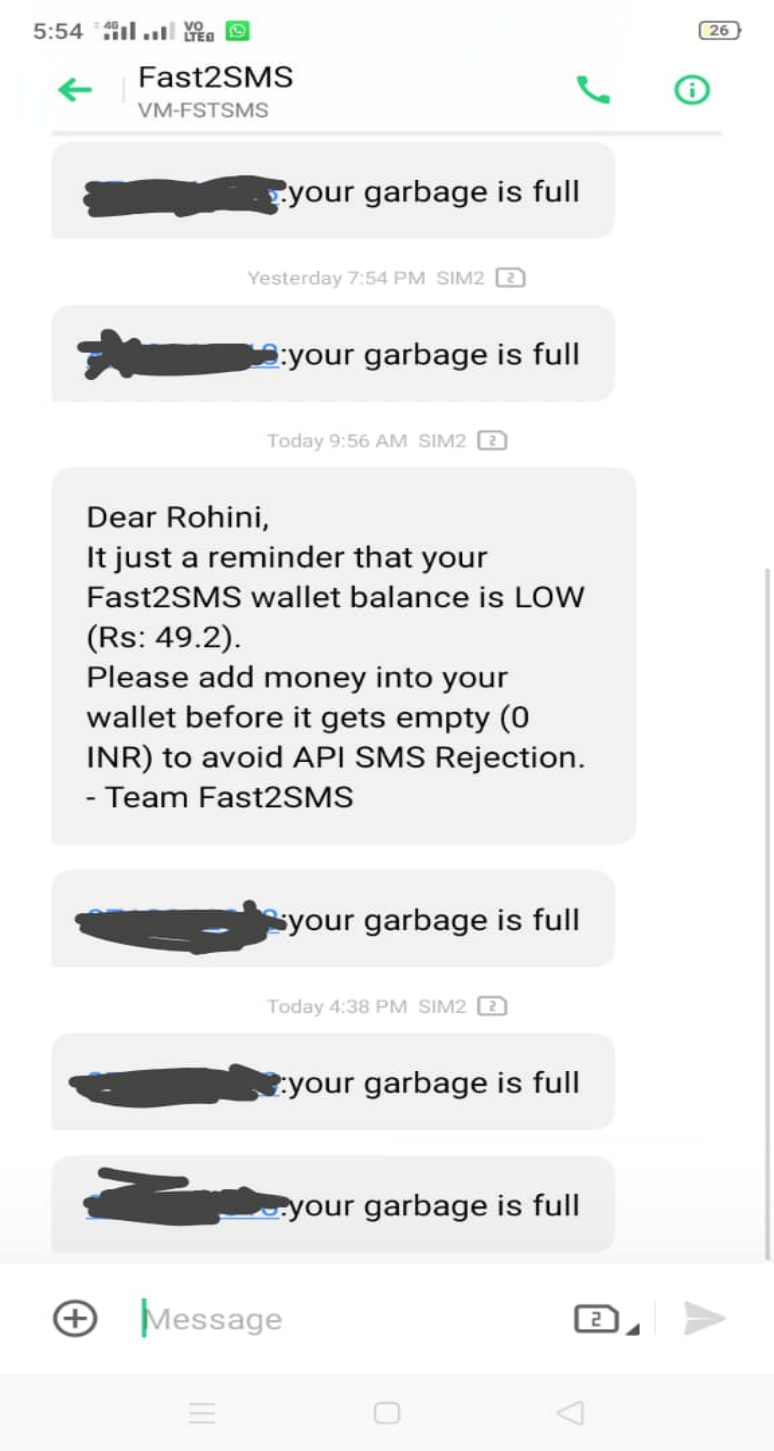




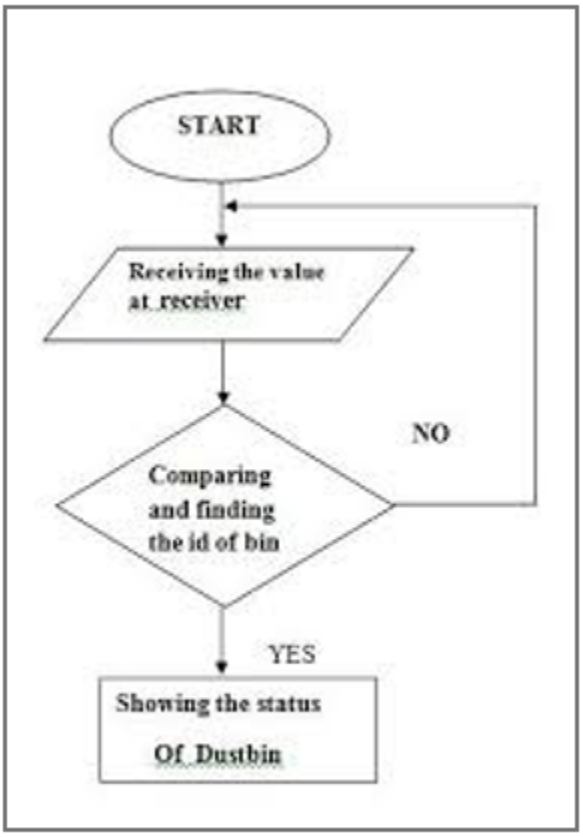
Displaying the garbage level and garbage weight using the mobile app.

Alert system: SMS alert system is implemented in such a way that SMS will be

triggered when to empty the bin and also alert is sent when ever level of garbage is full.



5.FLOW CHART:



6.RESULT:

Once see the details of waste and it alerts the person.

7.ADVANTAGES:

1 . Waste Level detection inside the garbage bins. Transmission of the information wirelessly to c oncerned officials .
2. System can be accessed anytime and from anywhere.
3. Real-time data transmission and access.
4. Avoids the overflows of garbage bins.
5. This project can only be used by municipal authorities or other private firms to tackle the c urrent problem of urban waste collection.
6.Improves Environment quality-Fewer smells-Cleaner cities.
7. This system has no individual use, but can be used by a city, state or a country.
8. Using this system, waste collection would become efficient and also reduction in t ransportation costs can be witnessed

DISADVANTAGES:

1.System requires more number of waste bins for separate waste collection as per population in t he city. This results into high initial cost due to expensive smart dustbins c ompare to other methods.
2 .Sensor nodes used in the dustbins have limited memory size.
3 . Wireless technologies used in the system such as zigbee and wifi have shorter range and l ower data speed. In RFID based systems, RFID tags are affected by surrounding metal objects (if any).
4 .Trainining It reduces man power requirements which results into increase in unemployments for unskilled people

8.APPLICATIONS:

1 . Waste Level detection inside the garbage bins. Transmission of the information wirelessly to c oncerned officials .
2. System can be accessed anytime and from anywhere.
3. Real-time data transmission and access.
4. Avoids the overflows of garbage bins.
5. This project can only be used by municipal authorities or other private firms to tackle the c urrent problem of urban waste collection.
6.Improves Environment quality-Fewer smells-Cleaner cities.
7. This system has no individual use, but can be used by a city, state or a country.
8. Using this system, waste collection would become efficient and also reduction in t ransportation costs can be witnessed.

9.CONCLUSION:

Monitoring the fullness of bins through the use of sensors, it is possible to achieve a more e fficient system than the current existing. Our idea of “Smart waste management system”, mainly concentrates on Monitoring the waste management, providing a smart technology for w aste system, avoiding human intervention, reducing human time and effort and which results i n healthy and waste ridden environment. The proposed idea can be implemented for smart c ities where the residents would be busy enough with their hectic schedule and wouldn’t have e nough time for managing waste. The bins can be implemented in a city if desired where there w ould be a large bin that can have the capacity to accumulate the waste of solid type for a single apartment.

10.FUTURE SCOPE:

There are several future works and improvements for the proposed system

1 . Change the system of user’s authentication and atomic lock of bins which would help in s ecuring the bin from any kind of damage or theft.

2. Concept of green-points that would encourage the involvement of the residents or the end u sers making the idea successful and helping to achieve joined efforts for the waste m anagement and hence fulfilling the idea of Swachch Bharath.

3. Having a case study or data analytics on the type and times the waste is collected on the type of days or season making the bin filling predictable and removing the dependency on e lectronic components and fixing the coordinates.

4. Improving graphical interfaces for the Server and complete Android applications has p ossibility of extending the system adding other use cases and applications for smart cities.

5. Moreover, the proposed solution is flexible and decoupled with respect to the determination of optimal number of bins and vehicles or to the algorithm that define the best route for vehicles.

11.BIBILOGRAPHY APPENDIX:

SOURCE CODE:

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
import requests
from cloudant.client import Cloudant
```

```
from cloudant.error import CloudantException
from cloudant.result import Result, ResultByKey

#Provide your IBM Watson Device Credentials
organization = "py1o4z"
deviceType = "dustbin"
deviceId = "636912"
authMethod = "token"
authToken = "rohini6369"

client = Cloudant("28934080-e562-4a0f-89d4-632ca856f05e-bluemix",
"c2fb9e1a7a8bf7cc45013ce5c1b8102cfba0d60bf5709f43ab623f262cd61e50",
url="https://28934080-e562-4a0f-89d4-632ca856f05e-bluemix:c2fb9e1a7a8bf7cc45013
ce5c1b8102cfba0d60bf5709f43ab623f262cd61e50@28934080-e562-4a0f-89d4-632ca
856f05e-bluemix.cloudantnosqldb.appdomain.cloud")
client.connect()
database_name = "smartwastemanagement"
my_database = client.create_database(database_name)

# Initialize GPIO

def myCommandCallback(cmd):
        print("Command received: %s" % cmd.data)
        print(type(cmd.data))
        i=cmd.data['command']
        if i=='covered':
                print("dustbin is covered")
        elif i=='uncovered':
                print("dustbin is uncovered")

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:
      L = random.randint(0, 100);
      F = random.randint(0, 100);
      data = { 'garbagelevel' : L, 'garbageweight': F}
      #print data
      def myOnPublishCallback():
          print ("Published Your Garbage_Level = %s %%" % L, "Garbage_Weight = %s %%" % F, "to IBM Watson")

                        success = deviceCli.publishEvent("event", "json", data, qos=0, on_publish=myOnPublishCallback)
      if not success:
         print("Not connected to IoTF")
      if L>90:

x=requests.get("https://www.fast2sms.com/dev/bulk?authorization=PZ6y5KcWmYAegTSJxH7RwsCE1Dba8hqI4t0fQjV9pGlFMdBoOnO3Q9Ce8fRwpysHGDvUa7Mnx5tPmI1Y&sender_id=FSTSMS&message=your garbage is full&language=english&route=p&numbers=8712262318")
            print(x.text)
      time.sleep(2)
      deviceCli.commandCallback = myCommandCallback

      if my_database.exists():
        print(f"'{database_name}' successfully created.")
      sample_data = [
         [1, "one", "garbagelevel", 67],
         [2, "two", "garbageweight", 40],]

# Create documents using the sample data.
# Go through each row in the array
```

```
        for document in sample_data:
    # Retrieve the fields in each row.
            L = document[0]
            F = document[1]

            json_document = {"Garbage Level":L ,"Garbage Weight":F}
      # Create a document using the Database API.
            new_document = my_database.create_document(json_document)
    # Check that the document exists in the database
            if new_document.exists():
                print(f"Document '{json_document}' successfully created.")

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