# IBM NALAIYA THIRAN 2022-23 PROJECT REPORT

# Smart Waste Management System For Metropolitan Cities

## TEAM ID-PNT2022TMID02550

# PNT2022TMID02550

# Project Report

| | |
|---|---|
| **Team ID** | **PNT2022TMID02550** |
| **Project Name** | **Smart waste management system for metropolitan cities** |

## 1. INTRODUCTION:

### 1.1Project Overview:

Waste has become a major worry for all of us due to the global population growth and industrialisation of nations. Over the years, academics came to the conclusion that, in this age of globalisation, waste management alone is insufficient for the efficient treatment and disposal of garbage. Researchers have developed IoT-based Smart Waste Management initiatives and solutions with the aid of technology, ensuring that the time and energy needed to deliver waste management services and lower the amount of waste generated is minimised. Unfortunately, a number of variables, including the socioeconomic context, prevent developing countries from implementing those current solutions. To ensure proper household waste disposal, collection, transportation, and recycling while using the fewest resources possible, we have focused our research on creating a smart Internet of Things-based waste management system for developing nations like INDIA.

### 1.2Purpose:

To efficiently establish a secure and sanitary workplace, we combine waste management with technology. Utilizing data and technology to make the trash business more effective is known as smart waste management. Smart trash management, which is based on Internet of Things (IoT) technology, aims to maximise resource allocation, lower operating costs, and improve the sustainability of waste services. This reduces the likelihood that any bin would be full for longer than a week while also enabling the trash collectors who empty the bins to plan more effective routes. The coordination between the trash haulers and the information provided by technology is good. This informs them of the current waste level and prompts them once the trash cans approach the threshold level. So that they can collect the trash on time and avoid cluttering the neighbourhood, they are issued with alert messages. Historical data can be used to identify certain container fill trends, which can then be controlled long-term accordingly. Mobile applications are

utilised to address issues with the conventional waste management system in addition to hardware fixes, such as tracking drivers while they are out in the field. As a result, smart waste management gives us the technology-assisted method that is most effective for managing garbage.

## 2. LITERATURE SURVEY:

- Existing problem:

In local towns and cities all around the world, waste management has grown to be a serious problem. Municipalities frequently have overflowing local dumpsters without being aware of it. This has a variety of effects on the locals, from the unpleasant odour to the hazardous and unclean environment. Poor waste management, which includes everything from nonexistent collection infrastructure to inefficient disposal, contaminates the air, water, and land. Open and unclean environments can infect people, spread diseases, and lead to the contamination of drinking water. As they accumulate throughout the food chain, toxic substances like persistent organic pollutants (POPs) pose particularly serious dangers to both human health and the ecosystem. Animals who consume polluted plants receive larger dosages of pollutants than those who are exposed to them directly. Hazardous elements from landfills, agricultural areas, feedlots, etc. will be absorbed by precipitation or surface water seeping through garbage and carried into surface and groundwater. Because it is frequently utilised for drinking, bathing, pleasure, as well as in agricultural and industrial processes, contaminated groundwater also offers a serious health danger. Various pests (insects, rodents, gulls, etc.) that seek out food in waste might be drawn to landfills and waste transfer terminals. These pests pose a threat to human health because they can transmit viruses and bacteria (such as salmonella and e-coli) that cause diseases.

## 2.2References:

**1.TITLE:** Arduino Microcontroller Based Smart Dustbins for Smart Cities

**AUTHOR NAME:** K. Suresh, S. Bhuvanesh and B. Krishna Devan

**PUBLICATION YEAR:** 2019

**DESCRIPTION:**

In this paper, a technique for cleaning up our surroundings and environment is described. The Indian government just began work on a smart city initiative, and in order for these towns to be smarter than they already are, the garbage collection and disposal system must be improved upon. Self-Monitoring

Automated Route Trash (SMART) dustbins are intended for use in smart buildings such as colleges, hospitals, and bus stops, among other places. In this study, we have employed the PIR and Ultrasonic sensors to detect human presence, the Servomotor to open the dustbin lid, and the Ultrasonic sensor to detect the level of rubbish. Signals between two trash cans are transmitted using a communication module, and the GSM module sends the message to the operator.

**2.TITLE:** IoT Based Waste Management for Smart City

**AUTHOR NAME:** Parkash Tambare, Prabu Venkatachalam

**PUBLICATION YEAR:** 2016

**DESCRIPTION:**

In the current situation, we frequently observe that the trash cans or dust cans that are located in public spaces in cities are overflowing due to an increase in the amount of waste produced each day. We are planning to construct "IoT Based Waste Management for Smart Cities" to prevent this from happening because it makes living conditions for people unsanitary and causes unpleasant odours in the surrounding area. There are numerous trash cans scattered throughout the city or on the campus that are part of the proposed system. Each trash can is equipped with a low-cost embedded device that tracks the level of the trash cans and an individual ID that will enable it to be tracked and identified.

3. **AUTHOR NAME:** Mohammad Aazam, Marc St-Hilaire, Chung-Horng Lung, Ioannis Lambadaris

**PUBLICATION YEAR:** 2016

**DESCRIPTION:**

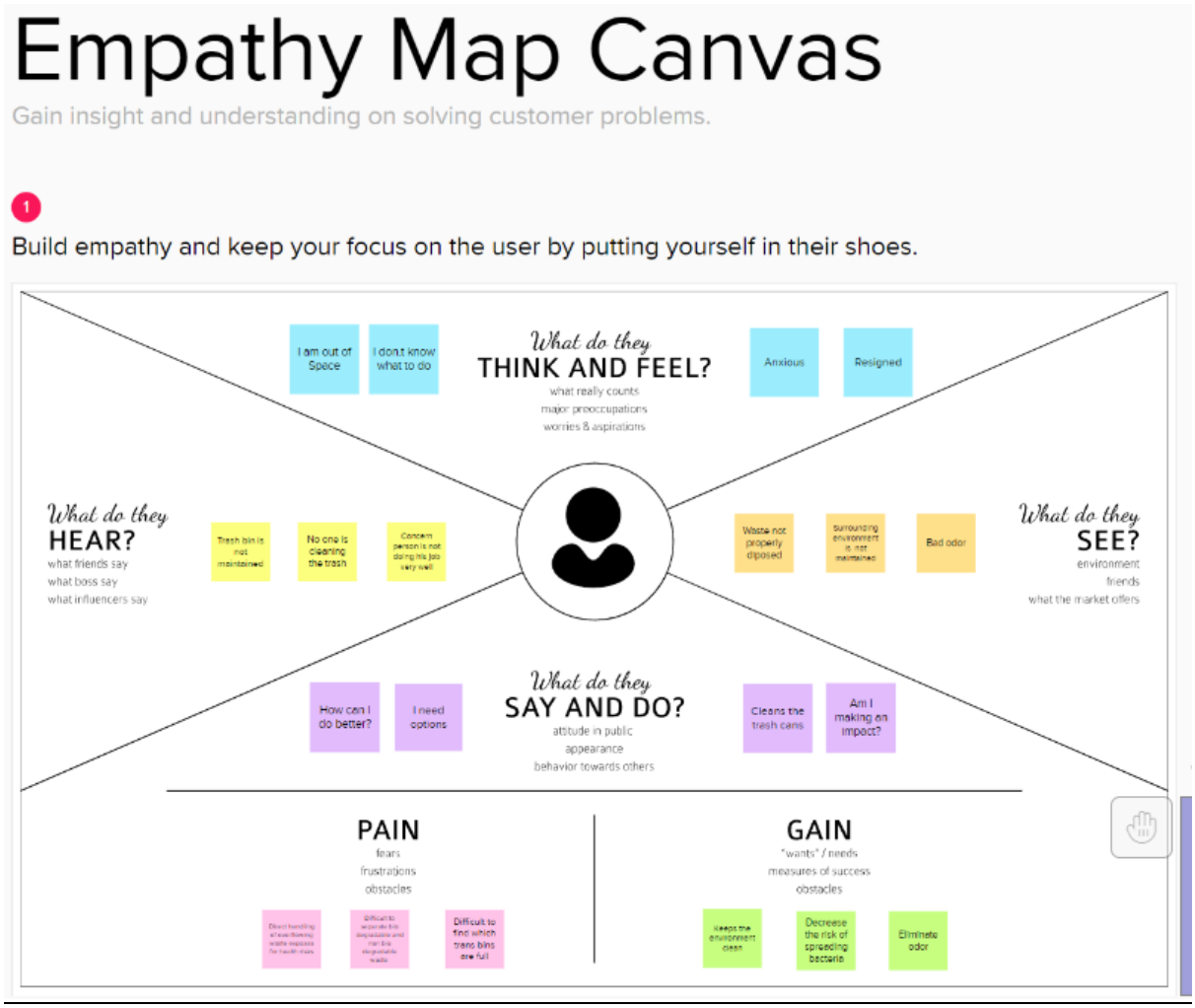
Each bin in the Cloud SWAM system that Mohammad Aazam et al suggested has sensors that can detect the amount of waste inside. There are separate bins for organic, plastic/paper/bottle/glass, and metal waste. This way, each form of waste is already divided, and it is known how much and what kind of waste is collected thanks to the status. Different entities and stakeholders may benefit from the accessibility of cloud-stored data in different ways. Analysis and planning can begin as soon as garbage is collected and continue through recycling and import/export-related activities. Timely garbage collection is provided via the Cloud SWAM system. A timely and effective method of waste collection improves health, hygiene, and disposal.

## 2.3 Problem Statement Definition:

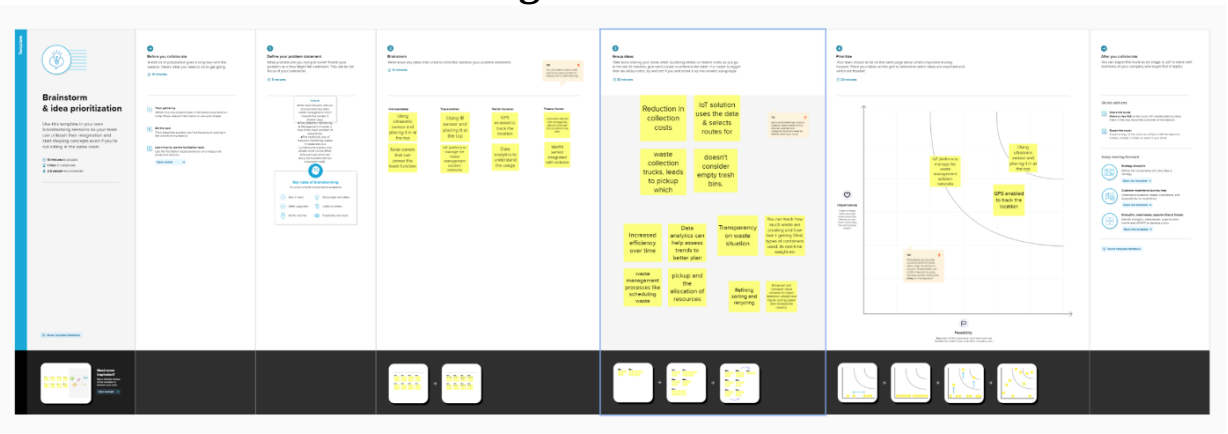
| **Problem Statement (PS)** | **I am (Customer)** | **I'm trying to** | **But** | **Because** | **Which makes me feel** |
|---|---|---|---|---|---|
| PS-1 | Municipal corporation authority | Get notified when the trash cans are full and be made aware of where the full cans are located. | Don't have the facilities at the moment | There is no tool available to determine the level of bins. | Frustrated |
| PS-2 | Individual working for a private limited corporation | Get rid of the example of a surplus of waste | The trash cans are always filled | I occupy a metropolitan where there is a city is invariably crowd. | Worried |

## 3.IDEATION & PROPOSED SOLUTION:

Empathy Map Canvas :
https://app.mural.co/invitation/mural/ibmproject6124/1663309328762?sender=u06dc1bc3927933e165684173&key=4b4a2d1a-9456-4cc8-aa3c-c4f0ac93ffea



## 3.2 Ideation & Brainstorming




## 3.3 Proposed Solution

Problem Statement :

- The main Concern with our environment has been waste management which impacts the society in several ways.
- The Detection, Monitoring & Management of waste is one of the major problem of present era.
- The traditional way of manually monitoring wastes in waste bins is a cumbersome process and utilizes more human effort, time and cost which can easily be avoided with our proposed model.

Idea / Solution description:

- By creating an app, the corporation of a certain neighbourhood inside a big city will be able to check the trash cans to see if they are full or not.
- This technique employs a cloud connection, non-biodegradable wastes, and an ultrasonic sensor to determine the amount of a rubbish container.

Novelty / Uniqueness:

- This plan advises us to use the transportation only as needed, in contrast to the conventional methods for collecting trash cans.
- Making it simpler and less time-consuming for people to keep a check on the trash cans.

Social Impact / Customer Satisfaction :

- It makes the environment cleaner for everyone and requires less human labour for waste removal.

- This concept will be very useful for a municipal corporation to check the cleanliness of various city neighbourhoods.

Business Model :

- The reduction in unnecessary transportation expenses to useless locations cuts the cost of gasoline for city enterprises significantly.
- This programme aims to help municipal corporations.Create a clean

# 4.REQUIREMENT ANALYSIS

### 4.1.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<br>Registration through Gmail<br>Registration through LinkedIN |
| FR-2 | User Confirmation | Confirmation via Email<br>Confirmation via OTP |
| FR-3 | Admin Updation | Add to the database the specifics of every trash can in the city. |
| FR-4 | Real Time Monitoring | The dashboard displays full trash cans. |
| FR-5 | Bins Updation after Cleaning | Update the information about the containers that were cleaned |
| | | |

### 4.2.Non-functional Requirements:

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

| FR No. | Non-Functional Requirement | Description |
|---|---|---|
| NFR-1 | **Usability** | Usability is a unique and significant perspective to examine user requirements, which can further enhance the design quality, according to IoT devices. Analyzing how well people |

| | | |
|---|---|---|
| | | interact with a product can help designers better understand customers' prospective demands for waste management, behaviour, and experience in the design process when user experience is at the centre. |
| NFR-2 | **Security** | Since the module provides real-time data rather than stored data, changing the data of the bins is impossible. |
| NFR-3 | **Reliability** | The module tries to provide correct data and is very reliable because it operates around-the-clock. |
| NFR-4 | **Performance** | Ultrasound technology is used by the Smart Sensors to<br>Bin fill levels should be measured together with other information.<br>multiple times per day. Using different IoT networks<br>The sensors provide the data via (NB-IoT, GPRS),<br>The Smart Waste Management Software from Sensoneo strong cloud-based platform System for<br>Everyday processes that are data-driven are likewise available as waste management software.<br>Customers are thus given data-driven decision-making options.<br>the creation and improvement of garbage collection routes, frequencies, and the route's resulting car loads<br>decrease of at least 30% |
| NFR-5 | **Availability** | By creating and deploying durable hardware<br>utilising gorgeous software, we enable enterprises, cities, and<br>better waste management in nations. |
| NFR-6 | **Scalability** | Scaling the modules is fairly simple because they contain very little hardware and software. |

# 1. PROJECT DESIGN

5.1 Data Flow Diagrams

• The conventional visual representation of how information moves through a system is a data flow diagram (DFD).

•

A tidy and understandable DFD can graphically represent the appropriate quantity of the system demand.

•

It demonstrates how information enters and exits the system, what modifies the data, and where information is kept.

• An intelligent trash management platform use analytics to convert the information gathered in your bins into useful insights that can be used to

You can receive data on metric such as:

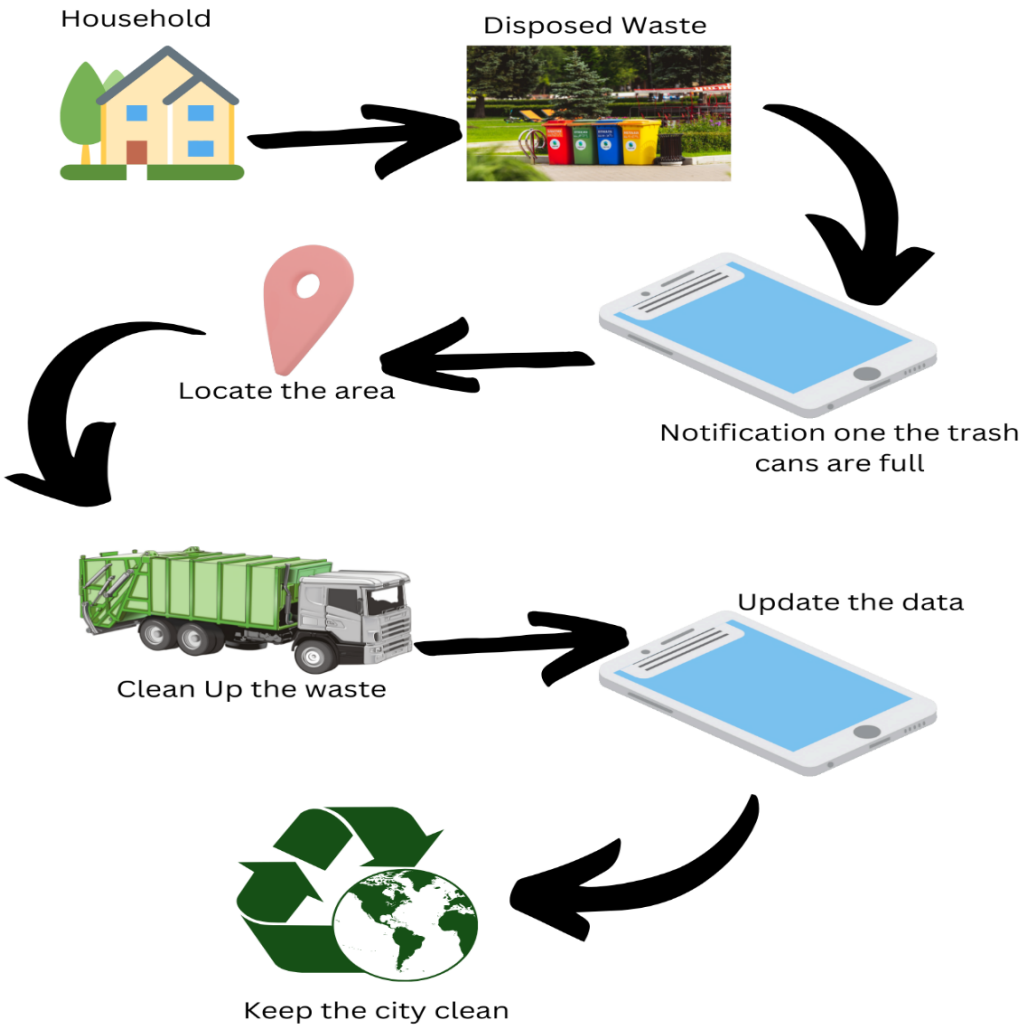
- The first test is when the trash can is empty or has a very low level of garbage.
- After that, the trash can is filled with more garbage until it reaches the first threshold value, which is set at 80%.
- At that point, the system sends the first warning SMS, as shown.
- After the trash can reaches the level of 85% full, the system sends the second notification SMS, which indicates that the trash can is at least 95% full.
- Overflow-prone areas, the number of bins required, the number of collection services that could be avoided, the amount of gasoline.

## 5.2 Data flow diagram:

| | |
|---|---|
| Date | 03 October 2022 |
| Team ID | PNT2022TMID02550 |
| Project Name | Project - Smart Waste management System for Metropolitan Cities |
| Maximum Marks | 4 Marks |

**Data Flow Diagrams:**

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.


Household
Disposed Waste
Notification one the trash cans are full
Locate the area
Clean Up the waste
Update the data
Keep the city clean

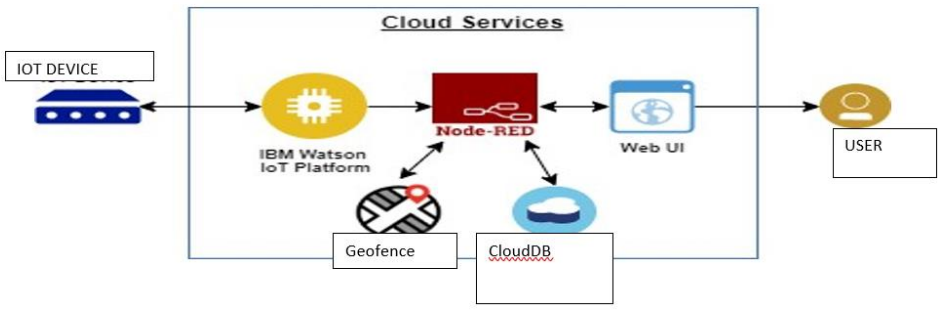
**User Stories**

| User Type | Functional Requirement (Epic) | User Story Number | User Story / Task | Acceptance criteria | Priority | Release |
|---|---|---|---|---|---|---|
| Customer (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 / dashboard. | High | Sprint-1 |
| Customer | Organise | USN-2 | Organize an additional monitoring tasks, such as garbage level monitoring. Precision in location, rubbish separation, and timely waste removal. | Monitor Garbage bin activity. | High | Sprint-2 |
| Customer | Monitor | USN-3 | Here comes the client, who will have access to mobile applications or login webpages to watch bin progress and report any issues. | He/She has right to make query. | High | Sprint-3 |
| Sanitary Worker | Clean | USN-4 | Here, a truck driver is a person who has certain responsibilities and who must report when and where, in accordance with the daily plan, the rubbish has been picked up. And ought to update the events on the specified website (webpage login). | Update the activity in the website. | High | Sprint-4 |

# 5.2 Solution & Technical Architecture:

| | |
|---|---|
| Date | 26-10-2022 |
| Team ID | PNT2022TMID02550 |
| Project Name | Project: Smart Waste management System for Metropolitan Cities |
| Maximum Marks | 4 Marks |

**Technical Architecture:**




| S.No | Component | Description | Technology |
|---|---|---|---|
| 1. | User Interface | The communication protocol being used in the proposed solution might act as an interface the way like Wi-Fi, Bluetooth and ZigBee | MIT app |

| | | | |
|---|---|---|---|
| 2. | Application Logic | To obtain the location of garbage cans that are being filled, the website is updated, and an alert is issued to the relevant authority. | IBM Watson STT service, python etc |
| 3. | Database | Data to be segregated and secured in the form of relational DBMS | MySQL |
| 4. | Cloud Database | IBM | IBM Cloudant |
| 5. | File Storage | File storage requirements | IBM Block Storage or Other Storage Service or Local Filesystem |
| 6. | External API-1 | To access the location of trash bins which are getting filled | GPS location monitoring etc |
| 7. | Infrastructure (Server / Cloud) | Application Deployment on Local System / Cloud Local Server Configuration | Cloud Foundry |

## 5.3 User Stories

Use the below template to list all the user stories for the product.

| **User Type** | **Functional Requirement (Epic)** | **User Story Number** | **User Story / Task** | **Acceptance criteria** | **Priority** | **Release** |
|---|---|---|---|---|---|---|
| Admin | Login | USN-1 | As an administrator, I assigned user names and passwords to each employee and managed them. | I can control my online account and dashboard. | Medium | Sprint-1 |
| Co-Admin | Login | USN-2 | As a Co-Admin, I'll control the waste level monitor. If a garbage filling alert occurs, I will notify the trash truck of the location and rubbish ID. | I can handle the waste collection. | High | Sprint-1 |
| Truck Driver | Login | USN-3 | As a Truck Driver, I'll follow Co Admin's instruction to reach the filled garbage. | I can take the shortest path to reach the waste filled route specified. | Medium | Sprint-2 |
| Local Garbage Collector | Login | USN-4 | As a Local Garbage Collector, I'II gather all the waste from the garbage, load it onto a garbage truck, and | I can collect the trach, pull it to the truck, and send it out. | Medium | Sprint-3 |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | deliver it to Landfills | | | |
| Municipality officer | Login | USN-5 | As a Municipality officer, I'll make sure everything is proceeding as planned and without any problems. | All of these processes are under my control. | High | Sprint-4 |

# 5.PROJECT PLANNING & SCHEDULING

## Sprint Planning & Estimation

## Sprint 1

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <ArduinoJson.h>
WiFiClient wifiClient;
#define ORG "nafgr4"
#define DEVICE_TYPE "Bin1"
#define DEVICE_ID "Tiruvallur"
#define TOKEN "T12345678"
#define speed 0.034
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/status1/fmt/json";
char topic[] = "iot-2/cmd/home/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
PubSubClient client(server, 1883, wifiClient);
void publishData();
const int trigpin=5;
const int echopin=19;
String command;
String data="";
String latitude="13.1231";
String longitude="79.9120";
long duration;
int dist;
String icon;
void setup()
{
Serial.begin(115200);
pinMode(trigpin, OUTPUT);
pinMode(echopin, INPUT);
wifiConnect();
mqttConnect();
}
void loop() {
publishData();
delay(500);
if (!client.loop()) {
mqttConnect();
}
}
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(".");
Serial.print("*");
delay(1000);
}
initManagedDevice();
Serial.println();
}
}
void initManagedDevice() {
if (client.subscribe(topic)) {
Serial.println(client.subscribe(topic));
Serial.println("subscribe to cmd OK");
}
else {
Serial.println("subscribe to cmd FAILED");
}
}
void publishData()
{
digitalWrite(trigpin,LOW);
digitalWrite(trigpin,HIGH);
delayMicroseconds(10);
digitalWrite(trigpin,LOW);
duration=pulseIn(echopin,HIGH);
dist=duration*speed/2;
if(dist<20){
icon="Bin is Full";
}
else{
icon="Bin is not Full";
}
DynamicJsonDocument doc(1024);
String payload;
doc["Latitude"]=latitude;
doc["Longitude"]=longitude;
doc["Distance"]=dist;
```

```
doc["Bin Status"]=icon;
serializeJson(doc, payload);
delay(3000);
Serial.print("\n");
Serial.print("Sending payload: ");
Serial.println(payload);
if (client.publish(publishTopic, (char*) payload.c_str())) {
Serial.println("Publish OK");
}
else {
Serial.println("Publish FAILED");
}
}
```

Bin2

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <ArduinoJson.h>
WiFiClient wifiClient;
#define ORG "nafgr4"
#define DEVICE_TYPE "Bin2"
#define DEVICE_ID "Chennai"
#define TOKEN "C12345678"
#define speed 0.034
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/status1/fmt/json";
char topic[] = "iot-2/cmd/home/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
PubSubClient client(server, 1883, wifiClient);
void publishData();
const int trigpin=5;
const int echopin=19;
String command;
String data="";
String latitude="13.0827";
String longitude="80.2707";
long duration;
int dist;
String icon;
void setup()
{
Serial.begin(115200);
pinMode(trigpin, OUTPUT);
pinMode(echopin, INPUT);
```

```
wifiConnect();
mqttConnect();
}
void loop() {
publishData();
delay(500);
if (!client.loop()) {
mqttConnect();
}
}
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(".");
Serial.print("*");
delay(1000);
}
initManagedDevice();
Serial.println();
}
}
void initManagedDevice() {
if (client.subscribe(topic)) {
Serial.println(client.subscribe(topic));
Serial.println("subscribe to cmd OK");
}
else {
Serial.println("subscribe to cmd FAILED");
}
}
void publishData()
{
digitalWrite(trigpin,LOW);
digitalWrite(trigpin,HIGH);
delayMicroseconds(10);
digitalWrite(trigpin,LOW);
duration=pulseIn(echopin,HIGH);
```

```
dist=duration*speed/2;
if(dist<20){
icon="Bin is Full";
}
else{
icon="Bin is not Full";
}
DynamicJsonDocument doc(1024);
String payload;
doc["Latitude"]=latitude;
doc["Longitude"]=longitude;
doc["Distance"]=dist;
doc["Bin Status"]=icon;
serializeJson(doc, payload);
delay(3000);
Serial.print("\n");
Serial.print("Sending payload: ");
Serial.println(payload);
if (client.publish(publishTopic, (char*) payload.c_str())) {
Serial.println("Publish OK");
}
else {
Serial.println("Publish FAILED");
}
}
```

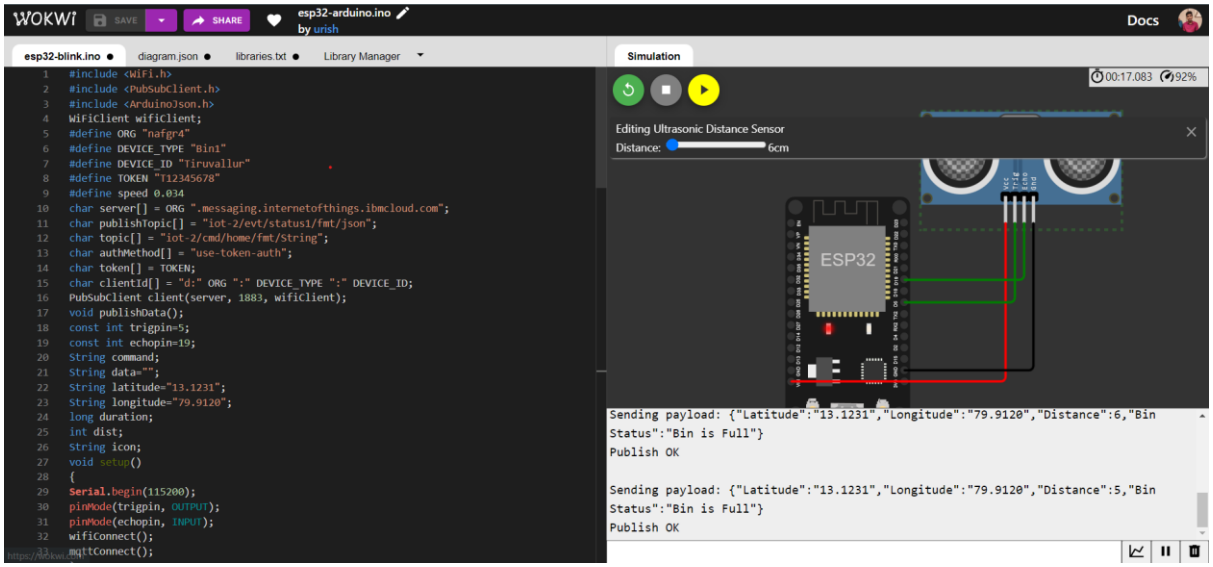
Bin3

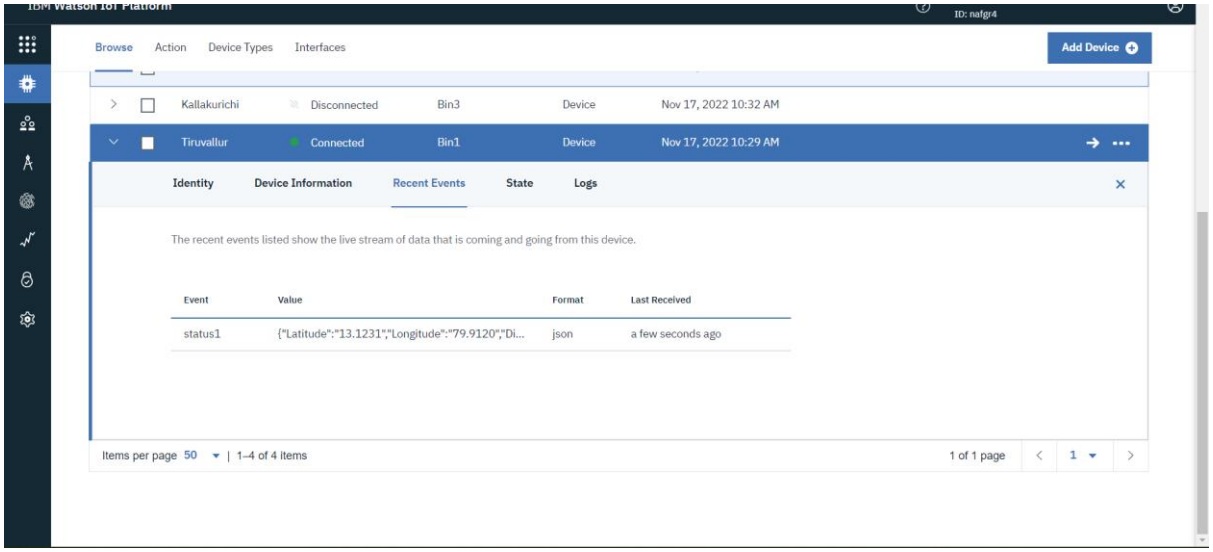
```
#include <WiFi.h>
#include <PubSubClient.h>
#include <ArduinoJson.h>
WiFiClient wifiClient;
#define ORG "nafgr4"
#define DEVICE_TYPE "Bin3"
#define DEVICE_ID "Kallakurichi"
#define TOKEN "K12345678"
#define speed 0.034
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/status1/fmt/json";
char topic[] = "iot-2/cmd/home/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
PubSubClient client(server, 1883, wifiClient);
void publishData();
const int trigpin=5;
const int echopin=19;
String command;
```

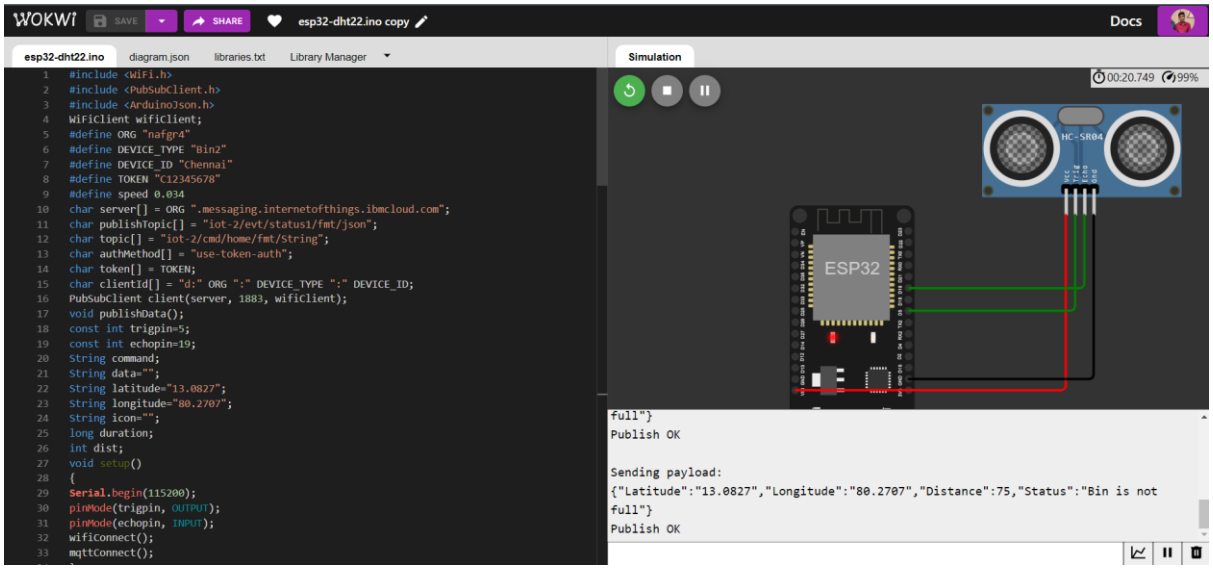
```
String data="";
String latitude="11.7348";
String longitude="78.9639";
String icon="";
long duration;
int dist;
void setup()
{
Serial.begin(115200);
pinMode(trigpin, OUTPUT);
pinMode(echopin, INPUT);
wifiConnect();
mqttConnect();
}
void loop() {
publishData();
delay(500);
if (!client.loop()) {
mqttConnect();
}
}
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(".");
Serial.print("*");
delay(1000);
}
initManagedDevice();
Serial.println();
}
}
void initManagedDevice() {
if (client.subscribe(topic)) {
Serial.println(client.subscribe(topic));
Serial.println("subscribe to cmd OK");
}
```

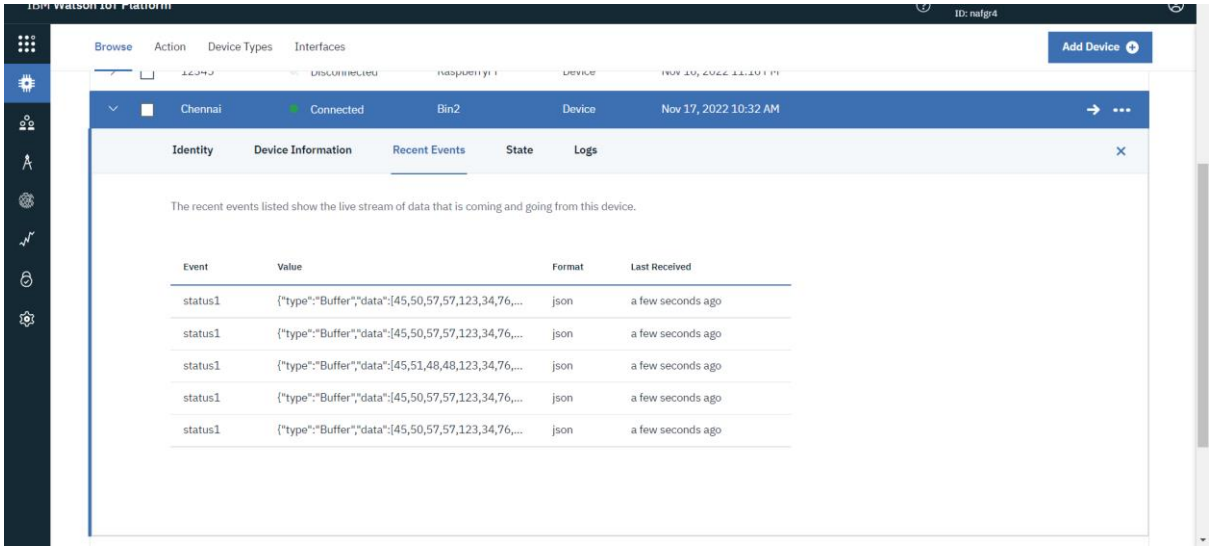
```
else {
Serial.println("subscribe to cmd FAILED");
}
}
void publishData()
{
digitalWrite(trigpin,LOW);
digitalWrite(trigpin,HIGH);
delayMicroseconds(10);
digitalWrite(trigpin,LOW);
duration=pulseIn(echopin,HIGH);
dist=duration*speed/2;
if(dist<20){
icon="Bin is full";
}
else{
icon="Bin is not full";
}
DynamicJsonDocument doc(1024);
String payload;
doc["Latitude"]=latitude;
doc["Longitude"]=longitude;
doc["Distance"]=dist;
doc["Status"]=icon;
serializeJson(doc, payload);
delay(3000);
Serial.print("\n");
Serial.print("Sending payload: ");
Serial.println(payload);
if (client.publish(publishTopic, (char*) payload.c_str())) {
Serial.println("Publish OK");
}
else {
Serial.println("Publish FAILED");
}
}
```

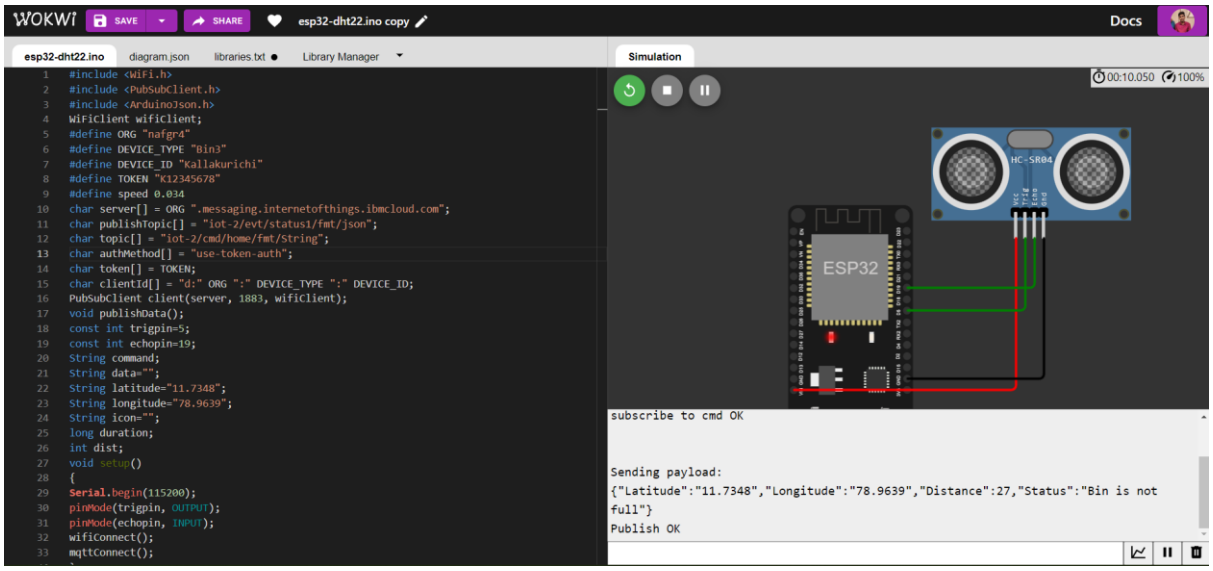
Sprint2:


WOKWI
SAVE
SHARE
esp32-arduino.ino
by urish
Docs
esp32-blink.ino
diagram.json
libraries.txt
Library Manager
Simulation
00:17.083
92%
Editing Ultrasonic Distance Sensor
Distance:
6cm
ESP32
HC-SR04
Sending payload: {"Latitude":"13.1231","Longitude":"79.9120","Distance":6,"Bin
Status":"Bin is Full"}
Publish OK
Sending payload: {"Latitude":"13.1231","Longitude":"79.9120","Distance":5,"Bin
Status":"Bin is Full"}
Publish OK

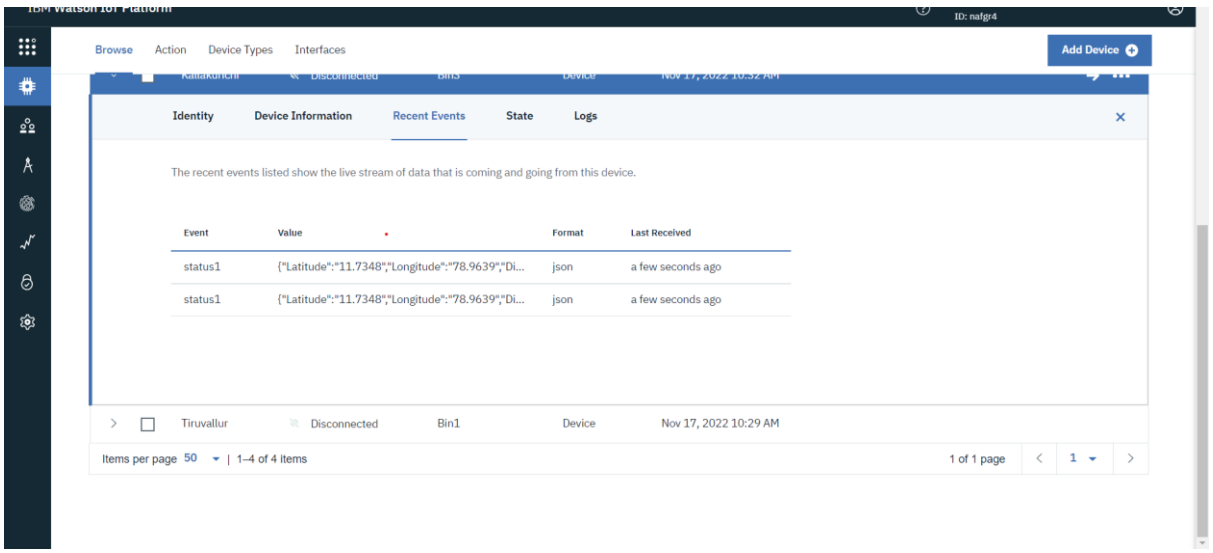

IBM Watson IoT Platform
ID: nafgr4
Browse
Action
Device Types
Interfaces
Add Device
Kallakurichi
Disconnected
Bin3
Device
Nov 17, 2022 10:32 AM
Tiruvallur
Connected
Bin1
Device
Nov 17, 2022 10:29 AM
Identity
Device Information
Recent Events
State
Logs
The recent events listed show the live stream of data that is coming and going from this device.
Event
Value
Format
Last Received
status1
{"Latitude":"13.1231","Longitude":"79.9120","Di...
json
a few seconds ago
Items per page 50
1–4 of 4 items
1 of 1 page
1


WOKWI
SAVE
SHARE
esp32-dht22.ino copy
Docs
esp32-dht22.ino
diagram.json
libraries.txt
Library Manager
Simulation
00:20.749
99%
ESP32
HC-SR04
full"}
Publish OK
Sending payload:
{"Latitude":"13.0827","Longitude":"80.2707","Distance":75,"Status":"Bin is not
full"}
Publish OK

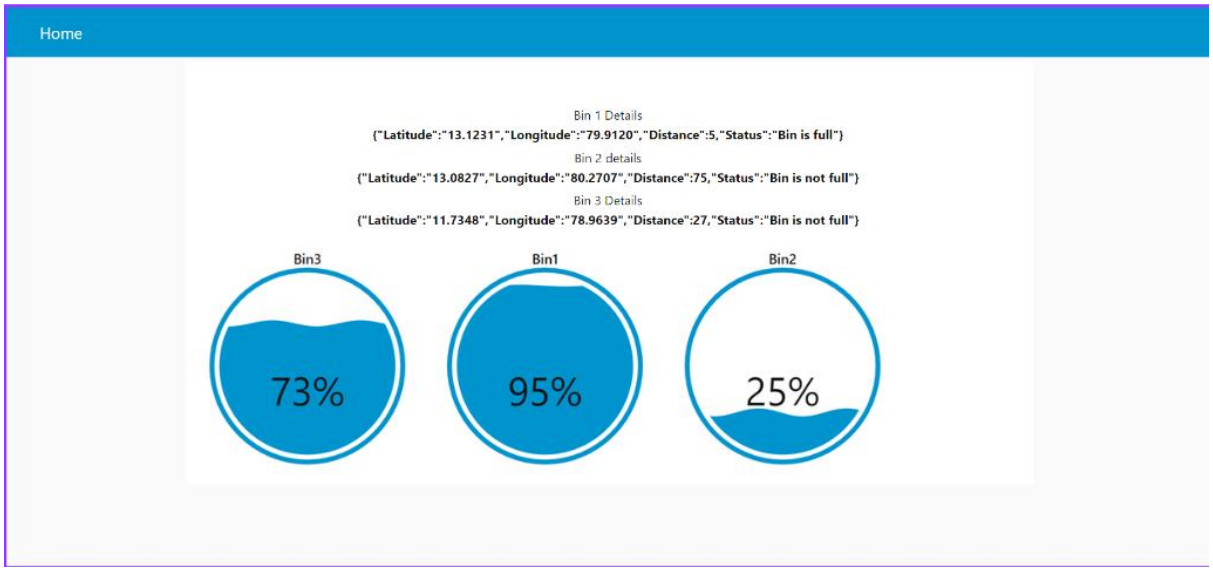



```
#include <WiFi.h>
#include <PubSubClient.h>
#include <ArduinoJson.h>
WiFiClient wifiClient;
#define ORG "nafgr4"
#define DEVICE_TYPE "Bin3"
#define DEVICE_ID "Kallakurichi"
#define TOKEN "K12345678"
#define speed 0.034
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/status1/fmt/json";
char topic[] = "iot-2/cmd/home/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
PubSubClient client(server, 1883, wifiClient);
void publishData();
const int trigpin=5;
const int echopin=19;
String command;
String data="";
String latitude="11.7348";
String longitude="78.9639";
String icon="";
long duration;
int dist;
void setup()
{
Serial.begin(115200);
pinMode(trigpin, OUTPUT);
pinMode(echopin, INPUT);
wifiConnect();
mqttConnect();
```

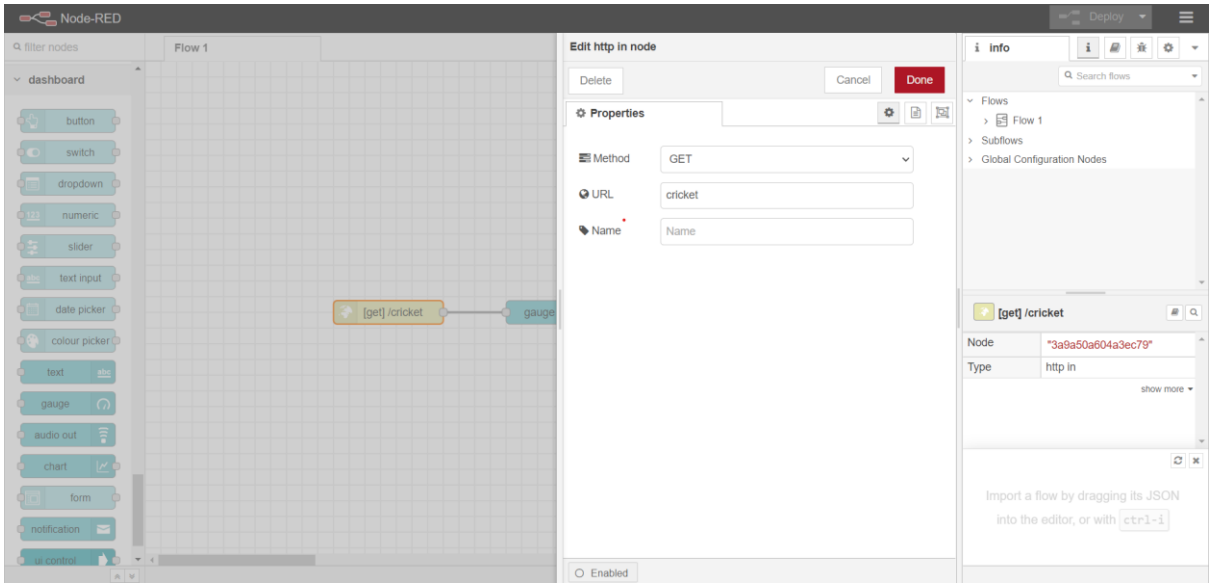


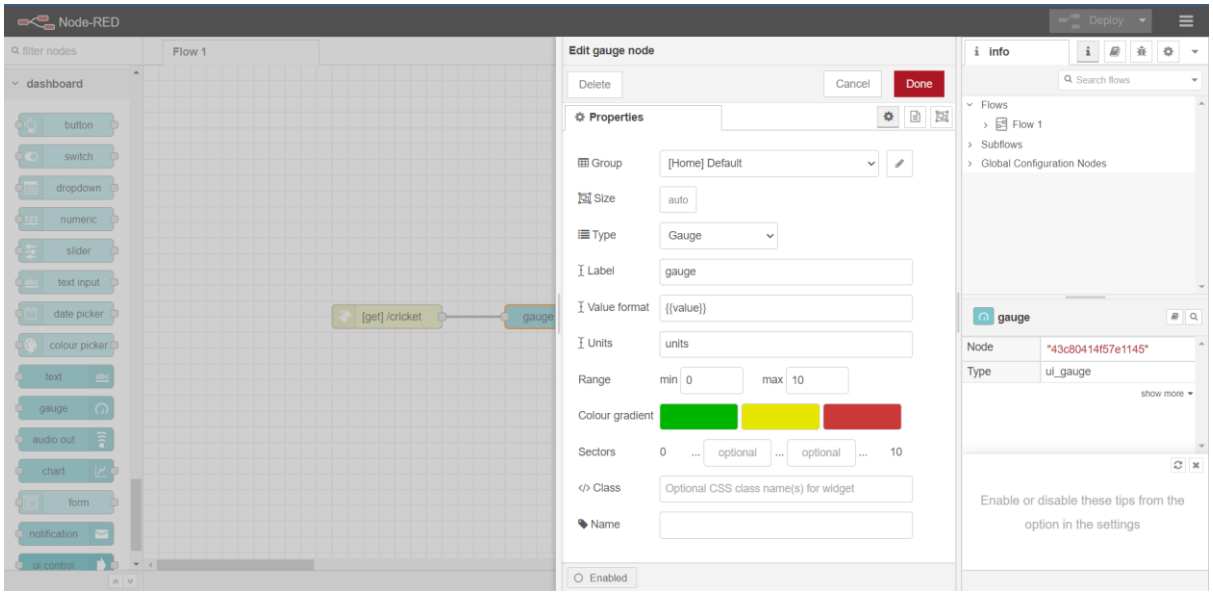
# Final Outptut:

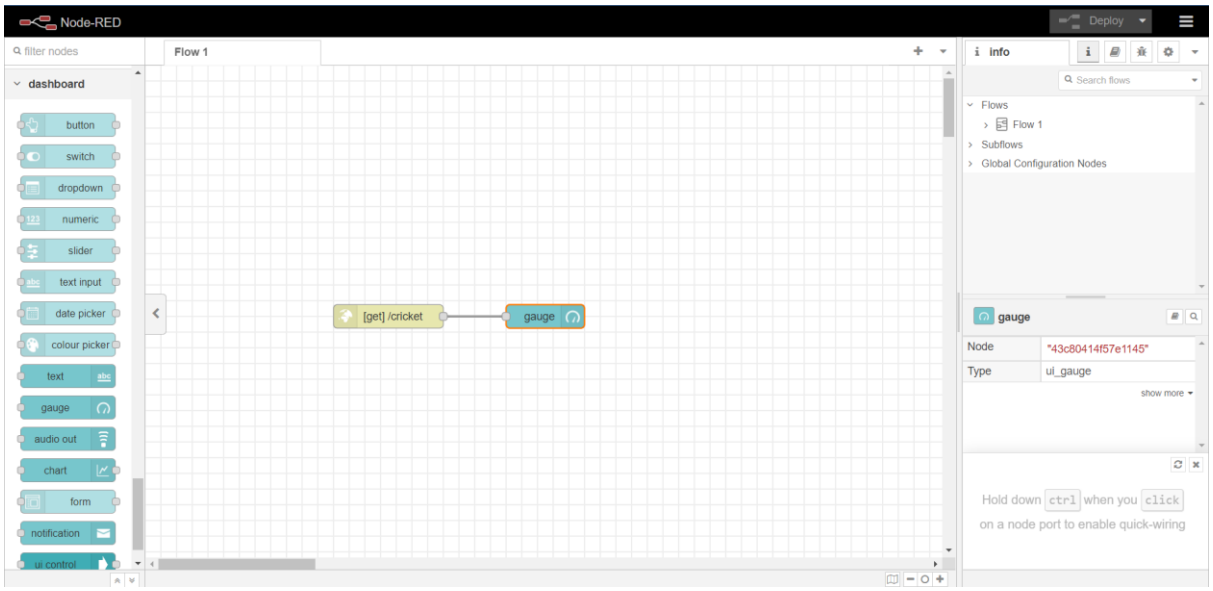


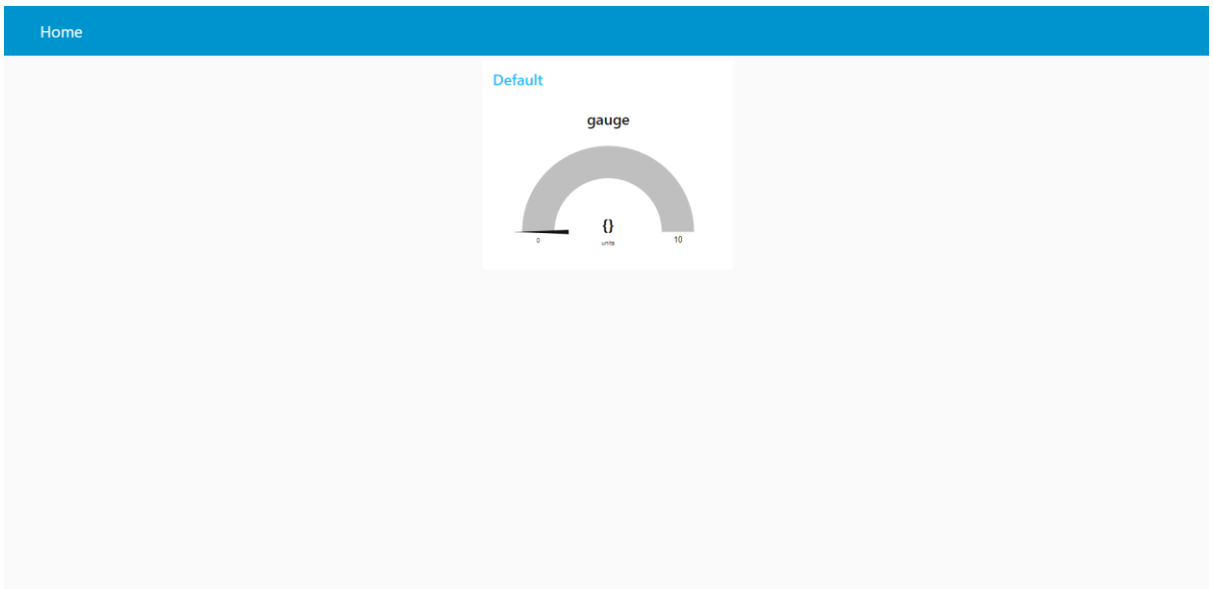

# Sprint 3:

## Dashboard Creation

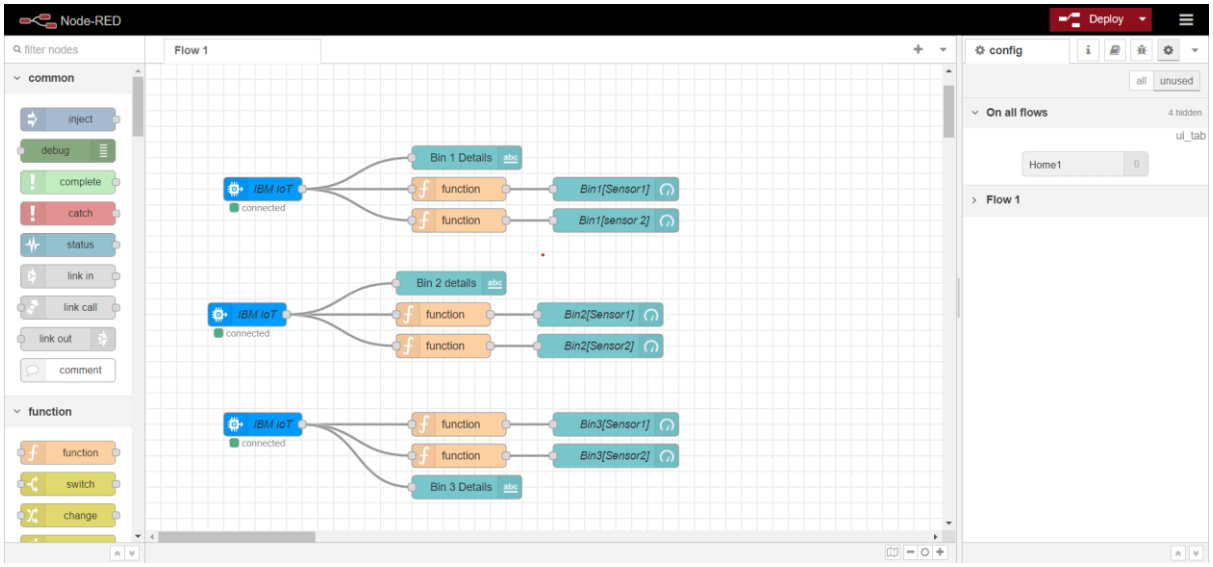


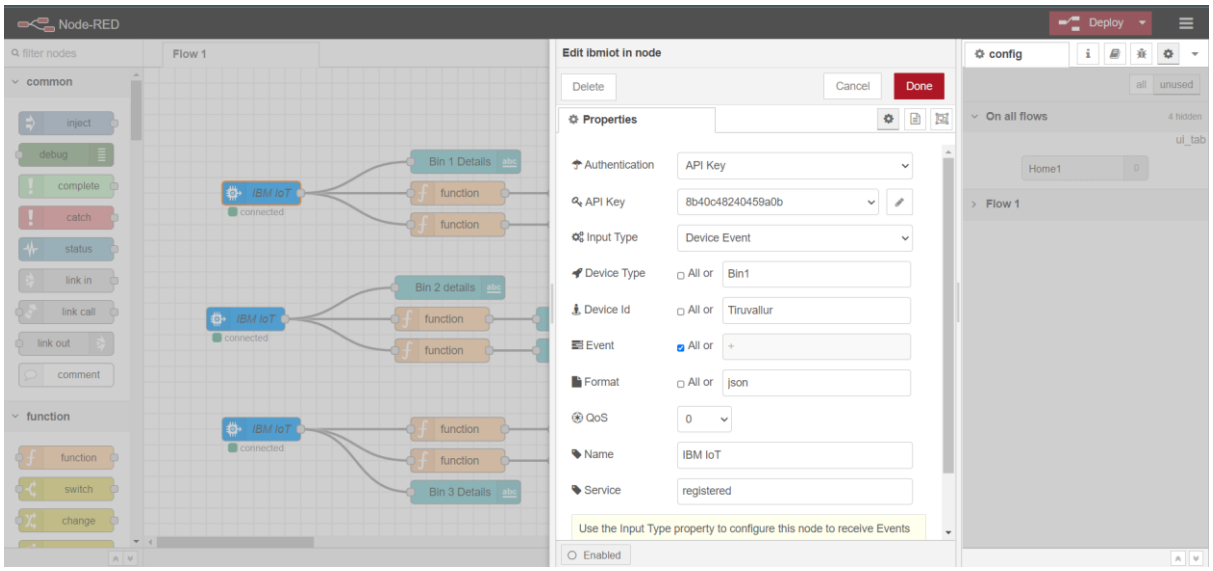


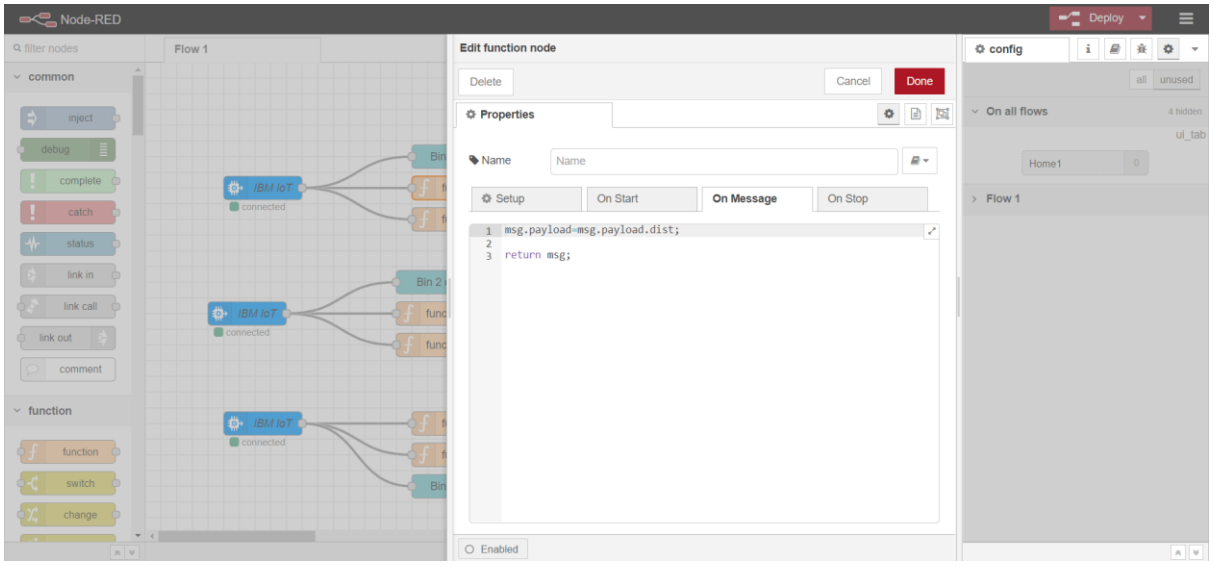



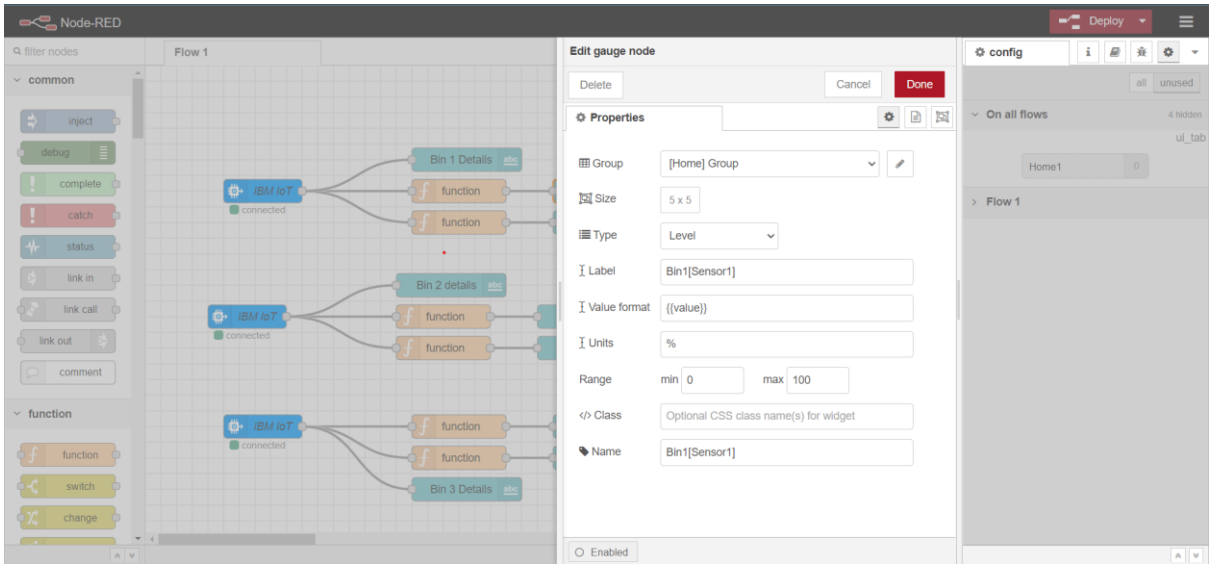




Project Screenshorts


Node-RED
Deploy
filter nodes
Flow 1
common
inject
debug
complete
catch
status
link in
link call
link out
comment
function
function
switch
change
IBM IoT
connected
Bin 1 Details
function
function
Bin1[Sensor1]
Bin1[sensor 2]
Bin 2 details
Bin2[Sensor1]
Bin2[Sensor2]
Bin3[Sensor1]
Bin3[Sensor2]
Bin 3 Details
config
all
unused
On all flows
4 hidden
ui_tab
Home1
Flow 1


Node-RED
Deploy
Edit ibmiot in node
Delete
Cancel
Done
Properties
Authentication
API Key
API Key
8b40c48240459a0b
Input Type
Device Event
Device Type
All or
Bin1
Device Id
All or
Tiruvallur
Event
All or
Format
All or
json
QoS
0
Name
IBM IoT
Service
registered
Use the Input Type property to configure this node to receive Events
Enabled


Node-RED
Deploy
Edit function node
Delete
Cancel
Done
Properties
Name
Name
Setup
On Start
On Message
On Stop
msg.payload=msg.payload.dist;
return msg;
Enabled




## Sprint 4

# Sprint Delivery Schedule:

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

Use the below template to create product backlog and sprint schedule

| Sprint | Functional Requirement (Epic) | User Story Number | User Story / Task | Story |
|---|---|---|---|---|
| Sprint-1 | Registration | USN-1 | As a user, I can register for the application by entering my email, password, and confirming my password. | |
| Sprint-1 | | USN-2 | As a user, I will receive confirmation email once I have registered for the application | |
| Sprint-2 | | USN-3 | As a user, I can register for the application through Facebook | |
| Sprint-1 | | USN-4 | As a user, I can register for the application through Gmail | |
| Sprint-1 | Login | USN-5 | As a user, I can log into the application by entering email & password | |
| | Dashboard | | | |
| | | | | |
| | | | | |

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

| Sprint | Total Story Points | Duration | Sprint Start Date | Sprint End Date (Planned) | Story Points Completed (as on Planned End Date) | Sprint Release Date (Actual) |
|---|---|---|---|---|---|---|
| Sprint-1 | 20 | 6 Days | 24 Oct 2022 | 29 Oct 2022 | 20 | 29 Oct 2022 |
| Sprint-2 | 20 | 6 Days | 31 Oct 2022 | 05 Nov 2022 | 18 | 31 Oct 2022 |
| Sprint-3 | 20 | 6 Days | 07 Nov 2022 | 12 Nov 2022 | 20 | 10 Nov 2022 |
| Sprint-4 | 20 | 6 Days | 14 Nov 2022 | 19 Nov 2022 | 20 | 19 Nov 2022 |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |

**Velocity:**
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

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

**Burndown Chart:**

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.

**https://www.visual-paradigm.com/scrum/scrum-burndown-chart/**

**https://www.atlassian.com/agile/tutorials/burndown-charts**

**Reference:**

https://www.atlassian.com/agile/project-management

https://www.atlassian.com/agile/tutorials/how-to-do-scrum-with-jira-software

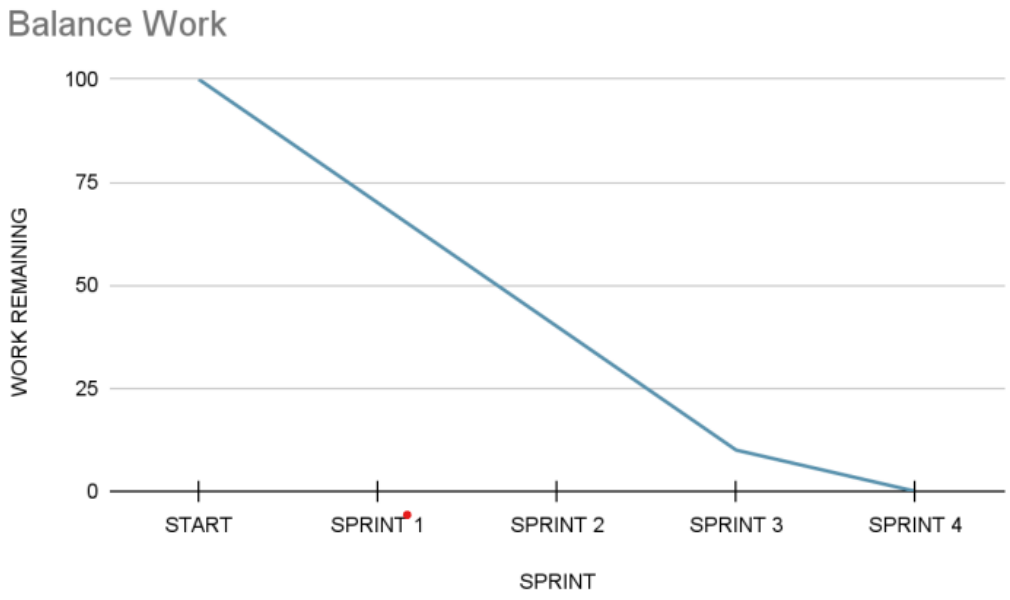
https://www.atlassian.com/agile/tutorials/epics

https://www.atlassian.com/agile/tutorials/sprints

https://www.atlassian.com/agile/project-management/estimation

https://www.atlassian.com/agile/tutorials/burndown-charts

**Burndown Chart:**



## 9. ADVANTAGES & DISADVANTAGES

### ADVANTAGES:

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

### DISADVANTAGES:

- When compared to other techniques, the system requires more trash cans for separate waste collection due to the city's population, which results in a higher initial cost.
- The memory size of the sensor nodes utilised in the trash cans is constrained.

## 10.CONCLUSION

- By employing sensors to track the filling of bins, a Smart Waste Management system that is more effective than the one now in use can be created.
- Our idea of a "smart waste management system" focuses on tracking waste management, providing intelligent technology for waste systems, doing away with human intervention, reducing human time and effort, and creating a clean, healthy environment.
- In smart cities where citizens have hectic schedules that provide little time for garbage management, the suggested solution can be put into practise.
- If desired, the bins might be placed in a city where a big enough container could carry enough solid waste for one unit. The cost could be high.

## 11.FUTURE SCOPE

The proposed system has to be improved in a number of ways, including the following:

- Modify the user authentication and atomic bin locking systems to help safeguard the bins against theft and damage.
- The idea of green points will promote resident or end-user participation, making the concept successful and assisting in the accomplishment of cooperative waste management activities, thus realising the idea of Swachh Bharath.
- Having a case study or data analytics on the types of waste that are collected and when they are collected on different days or seasons, removing the reliance on electrical components and making bin filling predictable, as well as setting the coordinates.Improving the graphical user interfaces of Android and the Server.