NALAIYATHIRAN

ARUNACHALA COLLEGE OF ENGINEERING FOR WOMEN, MANAVILAI

ANNA UNIVERSITY:: CHENNAI 600 025

SMART WASTE MANAGEMENT FOR METROPOLITAN CITIES

TEAM ID: PNT2022TMID34114

ALFHA. A (960219106019)

AMRIN FARHA MOHAMED ASLAM (960219106020)

ARFIYA M (960219106031)

ASMITHA. A (960219106042)

PROJECT GUIDE

Industry Mentor: Mr. Dinesh
Faculty Mentor: Ms. Ablin

Project Report

1. INTRODUCTION

1.1 Project Overview

With the increase in human population and the growth of global economy, there has been an increase in the waste produced. This leads to a poor and unclean environment where the dumpsters will be overflowing and the garbage being spilled out of the bins can be seen all around, thereby causing a pollution and public health disasters are the major concern in the urban areas. The biggest challenge is the complexity in sorting of waste and the collection of garbage, which is compounded when hazardous waste mixes with the general waste. Therefore, a reliable and an efficient smart waste management system needs to be properly managed to protect the human health and thereby maintaining the ecosystem. Keeping all this in mind, we propose a cloud-based smart waste management system in which the waste bins are equipped with sensors, capable of notifying their waste level status and upload the status to cloud. It will be also possible to do route optimization and select path for waste collection according to the status of waste bins in a metropolis, helping in fuel and time efficiency.

1.2 Purpose

Daily human activities generate waste which requires to be properly

managed to protect human health and environment while enhancing aesthetics. As the population is increasing the solid waste is also increasing in metropolitan areas and waste management has become a global concern. The pictures of dustbins being overfull and the garbage being spilled out from the bins can be seen all around. We need to take right decision in order to manage this overflowing garbage. It is evident that a much more efficient and effective waste management system is required to provide efficient and sustainable solid waste services, which helps in identifying the stakeholders, informing them in time about what is coming up in the waste and in what quantity. In our proposed system waste bins are connected to the cloud and data is stored there in real time.

The bins are equipped with sensors, bins update their status to the cloud and making it more efficient and convenient way to handle waste. The waste collection is also done when it is required, helping the waste management to decide a cost-effective route while collecting the waste within a metropolis.

2. LITERATURE SURVEY

2.1 Existing Problem

The existing smart waste management system is not appropriate in certain conditions which can create a negative impact on people’s health and well-being and on the environment. The improper disposal and maintenance of domestic waste is a serious health hazard and leads to the spread of infectious diseases. The lack of finances resources available, which hinders efforts towards a sustainable waste management system. This has a negative implications for public and private stakeholders.

2.2 References

S. NO	TITLE & AUTHOR	YEAR	METHODOLOGY USED	ADVANTAGE	DRAWBACK
1	IOT-Based Solid Waste Management Prof.Aderemi A.Atayero, Rotimi Williams, Segun I.Popoola, Sanjay Misra	2019	Internet Of Things and Cloud Computing	It provides autonomous power supply and helps to save more energy	Not appropriate in certain conditions
2	Smart Waste Management System using IOT Tejashree Kadus, Pawankumar, Nirmal Kartikee, Kulkarnee.	April 2020	This is based on Automation in Waste Management System and Smart net bin withWi-Fi connection is used.	Provides practical solution for managing waste and provides free internet for a specific time once the trashis dumped into the bin.	Improper disposal and maintenance of domestic waste

3	Smart Solid Waste Management System using IOT M.P. Suresh Kumar, S.Pavithran	Nov 2019	Shortest path spanning Tree Algorithm	The data stored in the server helps to compute optimized collection route for the workers.	Less collaborations between public and private stakeholders.
4	Smart Waste Management: Garbage Monitoring using IOT Mrs. Sarmila SS, Siva Kumar V, Vasanth Kumar P K	April 2018	Arduino UNO, Sensors like Ultrasonic Sensor and Gas Sensor. It employs duty cycle technique.	Reduces human intervention, time and effort.	Improper working of sensors.
5	IOT based Waste Collection System using IR sensors Abhimanyu Singh, Pankhuri Aggarwal,	April 2020	Azure Machine Learning System	The proposed method can easily provide information. It helps the company to effectively route the collection of	It relies in real time generated data and collection of waste.

	Rahul Arora			garbage.	
6	Waste Segregation System using Artificial Neutral Networks Seema Singh, Mamatha K R, Anusha N, Susmi Zacharia	2018	Classification using Convolutional Neutral Networks	It helps to achieve segregation of waste by reducing human interventions.	No physical mechanical device to categorize wastes into different bins.
7	Automatic Waste Segregator and Monitoring System Aleena V J, Kavya Balakrishnan,	2020	This method uses Ultrasonic sensor and induction sorting	Sorting of waste at the primary stage will make waste management more effective.	It is very costly, Waste separation is time consuming.

8	Intelligent Waste Separator (IWS) Oscar Rodea-Aragon, Omar Longoria-Gandara, Andres Torres Garcia	2019	Machine Learning was used here.	Avoids mixing waste in a bin, fewer ratio of error.	Capacity of waste separator do not allow in obtaining information and the response is slow.
9	Automated Waste Segregation System using Arduino Rosmi T.B, Sreejith S	2021	Arduino IDE is used here	This system separate the waste into magnetic and wet category.	Hardware failure can occur.
10	Smart Garbage Dustbin Shephali Rakhunde, Shreya Ghavghave, Shraddha Jagtap	2022	Micro controllers along with some sensors are used.	Monitors the garbage bin and informs about the level of garbage collected in the bins.	Improper working of sensors

2.2. Problem Statement Definition

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS 1	Municipality crews	manage the waste	the complexity of waste streams affects the complexity of management	of growing population and economy	pressurized
PS 2	Society	dispose waste	could not dispose properly	of overflow of bins	unpleasant
PS 3	Humans	dispatch the trash	the is not collected well	bins are not placed well	annoyed
PS 4	Citizens	segregate the trash	there is no technique for that	only few waste segregation options are available	infuriated
PS 5	Municipality members	empty the bins	it takes long time	of unfilled bins	frustrated

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

3.2 Ideation & Brainstorming

3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	The world is facing a common problem of ineffective management of waste. With the increase in human population and the growth of global economy, there has been an increase in the waste produced. This leads to a poor and unclean environment where the dumpsters will be overflowing and the garbage being spilled out of the bins can be seen all around, thereby causing a pollution and public health disasters are the major concern in the urban areas. The biggest challenge is the complexity in sorting of waste and the collection of garbage, which is compounded when hazardous waste mixes with the general waste. Therefore, a reliable and an efficient smart waste management system needs to be properly managed to protect the human health and thereby maintaining the ecosystem.

2.	Idea / Solution description	To manage waste in an effective way appears to be one of the major challenges facing by the humanity and planet. To overcome this challenge, we propose the smart waste management system characterized by the usage of technology in order to be more efficient when it comes to managing waste. This makes it possible to minimize the chance of any bin being full for over a week by using different sensors employed on the garbage bin to detect the level of trash and sort the waste accordingly within the bin itself. Thus, it enhances the proper waste management system and the segregated waste can be further recycled for different purpose.
3.	Novelty / Uniqueness	• The software algorithms automatically setup the optimum pick-up routes. • Sensors attached to the trash bin used to measure the fill level of the trash bin. • Measured data is sent to the Cloud for further processing and analysis. • By exploiting this data, trash collection can be planned as well as truck routes can be optimized. • Real-time GPS assistance directs

		the garbage truck drivers to the pre-decided route. • RFID technology is employed to identify the material to be recycled at the time of disposal.
4.	Social Impact / Customer Satisfaction	• It creates pollution free environment. • It provides real-time insights on waste fill levels, collection routes, and bin movements and locations thereby reducing the overflowing of dustbins. • Traffic reduction due to fewer collection visits helps to reduce carbon dioxide and other emissions. • Optimizing the pickup routes for garbage trucks reduce the cost of waste collection. • It boosts the circular economy because glass and plastic wastes can be recycled faster.

5.	Business Model (Revenue Model)	• Waste Management helps business make their supply chain more effective, improve ordering, reduce waste materials and save money. • It works best for developing waste to energy recycling and landfill restoration solutions. • It is suited for situations where a business is expecting to grow. • This system works well for community organization and non profits who want to become self sustaining and relay less on external contributions to achieve their mission.

6.	Scalability of the Solution	• The proposed system focuses on the implementation of sensor on preparing a community to effectively manage waste, maximize recycling, minimize waste, reduce consumption and ensures that products are made to be recycled back into nature or the marketplace. • It also provides hygienic, efficient and economic solid waste storage, collection, transportation and treatment or disposal of waste without polluting the atmosphere, soil or water. • Thus the proposed system provides to be a user friendly and makes it cheaply available in the market.

3.4 Problem Solution Fit

4. REQUIREMENT ANALYSIS

4.1 Functional Requirement

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)

FR-1	Registration and Login	User needs to login to the app by using their Gmail. Registration needs to be done by giving their name, mobile number and their locality. This is the required field.
FR-2	Subscription	Plans may vary based on the distance. A user is charged based on the number of times he/she disposes the waste.
FR-3	Smart Bin Locations	Based on the locality entered by the user, nearby active smart bins can be viewed by using Google maps. The active bins will be highlighted in orange dots.
FR-4	Bin Monitoring Details	This process gives a brief description of bins. When the user presses the orange dot, brief description of the bin will be popped up. It mainly includes the fill level of the bins and the types of waste to be disposed. When the user clicks the orange dot, he/she will be able to know whether the bin is full or empty. When the colour of dot changes to red, it represents that the bin is full and when the colour is green, it signifies that the bin is empty or semi filled.

FR-5	Rating	Based on the efficient use, feasibility and interactions with the app, user can rate between 1-10.

4.2 Non-Functional Requirement

NFR No.	**Non-Functional Requirement**	**Description**
NFR-1	Usability	Usability is a method for improving ease-of-use during the design process. Smart waste management prototype was built using IOT sensors and Cloud based Server running with custom software incorporating specialized algorithms and a graphical user interface. A model was simulated on a local machine network to check if the required goals can be met and if the proposed solution serves the purpose.

		So, it assesses how easy user interfaces are to use.
NFR-2	Security	Buy reusable water bottles, straws and lunch containers reduce trash and use of nonrecyclable plastics. Choose to purchase from companies that value sustainable practices. Composting the food waste helps to provide you with a beneficial return on your investment of time and effort. Shop eco-friendly with reusable bags
NFR-3	Reliability	Waste Management helps business make their supply chain more effective, improve ordering, reduce waste materials and save money. It works best for developing waste to energy recycling and landfill restoration solutions. It is suited for situations where a business.

NFR-4	Performance	Sensors attached to the trash bin used to measure the fill level of the trash. Measured data is sent to the Cloud for further processing and analysis. By exploiting this data, trash collection can be planned as well as truck routes can be optimized. Thus, a reduction in the number of waste collections needed by up to 80%, resulting in less manpower, emissions, fuel use and traffic congestion.
NFR-5	Availability	By using sensors, cloud server and Real-time GPS assistance directs the garbage truck drivers to the pre-decided route. Hence the waste is collected before bins get filled and unhealthy conditions occur. So, the waste is managed smarter and creates a pollution free environment.
NFR-6	Scalability	The proposed system focuses on the implementation of sensor on preparing a community to effectively manage waste, maximize recycling, minimize waste, reduce consumption and ensures that products are made to be recycled back into nature or the marketplace. Thus, it provides to be a user friendly and makes it cheaply

		available in the market.

5. PROJECT DESIGN

5.1 Data Flow Diagram

5.2 Solution & Technical Architecture

5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story/ Task	Acceptance Criteria	Priority	Release
Mobile User	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account.	High	sprint 1

Admin	Registration and login	USN-2	As an Admin, I will manage the details entered by the user.	I can manage the account	High	sprint 1
Co Admin	Login	USN-3	As a Co Admin, I will manage bin details and I will sent the information to the municipality.	I can handle bin details.	High	sprint 2
Truck Driver	Login	USN-4	As a Truck driver, I will collect the trash from the filled bins.	I can reach the bin location.	Medium	sprint 3
Municipality	Login	USN-5	As a Municipality, I will monitor the entire process.	I can manage the entire process.	High	sprint 5

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Amrin Farha Mohamed Aslam
Sprint1	Registration and Login	USN-2	As a user, I will receive confirmation email once I have registered for the application.	3	High	Asmitha A

Sprint2	Login	USN-3	As a user, I can register for the application through Facebook.	5	Low	Alfha A
Sprint3	Login	USN-4	As a user, I can register for the application through Gmail.	8	Medium	Arfiya A
Sprint4	Login	USN-5	As a user, I can log into the application by entering email & password.	12	High	Amrin Farha, Asmitha A, Alfha A, Arfiya A

6.2 Sprint Delivery Schedule

Sprint	**Total Story Points**	**Duration**	**Sprint Start Date**	**Sprint End Date (planned)**	**Story Points completed (as on Planned**	**Sprint Release Date (Actual)**

					End Date)	
Sprint-1	30	4 Days	04 Nov 2022	07 Nov 2022	5	19 Nov 2022
Sprint-2	30	4 Days	08 Nov 2022	11 Nov 2022	5	19 Nov 2022
Sprint-3	30	4 Days	12 Nov 2022	15 Nov 2022	8	19 Nov 2022
Sprint-4	30	4 Days	16 Nov 2022	19 Nov 2022	12	19 Nov 2022

6.3 Reports from JIRA

JIRA Backlog

JIRA Roadmap

Jira Burndown Chart

7. CODING AND SOLUTIONING

7.1 Feature 1

The sample registration form for the front-end application has been created using html and CSS. The first feature of the smart waste management is to find the GPS location using latitude and longitude and sending it to Node-Red using IBM Watson platform and to view the live location of bins on the map.

HTML and CSS CODE FOR WEB APPLICATION

```
<!DOCTYPE html>
<html>
<head>
  <title>Registration system PHP and MySQL</title>
  <link rel="stylesheet" href="style.css">
```

```
</head>
<body>
<div class="header">
  <h2>Register</h2>
</div>
<form method="post" action="register.php">
  <div class="input-group">
        <label>Username</label>
      <input type="text" name="username" value="">
    </div>
  <div class="input-group">
        <label>Email</label>
      <input type="email" name="email" value="">
    </div>
  <div class="input-group">
        <label>Password</label>
      <input type="password" name="password_1">
    </div>
  <div class="input-group">
       <label>Confirm password</label>
      <input type="password" name="password_2">
    </div>
  <div class="input-group">
      <button type="submit" class="btn" name="register_btn">Register</button>
```

```
    </div>
    <p>
      Already a member? <a href="login.php">Sign in</a>
        </p>
</form>
</body>
</html>
```

Style.css

```
* { margin: 0px;
padding: 0px; } body
{
    font-size: 120%;
background:
#F8F8FF;
}
.header {
    width: 40%;
margin: 50px auto
0px;          color:
white;
background:
#5F9EA0;       text-
align: center;
border: 1px solid
#B0C4DE;
border-bottom: none;
border-radius: 10px
10px 0px 0px;
```

```
  padding: 20px;
}
form, .content {
    width: 40%;
margin: 0px auto;
padding: 20px;      border:
1px solid #B0C4DE;
background: white;
border-radius: 0px 0px
10px 10px;
}
.input-group {
 margin: 10px 0px 10px 0px;
}
.input-group label {
    display: block;
text-align: left;
margin: 3px;
}
.input-group input {
    height: 30px;
width: 93%;
padding: 5px
10px;       font-
size: 16px;
border-radius:
5px;       border:
1px solid gray;
```

```
}
#user_type {
    height: 40px;
width: 98%;
padding: 5px
10px;
background:
white;      font-
size: 16px;
border-radius:
5px;      border:
1px solid gray;
}
.b
tn
{
    padding:
10px;      font-
size: 15px;
color: white;
background:
#5F9EA0;
border: none;
border-radius:
5px;
}
.error {
  width: 92%;
 margin: 0px auto;
    padding: 10px;
border: 1px solid
```

```
#a94442;     color:
#a94442;
background:
#f2dede;
border-radius: 5px;
text-align: left;

}

.success {

    color: #3c763d;
background: #dff0d8;
border: 1px solid
#3c763d;       margin-
bottom: 20px;

}

.profile_info img {

    display:
inline-block;
width: 50px;
height: 50px;
margin: 5px;
float: left;

}

.profile_info div {

    display: inline-
block;       margin:
5px;

}

.profile_info:af
ter {
```

```
content: "";
display: block;
clear: both;
}
```

REGISTRATION FORM

PYTHON CODE FOR TRACKING LIVE LOCATION OF THE BIN

```
import wiotp.sdk.device
import time import
random myConfig = {
    "identity":{
        "orgId":"j5bxb7",
        "typeId":"IOT123edevicetype",
        "deviceId":"IOTece4"
    },
    "auth": {
        "token":"e2)-17xkqIFMvm3@lI"
    }
}
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()
def pub(data):
```

```
client.publishEvent(eventId="binstatus",msgFormat="json",data="myData",qos
=0,onPublish=None)
      print("Published data Successfully:%s",myData)
while True:

myData={'name':'Bin1','lat':13.092677,'lon':80.188314}
pub(myData)        time.sleep(3)
      client.commandCallback=myCommandCallback
client.disconnect()
```

7.2 Feature 2

The second feature is that the distance between the user and the garbage bin can be detected by using the ultrasonic and PIR sensors with the help of IBM IOT Watson platform devices, IBM Cloud Interface and Node-Red for creating the dashboard nodes to display the distance and alerts the bin level on the LCD display. By detecting the distance, the lid automatically opens until it reaches the set threshold value.

CODE FOR DATA TRANSFER FROM SENSORS:

```
#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 "j5bxb7" //IBM organisation id
#define DEVICE_TYPE "IOT123edevicetype" // Device type mentioned in ibm
watson iot platform
```

```
#define DEVICE_ID "IOTece4" // Device ID mentioned in ibm watson iot platform

#define TOKEN "e2)-17xkqIFMvm3@lI" // Token

// customise above values - char server[] = ORG

".messaging.internetofthings.ibmcloud.com"; // server name  char

publishTopic[] = "iot-2/evt/data/fmt/json";

char topic[] = "iot-2/cmd/led/fmt/String"; // cmd Represent type and

command is test format of strings  char authMethod[] = "use-token-auth";

// authentication method  char token[] = TOKEN; char clientId[] = "d:"

ORG ":" DEVICE_TYPE ":" DEVICE_ID; //Client id //

WiFiClient wifiClient; // creating instance for 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(4, 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);
}
else
{
digitalWrite(15, LOW);
}
if(digitalRead(34)== true)
{
if(cm <= 100) //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 > 150 && cm < 250)
{
digitalWrite(4, HIGH);
Serial.println("Warning!!,Trash is about to cross 50% of
bin level");  digitalWrite(2,LOW); digitalWrite(23,
LOW);
}
else if(cm > 250 && cm <=400)
{
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");
}
if(cm <= 100)
{
digitalWrite(21,HIGH);
String payload = "{\"High
Alert!!\":\""; payload += cm;
payload += "left\" }";
Serial.print("\n");
Serial.print("Sending payload: ");
Serial.println(payload);
if (client.publish(publishTopic, (char*) payload.c_str())) // if data is uploaded to cloud successfully,prints publish ok or prints publish failed
{
Serial.println("Publish OK");
}
}
if(cm <= 250)
{
digitalWrite(22,HIGH);
```

```
String payload =
"{\"Warning!!\":\""; payload
+= dist;  payload += "left\" }";
Serial.print("\n");
Serial.print("Sending distance:
");  Serial.println(cm);
if(client.publish(publishTopic,(
char*) payload.c_str()))
{
Serial.println("Publish OK");
}
else
{
Serial.println("Publish FAILED");
}
}
float inches = (cm / 2.54); //print
on LCD  lcd.setCursor(0,0);
lcd.print("Inches");
lcd.setCursor(4,0);
lcd.setCursor(12,0);
lcd.print("cm");
lcd.setCursor(1,1);
lcd.print(inches, 1);
```

```
lcd.setCursor(11,1); lcd.print(cm,
1);  lcd.setCursor(14,1);
delay(1000);
lcd.clear();
}
```

OUTPUT

7.3 Feature 3

The additional feature of smart waste management system is to measure the weight of the bin by using the load cell. It also alerts the collection authority to empty the garbage bin when it reaches the threshold value.

PYTHON SCRIPT:

```
import requests
import json
import ibmiotf.application
import ibmiotf.device
import time
import random
import sys
# watson device details
organization = "j5bxb7"
devicType =
```

```
"IOT123edevicetype"
deviceId = "IOTece4"
authMethod= "token"
authToken= "e2)-
17xkqIFMvm3@lI"
#generate random values for randomo variables
(temperature&humidity) def myCommandCallback(cmd):
   global a
   print("command recieved:%s" %cmd.data['command'])
control=cmd.data['command']
   print(control)
try:
      deviceOptions={"org": organization, "type": devicType,"id":
deviceId,"authmethod":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 "temp" with value integer value into the cloud
as a type of event for every 10 seconds deviceCli.connect() while True:
   distance=
random.randint(10,70)
loadcell= random.randint(5,15)
```

```
data=
{'dist':distance,'load':loadcell}

    if loadcell < 13 and loadcell > 15:
        load = "90 %"

    elif loadcell < 8 and loadcell > 12:
        load = "60 %"

    elif loadcell < 4 and loadcell > 7:
        load = "40 %"
    else:
        load = "0 %"

    if distance < 15:
        dist = 'Risk warning:' 'Dumpster poundage getting high, Time to collect
:) 90 %'


    elif distance < 40 and distance >16:
        dist = 'Risk warning:' 'dumpster is above 60%'

    elif distance < 60 and distance > 41:
        dist = 'Risk warning:' '40 %'
```

```
    else:
      dist = 'Risk warning:' '17 %'



    if load == "90 %" or distance == "90 %":
        warn = 'alert :' ' Dumpster poundage getting high, Time to collect :)'


    elif load == "60 %" or distance == "60 %":


        warn = 'alert :' 'dumpster is above 60%'
    else :
        warn = 'alert :' 'No need to collect right now '
    def myOnPublishCallback(lat=10.678991,long=78.177731):
        print("Gandigramam, Karur")
        print("published distance = %s " %distance,"loadcell:%s "
%loadcell,"lon = %s "
%long,"lat = %s" %lat)
        print(load)
print(dist)
print(warn)


    time.sleep(10)
```

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

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



    if not success:
      print("not connected to ibmiot")
time.sleep(30)



    deviceCli.commandCallback=myCommandCallback
#disconnect the device
deviceCli.disconnect
```

8. TESTING

8.1 Test Cases

Unit Testing

Test Case no	Sensor/Stage	Input	Expected Output	Obtained Output	Status
1.	Ultrasonic	Distance between the person and the garbage bin.	Detects the exact distance.	As expected	Pass
2.	PIR	Motion of the User.	Senses the motion of the User.	As expected	Pass
3.	ESP 32	It collects and process the input data from the sensor.	It successfully interfaces with the IBM Cloud and IBM Wastson IOT platform.	As expected	Pass
4.	Load Cell	Measures the weight of the bin	Calculate the weight of bin.	As expected	Pass
5.	LCD	Data from the IBM Cloud	Displays the data	As expected	Pass

9. RESULTS

SAMPLE OUTPUT

h

10. ADVANTAGES & DISADVANTAGES

10.1 Advantages

- Easy maintenance and cost effective.
- Real-time monitoring of fill level in the bin.
- It keeps the environment clean healthy.
- It is Eco-Friendly and promotes recycling of waste.
- Reduces manpower requirement.

10.2 Dis-advantages

- Awareness should be created among people on smart waste management.
- Initial deployment cost is higher,
- More dust bins should be required.

11. CONCLUSION

Waste management is faced with a number of issues which include lack of throughput, inadequate waste data, efficiency problem, delays in collection and resistance to new technologies. Presently, waste management is a major problem for authorities who are responsible for such task because it is a costly service and it hugely impacts the environment as a whole. Thus, the proposed system was achieved through the development of smart waste management with the use of IOT.

12. FUTURE SCOPE

The moisture sensor can be implemented for segregating the waste can be created to overcome the issues related to waste segregation. Another

recommendation is the use of solar panel for power generation making its power supply autonomous and cost effective. Automatic waste collection can be made to reduce the manpower cost and efforts. Safety must be the first consideration before undertaking any of the procedures.

1. APPENDIX

SOURCE CODE

```
<!DOCTYPE html>
<html>
<head>
  <title>Registration system PHP and MySQL</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
<div class="header">
  <h2>Register</h2>
</div>
<form method="post" action="register.php">
  <div class="input-group">
        <label>Username</label>
      <input type="text" name="username" value="">
    </div>
  <div class="input-group">
        <label>Email</label>
      <input type="email" name="email" value="">
```

```
  </div>
 <div class="input-group">
       <label>Password</label>
     <input type="password" name="password_1">
  </div>
 <div class="input-group">
       <label>Confirm password</label>
     <input type="password" name="password_2">
  </div>
 <div class="input-group">
     <button type="submit" class="btn" name="register_btn">Register</button>
  </div>
  <p>
     Already a member? <a href="login.php">Sign in</a>
        </p>
</form>
</body>
</html>
```

Style.css

```
* { margin: 0px;
padding: 0px; } body
{
    font-size: 120%;
background:
#F8F8FF;
```

```
}
.header {
    width: 40%;
margin: 50px auto
0px;          color:
white;
background:
#5F9EA0;       text-
align: center;
border: 1px solid
#B0C4DE;
border-bottom: none;
border-radius: 10px
10px 0px 0px;
  padding: 20px;
}
form, .content {
    width: 40%;
margin: 0px auto;
padding: 20px;      border:
1px solid #B0C4DE;
background: white;
border-radius: 0px 0px
10px 10px;
}
.input-group {
 margin: 10px 0px 10px 0px;
}
.input-group label {
```

```
    display: block;
text-align: left;
margin: 3px;

}

.input-group input {

    height: 30px;
width: 93%;
padding: 5px
10px;       font-
size: 16px;
border-radius:
5px;       border:
1px solid gray;

}

#user_type {

    height: 40px;
width: 98%;
padding: 5px
10px;
background:
white;       font-
size: 16px;
border-radius:
5px;       border:
1px solid gray;

}
.b
tn
{

    padding:
10px;      font-
```

```
size: 15px;
color: white;
background:
#5F9EA0;
border: none;
border-radius:
5px;
}
.error {
  width: 92%;
 margin: 0px auto;
    padding: 10px;
border: 1px solid
#a94442;    color:
#a94442;
background:
#f2dede;
border-radius: 5px;
text-align: left;
}
.success {
    color: #3c763d;
background: #dff0d8;
border: 1px solid
#3c763d;      margin-
bottom: 20px;
}
.profile_info img {
    display:
inline-block;
```

```
width: 50px;
height: 50px;
margin: 5px;
float: left;

}

.profile_info div {

    display: inline-
block;       margin:
5px;

}

.profile_info:af
ter {
content: "";
display: block;
clear: both;

}
```

PYTHON CODE FOR TRACKING LIVE LOCATION OF THE BIN

```
import wiotp.sdk.device
import time import
random myConfig = {
      "identity":{

            "orgId":"j5bxb7",

            "typeId":"IOT123edevicetype",

            "deviceId":"IOTece4"

      },

      "auth": {

            "token":"e2)-17xkqIFMvm3@lI"
```

```
        }
}
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()
def pub(data):

client.publishEvent(eventId="binstatus",msgFormat="json",data="myData",qos
=0,onPublish=None)
       print("Published data Successfully:%s",myData)
while True:

myData={'name':'Bin1','lat':13.092677,'lon':80.188314}
pub(myData)         time.sleep(3)
       client.commandCallback=myCommandCallback
client.disconnect()
```

CODE FOR DATA TRANSFER FROM SENSORS:

```
#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 "j5bxb7" //IBM organisation id
#define DEVICE_TYPE "IOT123edevicetype" // Device type mentioned in ibm
```

```
watson iot platform

#define DEVICE_ID "IOTece4" // Device ID mentioned in ibm watson iot
platform

#define TOKEN "e2)-17xkqIFMvm3@lI" // Token

// customise above values - char server[] = ORG

".messaging.internetofthings.ibmcloud.com"; // server name  char

publishTopic[] = "iot-2/evt/data/fmt/json";

char topic[] = "iot-2/cmd/led/fmt/String"; // cmd Represent type and

command is test format of strings  char authMethod[] = "use-token-auth";

// authentication method  char token[] = TOKEN; char clientId[] = "d:"

ORG ":" DEVICE_TYPE ":" DEVICE_ID; //Client id //

WiFiClient wifiClient; // creating instance for 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(4, 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);
}
else
{
digitalWrite(15, LOW);
}
if(digitalRead(34)== true)
{
if(cm <= 100) //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 > 150 && cm < 250)
{
digitalWrite(4, HIGH);
Serial.println("Warning!!,Trash is about to cross 50% of
bin level");  digitalWrite(2,LOW); digitalWrite(23,
LOW);
}
else if(cm > 250 && cm <=400)
{
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");
}
if(cm <= 100)
{
digitalWrite(21,HIGH);
String payload = "{\"High
Alert!!\":\""; payload += cm;
payload += "left\" }";
Serial.print("\n");
Serial.print("Sending payload: ");
Serial.println(payload);
if (client.publish(publishTopic, (char*) payload.c_str())) // if data is uploaded to cloud successfully,prints publish ok or prints publish failed
{
Serial.println("Publish OK");
}
}
if(cm <= 250)
{
```

```
digitalWrite(22,HIGH);
String payload =
"{\"Warning!!\":\""; payload
+= dist;  payload += "left\" }";
Serial.print("\n");
Serial.print("Sending distance:
");  Serial.println(cm);
if(client.publish(publishTopic,(
char*) payload.c_str()))
{
Serial.println("Publish OK");
}
else
{
Serial.println("Publish FAILED");
}
}
float inches = (cm / 2.54); //print
on LCD  lcd.setCursor(0,0);
lcd.print("Inches");
lcd.setCursor(4,0);
lcd.setCursor(12,0);
lcd.print("cm");
lcd.setCursor(1,1);
```

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

PYTHON SCRIPT:

```
import requests
import json
import ibmiotf.application
import ibmiotf.device
import time
import random
import sys
# watson device details
organization = "j5bxb7"
devicType =
"IOT123edevicetype"
deviceId = "IOTece4"
authMethod= "token"
authToken= "e2)-
17xkqIFMvm3@lI"
#generate random values for randomo variables
```

```
(temperature&humidity) def myCommandCallback(cmd):
    global a
    print("command recieved:%s" %cmd.data['command'])
control=cmd.data['command']
    print(control)
try:
        deviceOptions={"org": organization, "type": devicType,"id":
deviceId,"authmethod":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 "temp" with value integer value into the cloud
as a type of event for every 10 seconds deviceCli.connect() while True:
    distance=
random.randint(10,70)
loadcell= random.randint(5,15)
data=
{'dist':distance,'load':loadcell}

    if loadcell < 13 and loadcell > 15:
        load = "90 %"
```

```
    elif loadcell < 8 and loadcell > 12:
        load = "60 %"

    elif loadcell < 4 and loadcell > 7:
        load = "40 %"
    else:
        load = "0 %"

    if distance < 15:
        dist = 'Risk warning:' 'Dumpster poundage getting high, Time to collect
:) 90 %'

    elif distance < 40 and distance >16:
        dist = 'Risk warning:' 'dumpster is above 60%'

    elif distance < 60 and distance > 41:
        dist = 'Risk warning:' '40 %'
    else:
      dist = 'Risk warning:' '17 %'

    if load == "90 %" or distance == "90 %":
```

```
        warn = 'alert :' ' Dumpster poundage getting high, Time to collect :)'


    elif load == "60 %" or distance == "60 %":


        warn = 'alert :' 'dumpster is above 60%'
    else :
        warn = 'alert :' 'No need to collect right now '
    def myOnPublishCallback(lat=10.678991,long=78.177731):
        print("Gandigramam, Karur")
        print("published distance = %s " %distance,"loadcell:%s "
%loadcell,"lon = %s " %long,"lat = %s" %lat)
print(load)
print(dist)
print(warn)


    time.sleep(10)



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


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

```
    if not success:
      print("not connected to ibmiot")
time.sleep(30)



    deviceCli.commandCallback=myCommandCallback
#disconnect the device
deviceCli.disconnect
```