PROJECT REPORT FORMAT

1. **INTRODUCTION**
    1. Project Overview
    2. Purpose
2. **LITERATURE SURVEY**
    1. Existing problem
    2. References
    3. Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
    1. Empathy Map Canvas
    2. Ideation & Brainstorming
    3. Proposed Solution
    4. Problem Solution fit
4. **REQUIREMENT ANALYSIS**
    1. Functional requirement
    2. Non-Functional requirements
5. **PROJECT DESIGN**
    1. Data Flow Diagrams
    2. Solution & Technical Architecture
    3. User Stories
6. **PROJECT PLANNING & SCHEDULING**
    1. Sprint Planning & Estimation
    2. Sprint Delivery Schedule
    3. Reports from JIRA
7. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
    1. Feature 1
    2. Feature 2
    3. Database Schema (if Applicable)
8. **TESTING**
    1. Test Cases
    2. User Acceptance Testing
9. **RESULTS**
    1. Performance Metrics

PROJECT REPORT

Team ID	PNT2022TMID46424
Project Name	Smart waste management system for metropolitan cities

INTRODUCTION

Project Overview:

With the increasing population and industrialization of nations throughout the globe, waste has become a great concern for all of us. Over years, researchers figured that only waste management is not enough for its proper treatment and disposal techniques to preserve our environment and keeping it clean in this era of globalization. With the help of technology researchers have, introduced IoT based Smart Waste Management solutions and initiatives that ensures reduced amount of time and energy required to provide waste management services and reduce the amount of waste generated. Unfortunately, developing countries are not being able to implement those existing solutions due to many factors like socio-economic environment. Therefore, in this research we have concentrated our thought on developing a smart IoT based waste management system for developing countries like INDIA that will ensure proper disposal, collection, transportation and recycling of household waste with the minimum amount of resources being available

Purpose:

We amalgamate technology along with waste management in order to effectively create a safe and a hygienic environment. Smart waste management is about using technology and data to create a more efficient waste industry. Based on IoT (Internet of Things) technology, smart waste management aims to optimize resource allocation, reduce running costs, and increase the sustainability of waste services. This makes it possible to plan more efficient routes for the trash collectors who empty the bins, but also lowers the chance of any bin being full for over a week. A good level of coordination exists between the garbage collectors and the information supplied via technology. This makes them well aware of the existing garbage level and instigate them whenever the bins reach the threshold level. They are sent with alert messages so that they can collect the garbage on time without littering the surrounding area. The fill patterns of specific containers can be identified by historical data and managed accordingly in the long term. In addition to hardware solutions, mobile applications are used to overcome the challenges in the regular waste management system, such as keeping track of the drivers while they are operating on the field. Thus, smart waste management provides us with the most optimal way of managing the waste in an efficient manner using technology.

LITERATURE SURVEY:

Existing problem:

Waste management has become an alarming challenge in local towns and cities across the world. Often the local area bins are overflowing and the municipalities are

not aware of it. This affects the residents of that particular area in numerous ways starting from bad door to unhygienic and unsafe surroundings. Poor waste management - ranging from non-existing collection systems to ineffective disposal - causes air pollution, water and soil contamination. Open and unsanitary areas contribute to contamination of drinking water and can cause infection and transmit diseases. Toxic components such as Persistent Organic Pollutants (POPs) pose particularly significant risks to human health and the environment as they accumulate through the food chain. Animals eating contaminated plants have higher doses of contaminants than if they were directly exposed. Precipitation or surface water seeping through waste will absorb hazardous components from landfills, agricultural areas, feedlots, etc. and carry them into surface and groundwater. Contaminated groundwater also poses a great health risk, as it is often used for drinking, bathing and recreation, as well as in agricultural and industrial activities. Landfills and waste transfer stations can attract various pests (insects, rodents, gulls, etc.) that look for food from waste. These pests can spread diseases through viruses and bacteria (i.e., salmonella and e- coli), which are a risk to human health.

REFERENCES:

S. NO	TITLE OF THE PAPER	AUTHOR NAME	YEAR OF PUBLICATION	REMARKS	OUTPUT
1	IOT based smart waste bin monitoring and municipal solid waste management system for smart cities	Muhammed irfan, Abdullah saeed, Al wadie , adam	4-June/2020	Environmental Pollution. Improper collector and disposal mechanism	Collect the waste effectively. Detection of fire in waste material. Wirelessly connected with the central hub Of transmit the info about the bins filling level with existing collection. Avoid the overflow of bins.
2	Smart garbage segregator and IOT based waste collector system	Mrigank goel, Amogh harsh goyal, Preeti dhiman, Vikas deep, Purshottam sharma	05-March/2021	All wet waste are not used for urban agriculture, organic farming.	It segregates the metallic dry and wet waste. It also convert that it can be further used in urban agriculture, organic farming. It alert the waste management Center through IOT system

					whenever any of the metallic or dry garbage Bins is full to avoid serious environment hazards
3	A novel strategy for waste prediction using machine Learning algorithm with IOT based intelligent waste management system	G.Uganya, D.Rajalakshmi, Arun Radhakrishnan Ramya , Yuvaraja teeka, -raman	10-Feb/2022	Low cost Method High accuracy Complicated method Because of using machine learning algorithm	Automatic method, predicting the possibility of waste things. The waste capacity ,gas level, metal level monitored continuously Using IOT based dustbins. Tested by random forest algorithm gives the accuracy of 92.15% and give time consumptions of 0.2 ms.
4	System waste management	Arafat ali khan Farhana shetu Saimum bari Lawshik shikder	7-Jan/2021	Good enough to prevent the garbage overflow and ensures the partial is perfect waste management and monitoring system	Microcontroll er, sensor, GSM are used in the system. This proposed system would have an automated waste level detection process and also a smart monitoring and overall management process.
5	Real time smart garbage bin mechanism	Dominic Abuga N.S.Ragava	23-Oct/2021	Fuzzy logic is applied Hence real	This mechanism proposed

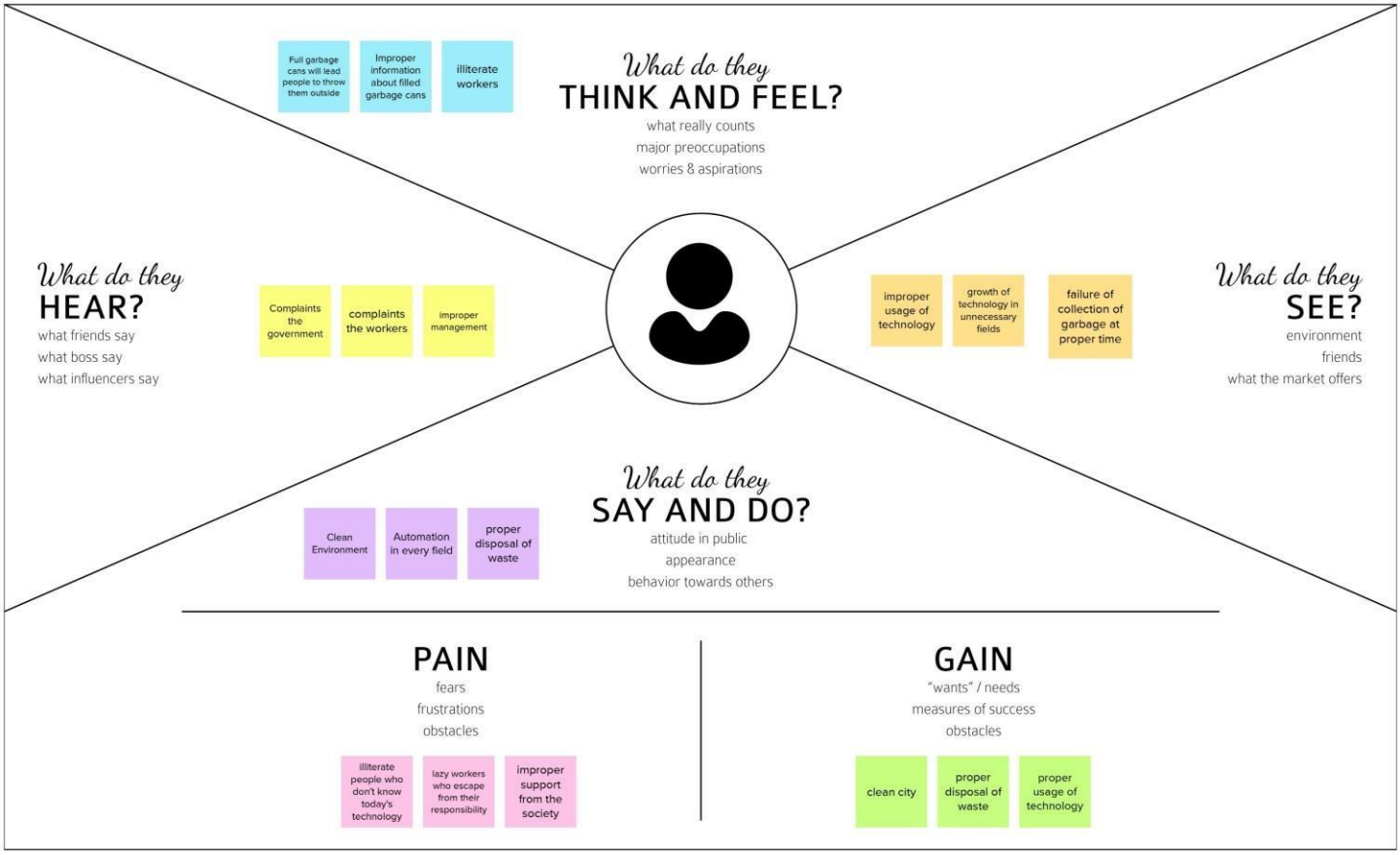
	for solid waste management in smart cities			time decision making avoid real time monitoring	accesses real time information of any smart garbage bin deployed across the city and helps to resolve the problem of waste overflow from garbage bins and keeps cities clean
6	Smart waste management system using IOT	V.Pavan sankeeth V.Bhavana V.Santhosh Markandeya	3-Nov/2019	Easy process but garbage truck driven must have cell phones	The server monitors garbage bins that are spread across the city at multiple location sever sends SMS to assigned mobile number which provide route to the driver based on all the data collected from bins
7	Automated waste garbage monitoring system with optimal route generation for collection	Aarati medehal Aniruddha Annaiuru Shalini T.S.Chander	1-Oct/2020	Using of machine learning algorithm improve the redundant inefficient	This purpose of this system is use powerful tools of IOT for completely automated the process of garbage monitoring using ultrasonic sensor and node MCU And provide an optimal route for garbage

					collection using cluster first route several ml algorithm
8	An IOT based garbage monitoring a disposal support system.	T.M.N Vamsi G.Kalyan chakravathi B.Divakar Protibha	10-April/2021	Consumption of fuel in garbage truck is reduced but clean personal must have smart phone to active this process	This system monitors garbage bins located at different locations and notifies about the level of garbage accumulated in garbage bins through android application to cleaning personnel and provides shortest path to the garbage bin location
9	IOT based solar powered smart waste management system with real time monitoring an advancement for smart city planning	Md.humaun Kabir,sujit roy, Md.tofail ahmed, Mahmudul alam	21-Oct/2020	Project costs complicated but this can be suitable for any kind of cities or town and ensures proper collection and disposal of garbage	It enables real time monitoring of solar powered several smart bins located in different point in the city which are connected to control system through long range (LDRA) Communicati on device and also supervises the waste collection and disposal time using automated

					vehicles locating system
10	Intelligent sensor based waste disposal system for smart cities	Chinmai shetty B.Dhenanyaya Deepa N.Rashmi	1-Nov/2020	Intelligent sensor makes a person to through garbage the bins	Sensor helps is identify the quality of garbage and real collected information the info is send to the drivers and the garbage collected information is updated on the web page this system also ensure that there is no waste thrown around the trash bin the intelligent sender should make beeping sound if any person through around the trash bin and not into the trash bin

EMPATHY MAP

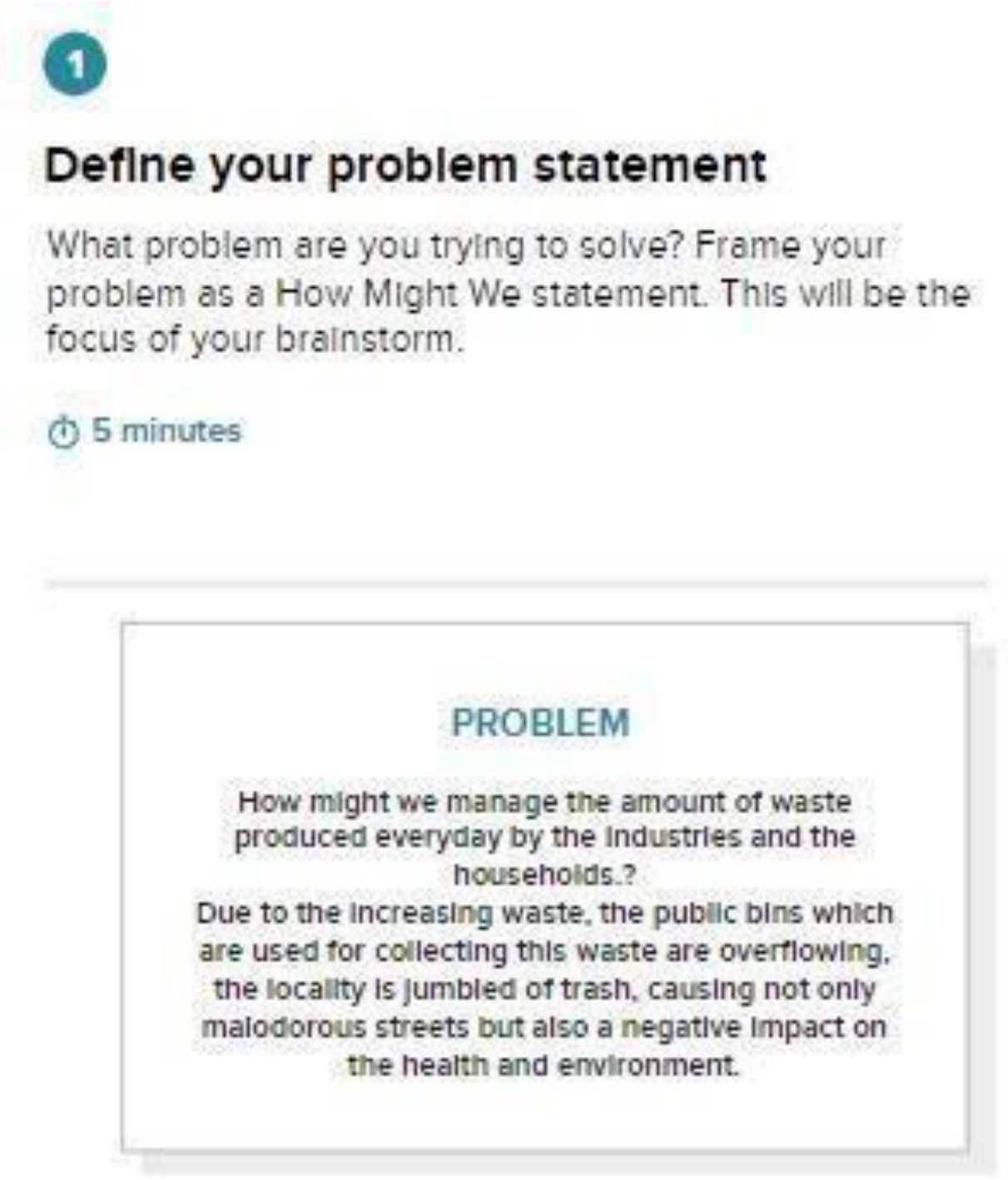
SMART WASTE MANAGEMENT SYSTEM FOR METROPOLITAN CITIES



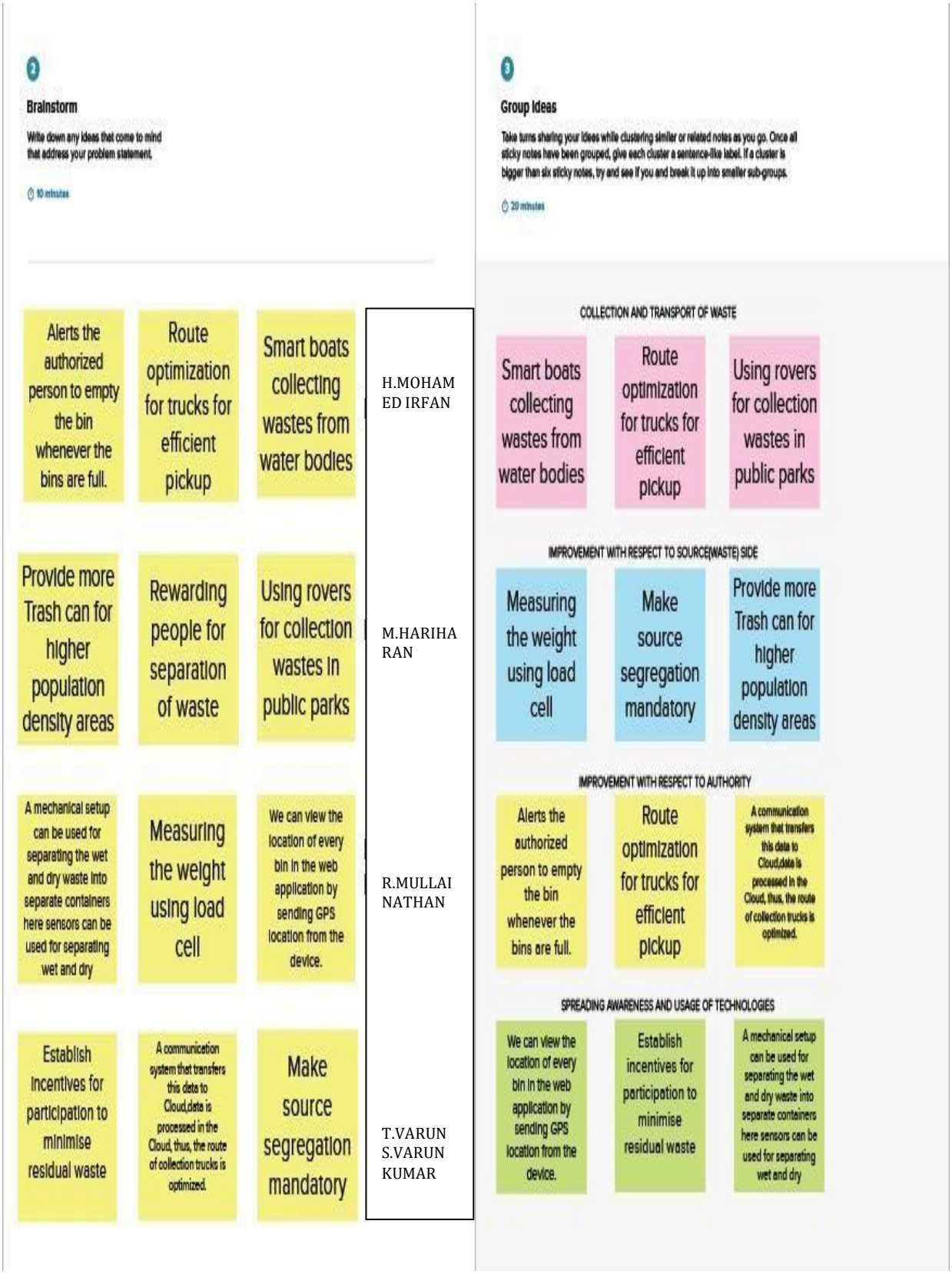
IDEATION PHASE
DEFINE THE PROBLEM STATEMENTS

IDEATION PHASE
BRAINSTORMING AND IDEA GENERATION

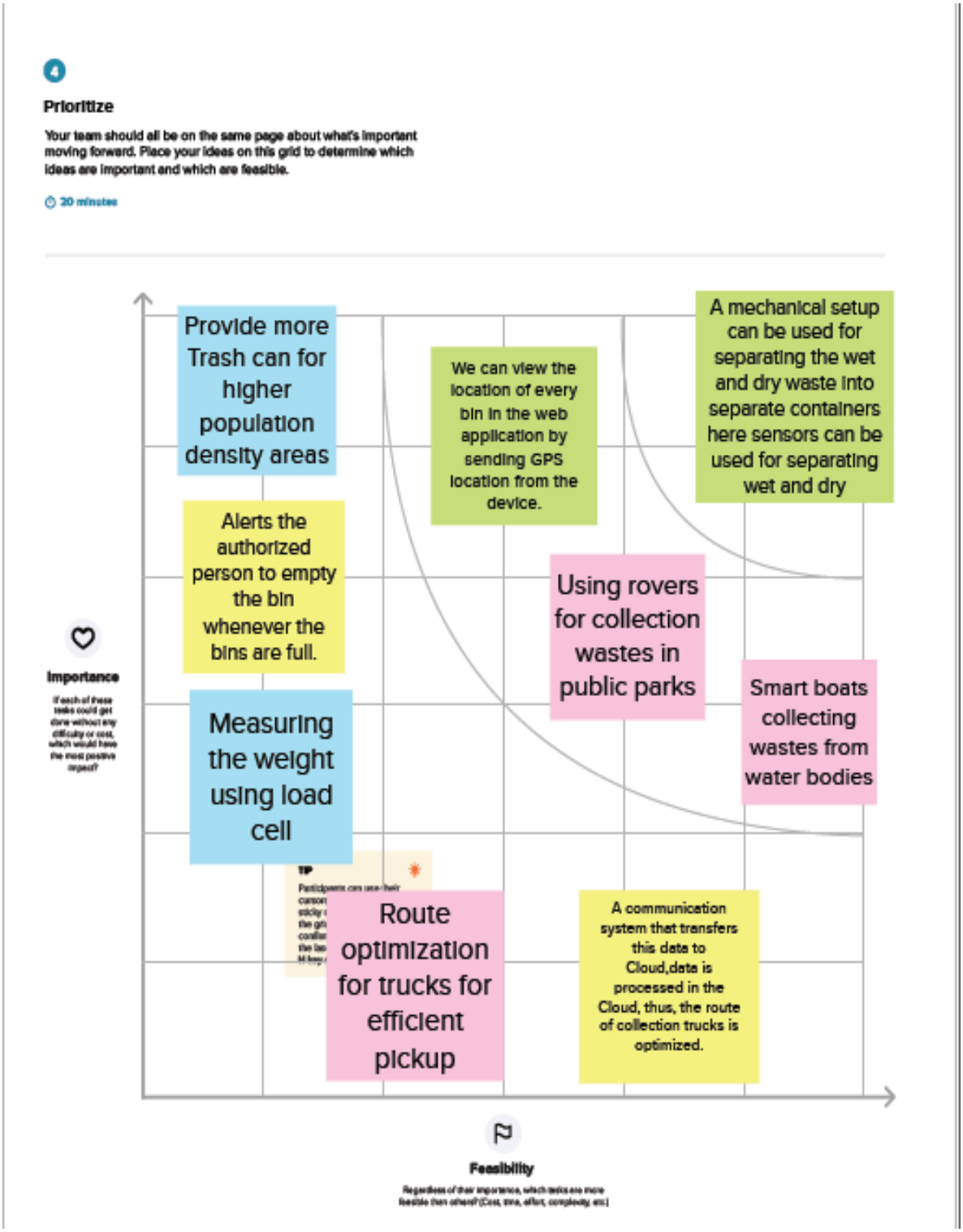
Step-1: Team Gathering, Collaboration and Select the Problem Statement



Step-2: Brainstorm, Idea Listing and Grouping



Step-3: Idea Prioritization

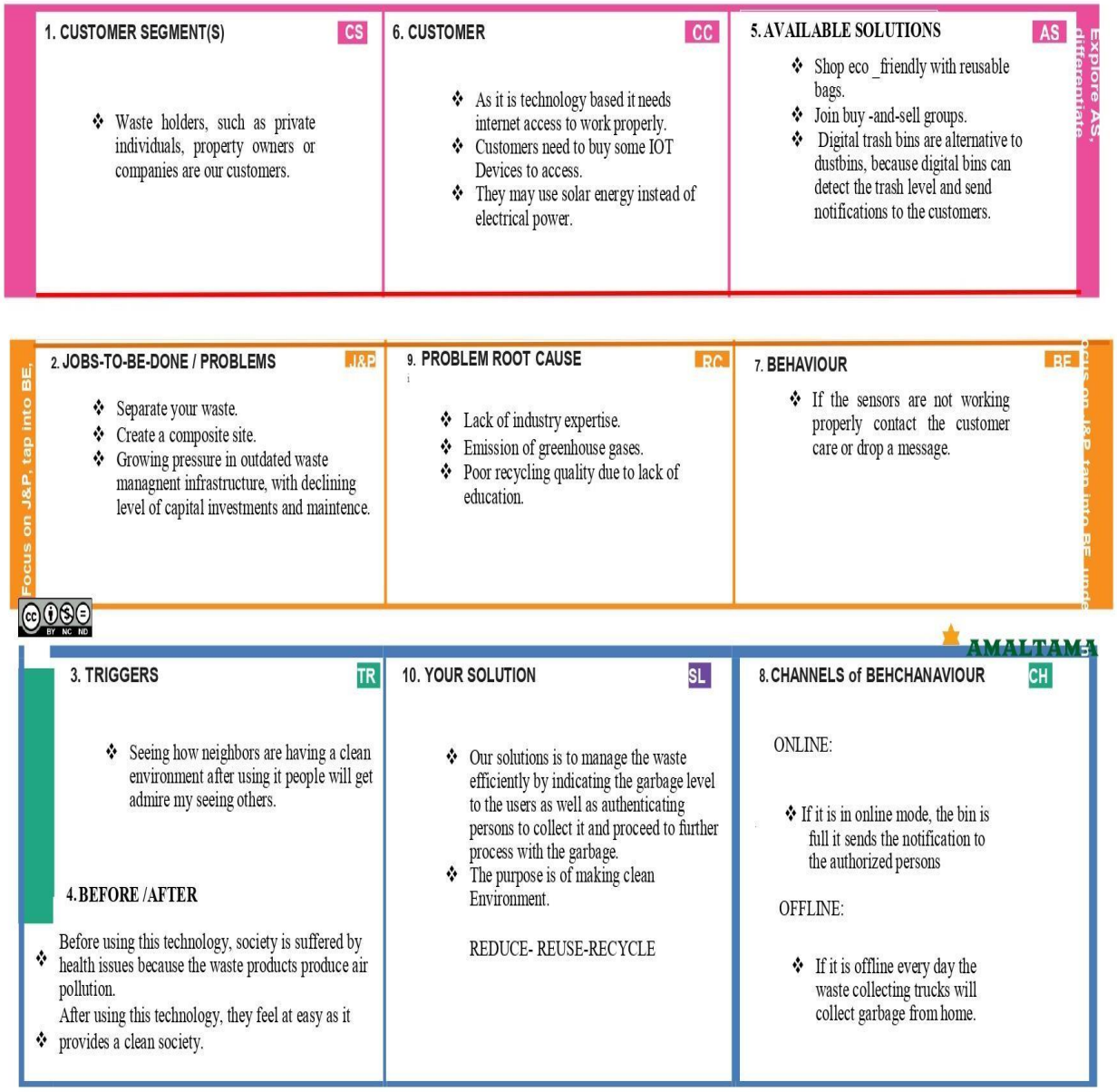


PROJECT DESIGN PHASE-I
PROPOSED SOLUTION

S.NO.	PARAMETER	DESCRIPTION
1.	Problem Statement (Problem to be solved)	This project deals with the problem of waste management in smart cities, where the garbage collection system is not optimized. This project enables the organizations to meet their needs of smart garbage management systems. This system allows the authorized person to know the fill level of each garbage bin in a locality or city at all times, to give a cost-effective and Time-saving route to the truck drivers.
2.	Idea / Solution description	The key research objectives are as follows: • The proposed system would be able to automate the solid waste monitoring process and management of the overall collection process using IOT (Internet of Things). • The Proposed system consists of main subsystems namely Smart Trash System (STS) and Smart Monitoring and Controlling Hut (SMCH). • In the proposed system, whenever the waste bin gets filled this is acknowledged by placing the circuit at the waste bin, which transmits it to the receiver at the desired place in the areaor spot. • In the proposed system, the received signal indicates the waste bin status at the monitoringand controlling system.
3.	Novelty / Uniqueness	We are going to establish SWM in our college but the real hard thing is that janitor (cleaner) don't know to operate these thing practically so here our team planned to build a wrist band to them, that indicate via light blinking when the dustbin fill and this is Uniqueness we made here beside from project constrain.
4.	Social Impact / Customer Satisfaction	From the public perception as worst impacts of present solid waste disposal practices are seen direct social impacts such as neighborhood of landfills to communities, breeding of pests andloss in property values

5.	Business Model (Revenue Model)	Waste Management organizes its operations into two reportable business segments: Solid Waste, comprising the Company's waste collection, transfer, recycling and resource recovery, and disposal services, which are operated and managed locally by the Company's various subsidiaries, which focus ondistinct geographic areas; and Corporate and Other, comprising the Company's other activities, including its development and operation of landfill gas-to- energy facilities in the INDIA, and its recycling brokerage services, as well as various corporatefunctions.
6.	Scalability of the Solution	In this regard, smart city design has been increasingly studied and discussed around the world to solve this problem. Following this approach, this paper presented an efficient IoT- based and real-time waste management modelfor improving the living environment in cities, focused on a citizen perspective. The proposed system uses sensor and communication technologies where waste data is collected from the smart bin, in real-time, and then transmitted to an online platform where citizens can access and check the availability of the compartments scattered around a city.

PROJECT DESIGN PHASE-I
PROBLEM – SOLUTION FIT



PROJECT DESIGN PHASE-II
SOLUTION REQUIREMENTS
(FUNCTIONAL & NON-FUNCTIONAL)

Functional Requirements:

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Fitting IoT device in the trashcans.	The IoT device need to be fixed in the dustbin with water proof safety. The IoT device consists Ultrasonic sensor, IR sensor, Weight sensor. To send data to the cloud GSM/GPRS is used.
FR-2	Connecting to the cloud.	The device should configure to connect to the cloud. The data of sensors need to be received and processed.
FR-3	Predictions for bin fulness.	In this system, a 24×7 monitoring system is designed for monitoring dumpsters, A smart and organized system is designed for selective clearing the ultrasonic sensor is used for measuring the level of waste in the dustbin, DC motor powered platform is used for segregating wet and dry waste, IR sensor and moisture sensor is used for separating wet and dry waste. If either of the containers is full then an alert message is sent from the dustbin to employees and the cloud. In turn, employees can clear the corresponding dumpster.
FR-4	Real-time waste monitoring	Trash and recycling containers can be outfitted or produced with low-cost sensors that monitor everything from the amount and types of material in a container to temperature, odour and location of the bin.
FR-5	Do not miss a pick	For periodically picked bins, we provide Pick evaluation. The tool records picks (sensor) and compares them to the schedule. Authorized person can immediately identify any missed, or off-schedule picks.
FR-6	Routes to the dumpsters	Based on current bin fill-levels and predictions of reaching full capacity, you are ready to respond and schedule waste collection. driver can compare planned vs. executed routes to identify any inconsistencies.

Non-functional Requirements:

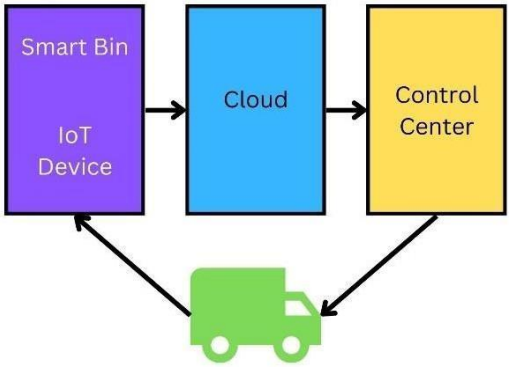
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	IoT solutions for waste management problems offer municipalities data intelligence and real-time insights. In that regard, the fill patterns of specific containers can be identified by historical data and managed accordingly in the long term. In addition to hardware solutions, mobile applications are used to overcome the challenges in the regular waste management system, such as keeping track of the drivers while they are operating on the field.
NFR-2	**Security**	Building and deploying IoT-based smart waste management in cities can be a complex, time-consuming and resource-intensive process. Many municipal IT departments will not have the resources or in-house skills to support such a project internally.
NFR-3	**Reliability**	One of the difficult operational problems of municipal and local authorities are facing is the collection of municipal solid waste. In recent years, due to environmental concerns and number of costs, most of the municipalities have been forced for assessing their solid waste management and examining their cost-effectiveness and environmental impact, for example, designing the collection of routes. During the past 15 years
NFR-4	**Performance**	An integrated Arduino program is developed to synchronize the identification system, automated lid system, micro-controller, display system, and communication system. An ultrasonic sensor is attached to the front side of the garbage bin. The transmitter of the ultrasonic sensor emits an ultrasonic sound that is beyond the human ear listening range, and the receiver receives the reflected sound waves by the solid objects.
NFR-5	**Availability**	Another purpose of this project is to make the proposed waste management system as cheap as possible. A cost in BDT is presented in the following Table 3 needs for the construction of the proposed smart bin.
NFR-6	**Scalability**	The city diverts about 80% of its waste from landfills and hopes to go “zero waste” by the end of 2020. Besides strict regulations and high waste management fees for end consumers and businesses.

PROJECT DESIGN PHASE-II

DATA FLOW DIAGRAM & USER STORIES

Date	15 October 2022
Team ID	PNT2022TMID46424
Project Name	Project - Smart Waste Management System For Metropolitan Cities
Maximum Marks	4 Marks

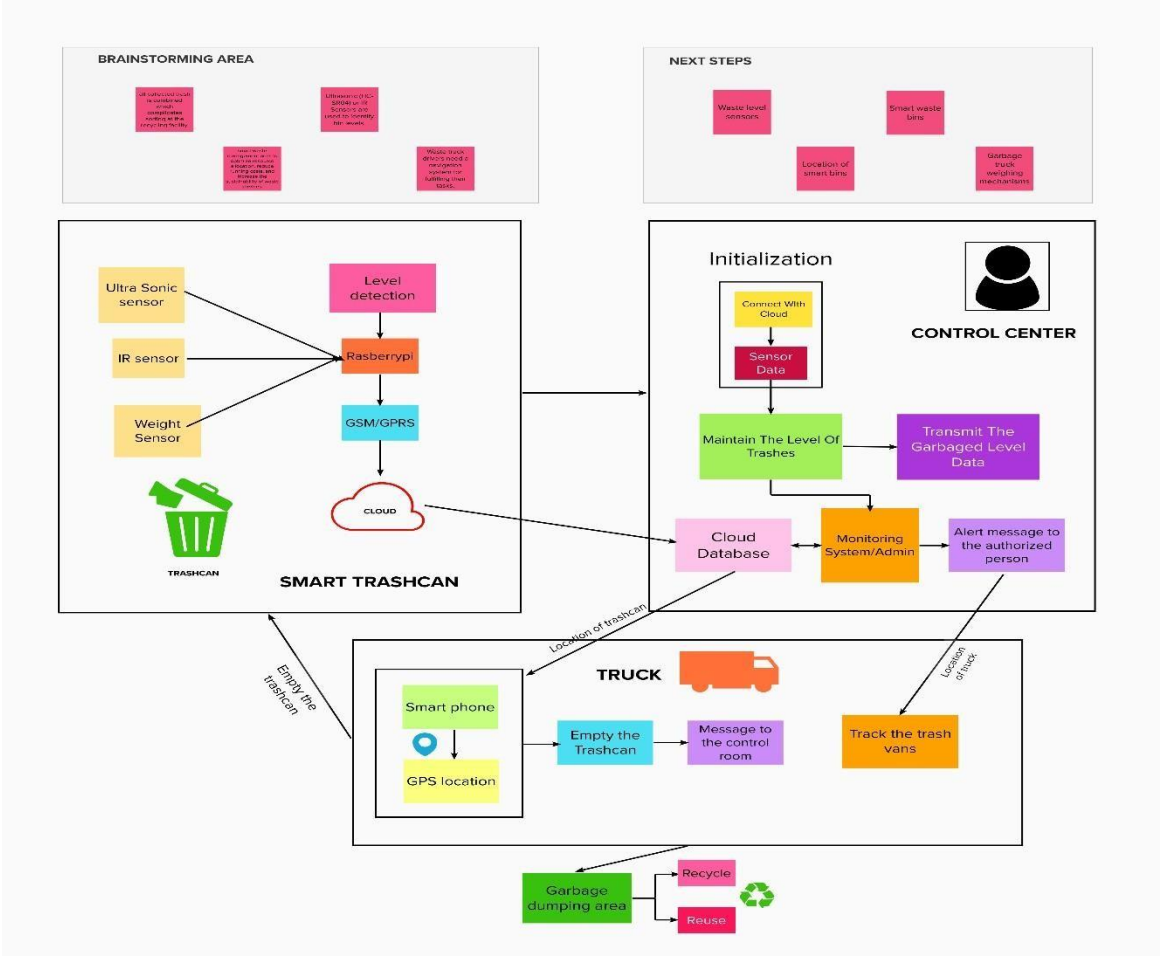
Data Flow Diagram:



Simplified Diagram

The IoT device is fitted in the trashcans.

- The sensors in the device senses the garbage level.
- The GSM/GPRS will send the information about the garbage level to the cloud.
- The admin in the control center notifies the authorized person to collect the garbage.



- The truck driver will be notified the route to the filled dumpsters.
- The trashes are loaded to the truck.
- The more number of bins needed in high populated area.
- The overflowing of trashcans can be avoided.
- No missed pickups of trashcans.
- New smart dustbins can be install by just connecting the IoT device to the cloud.

User Stories:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Admin	Login	USN-1	As an admin, I can monitor every dustbin and its garbage levels.	I can monitor the system.	High	Sprint-4
		USN-2	As an admin, I will inform the authorized person to empty the trashcan.	I can inform authorized person.	Medium	Sprint-2
		USN-3	As an admin, I can notice the trash level of every dustbin.	I can notice the trash level.	Low	Sprint-2
Admin 2	Login	USN-4	As a Co-Admin, I can send alert message to the truck drivers.	I can alert truck driver.	High	Sprint-1
Trash Van Driver	Login	USN-5	As a trash van driver, I will follow the route to the dustbin.	I can reach the filled trashcans.	High	Sprint-2
Garbage Collector		USN-6	As a waste collector, I will collect all the trash from the dumpsters and load it to the truck.	I can empty the trashcans.	Medium	Sprint-2

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Municipal officer	Login	USN-7	As a municipality officer, I can supervise the process and ensure the cleanliness of city.	I can manage all these process going good.	High	Sprint-1
Trashcan Monitor	Register	USN-8	As a trashcan monitor, I can initialize new trashcans.	I can register new smart trashcans.	Medium	Sprint-3
		USN-9	As a trashcan monitor, I can check the quality of IoT device's quality.	I can check the IoT device.	Medium	Sprint-3

PROJECT DESIGN PHASE-II

TECHNOLOGY STACK (ARCHITECTURE & STACK)

Technical Architecture:

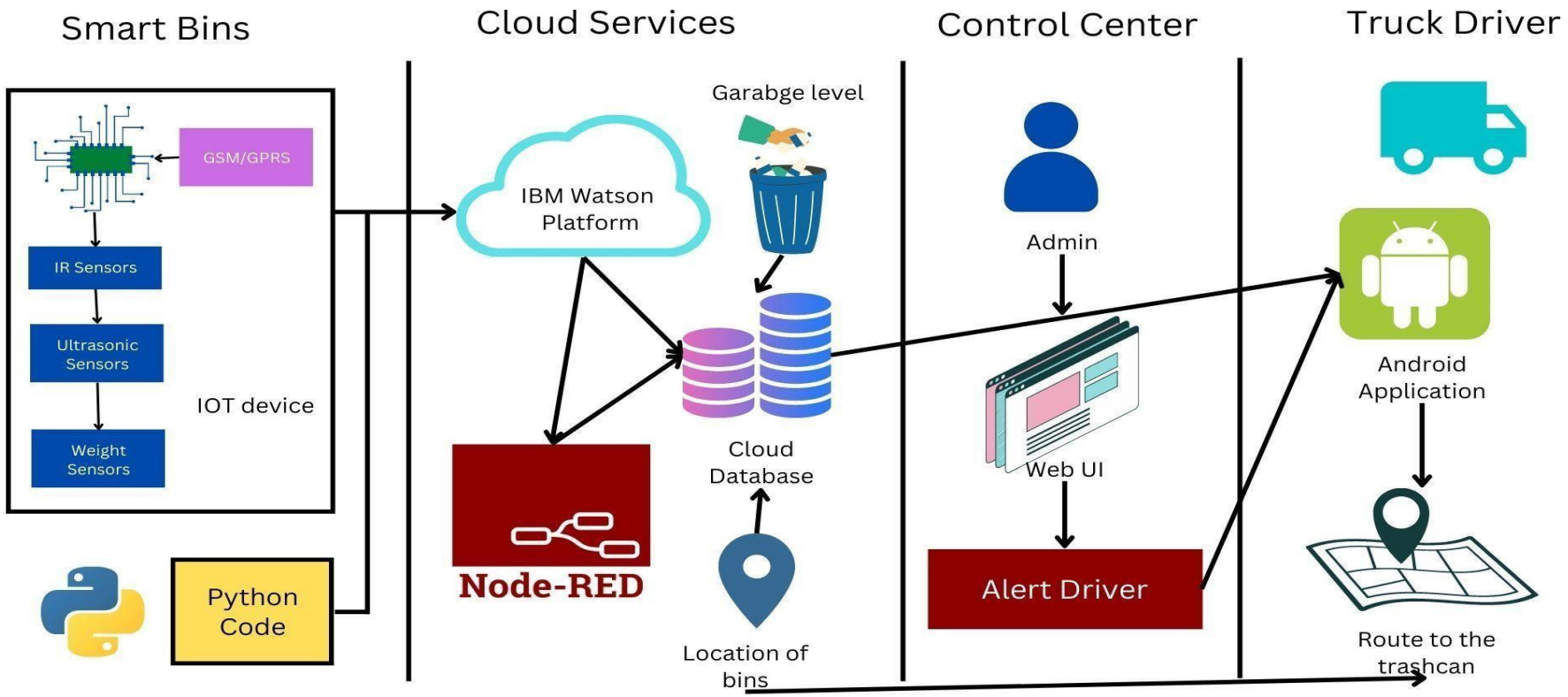


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	Arduino Uno	The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller.	Arduino programming itself is done in **C++.**
2.	Application Logic-1	Logic for IR sensor data.	C++/Python
3.	Application Logic-2	Logic for Ultrasonic sensor data.	C++/Python
4.	Application Logic-3	Logic for a Weight sensor data	C++/Python
5.	GPRS/GSM	The Arduino GSM shield allows an Arduino board to connect to the internet, send and receive SMS, and make voice calls using the GSM library.	C++/Python
6.	Cloud Sever	Application deployment on Local System / Cloud	IBM Watson IoT Platform, Node Red
7.	Cloud Database	Database Service on Cloud	IBM Watson IoT platform, Cloudant DB
8.	User Interface	How user interacts with application to alert the truck driver.	HTML, CSS, JavaScript , Python etc.
9.	External API-1	Purpose of External API used in the application to locate the trashcans.	Google Maps Geolocation API

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Microcontroller	Arduino Uno is used to make the IoT device	C++/Python
2.	Security	Encryption/Decryption used for security purpose	GSM/GPRS,Python
3.	Scalable Architecture	New features can be added.	Node Red
4.	Availability	Web application can be accessed from anywhere	IBM Watson IoT Platform, HTML, CSS, JavaScript
5.	Performance	All truck drivers can access the application at same time.	Cloudant DB, IBM Watson IoT Platform

PROJECT PLANNING PHASE
PROJECT PLANNING TEMPLATE (PRODUCT BACKLOG, SPRINT PLANNING, STORIES, STORY POINTS)

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

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Login	USN-1	As a Administrator, I need to give user id and passcode for ever workers over there in municipality	10	High	Mohamed Irfan
Sprint-1	Login	USN-2	As a Co-Admin, I'll control the waste level by monitoring them vai real time web portal. Once the filling happens, I'll notify trash truck with location of bin with bin ID	10	High	Hariharan
Sprint-2	Dashboard	USN-3	As a Truck Driver, I'll follow Co-Admin's Instruction to reach the filling bin in short roots and save time	20	Low	Varun
Sprint-3	Dashboard	USN-4	As a Local Garbage Collector, I'II gather all the waste from the garbage, load it onto a garbage truck, and deliver it to Landfills	20	Medium	Varun kumar
Sprint-4	Dashboard	USN-5	As a Municipality officer, I'll make sure everything is proceeding as planned and without any problems	20	High	Mullainathan

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	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 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)

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

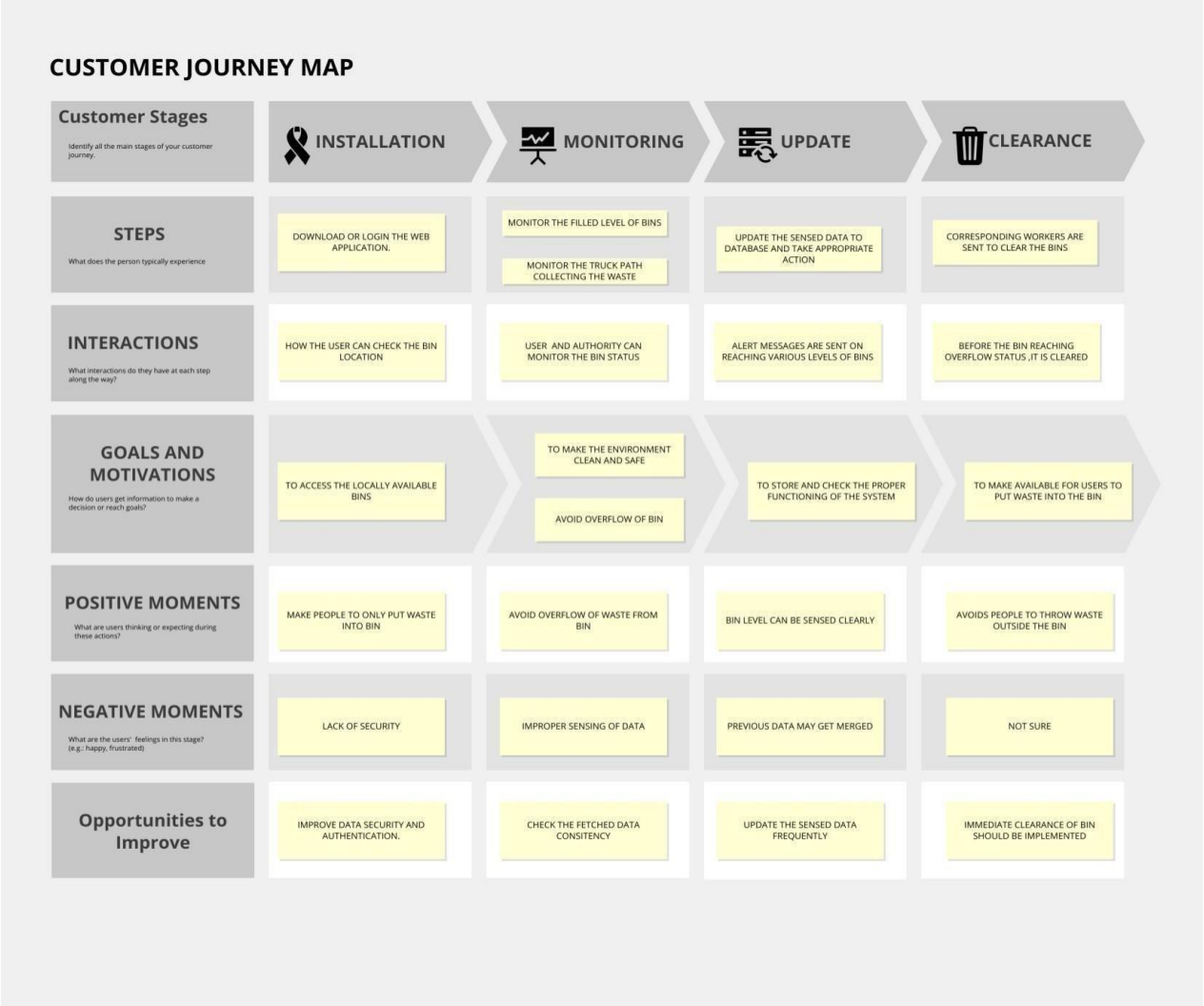
PROJECT PLANNING PHASE
MILESTONE AND ACTIVITY LIST

TITLE	DESCRIPTION	DATE
Literature Survey & Information Gathering	Literature survey on the selected project & gathering information by referring the, technical papers, research publications etc.	28 SEPTEMBER 2022
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem statements	24 SEPTEMBER 2022
Ideation	List the by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.	25 SEPTEMBER 2022
Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.	23 SEPTEMBER 2022
Problem Solution Fit	Prepare problem - solution fit document.	30 SEPTEMBER 2022
Solution Architecture	Prepare solution architecture document.	28 SEPTEMBER 2022

Customer Journey	Prepare the customer journey maps to understand the user interactions & experiences with the application (entry to exit).	20 OCTOBER 2022
Functional Requirement	Prepare the functional requirement document.	8 OCTOBER 2022
Data Flow Diagrams	Draw the data flow diagrams and submit for review.	9 OCTOBER 2022
Technology Architecture	Prepare the technology architecture diagram.	10 OCTOBER 2022
Prepare Milestone & ActivityList	Prepare the milestones & activity list of the project.	22 OCTOBER 2022
Project Development - Delivery of Sprint-1, 2, 3 & 4	Develop & submit the developed code by testing it.	IN PROGRESS..

PROJECT DESIGN PHASE-I
SOLUTION ARCHITECTURE

PROJECT DESIGN PHASE-II
CUSTOMER JOURNEY MAP

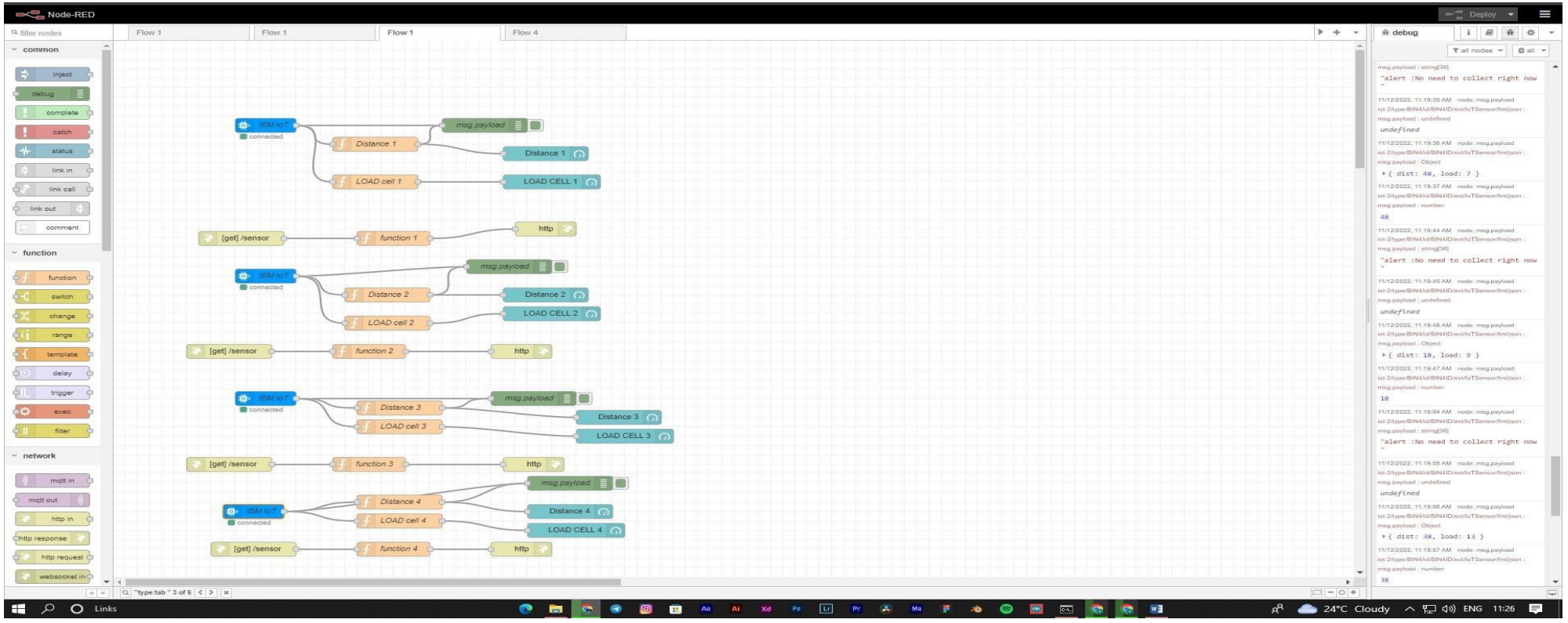


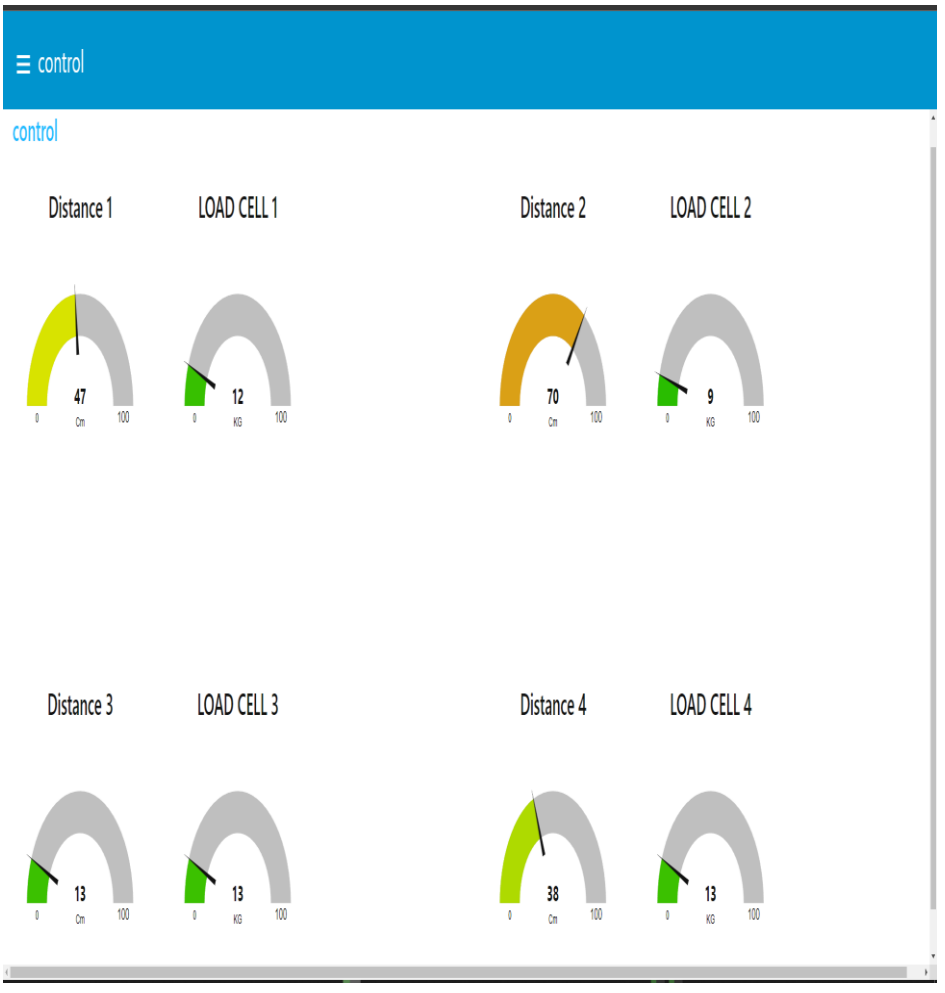
USE DASHBOARD NODES FOR CREATING UI (WEB APPS)

Team ID	PNT2022TMID46424
Project Name	Project - Smart Waste Management System for Metropolitan Cities

Step 1: Open Node red and pick and place blocks according to python scriptflow
Step 2: Make sure necessary blocks are installed in Node Red
Step 3: After creating the flow click on deploy
Step 4: Output is displayed in Node-red Debug window
Step 5: Also, web UI can also be seen by the URL followed by/ui

SCREENSHOTS: **NODE – RED FLOW**




control
control
Distance 1
47
Cm
0
100
LOAD CELL 1
12
KG
0
100
Distance 2
70
Cm
0
100
LOAD CELL 2
9
KG
0
100
Distance 3
13
Cm
0
100
LOAD CELL 3
13
KG
0
100
Distance 4
38
Cm
0
100
LOAD CELL 4
13
KG
0
100

CODING
HTML:

```
<!DOCTYPE html>
<html>

<head>
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.css"
integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1
T" crossorigin="anonymous">
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width">
        <title>Garbage Management System</title>
        <link rel="icon" type="image/x-icon" href="/Images/DUMPSTER.png">
        <link href="style.css" rel="stylesheet" type="text/css" />
        <script src="https://www.gstatic.com/firebasejs/8.10.1/firebase-
app.js"></script>
        <script src="https://www.gstatic.com/firebasejs/8.10.1/firebase-
database.js"></script>

        <script>
                var firebaseConfig =
                {
                        apiKey: "AIzaSyB9ysbnaWc3IyeCioh-aJQT_UCMd5CBFeU",
                        authDomain: "fir-test-923b4.firebaseapp.com",
                        databaseURL: "https://fir-test-923b4-default-rtdb.firebaseio.com",
                        projectId: "fir-test-923b4",
                        storageBucket: "fir-test-923b4.appspot.com",
                        messagingSenderId: "943542145393",
                        appId: "1:943542145393:web:9b5ec7593e6a3cbd7966d0",
                        measurementId: "G-BN7JNX1Q7B"
                };
                firebase.initializeApp(firebaseConfig)
        </script>
        <script defer src="database.js"></script>
</head>

<body style="background-color:#1F1B24;">
        <script src="map.js"></script>


                <div id="map_container">
                        <h1 id="live_location_heading" >LIVE LOCATION</h1>
```

```
                    <div id="map"></div>
                    <div id="alert_msg">ALERT MESSAGE!</div>
   </div>
        </div>
<center><a href="https://goo.gl/maps/G9XET5mzSw1ynHQ18"
  type="button" class="btn btn-dark">DUMPSTER</a></center>

        <script
            src="https://maps.googleapis.com/maps/api/js?key=AIzaSyBBLyWj-
3FWtCbCXGW3ysEiI2fDfrv2v0Q&callback=myMap"></script></div>
</body>

</html>
```

STYLE.CSS

```
html, body
        {
        height: 100%;
            margin: 0px;
            padding:0px;
        }
#container
{
        display: flex;
        flex-direction: row;
        height: 100%;
        width: 100%;
        position: relative;
}
#logo_container
{
        height: 100%;
        width: 12%;
        background-color: #C5C6D0;
        display: flex;
        flex-direction: column;
        vertical-align: text-bottom;
}
.logo
{
        width:70%;
        margin: 5% 15%;

/*      border-radius: 50%; */
}
#logo_3
{
        vertical-align: text-bottom;
```

```
}
#data_container
{
	height: 100%;
	width: 20%;
	margin-left: 1%;
	margin-right: 1%;
	display: flex;
	flex-direction: column;
}
#data_status
{
	height:60%;
	width:8%;
	margin:7%;
	background-color: #691F6E;
	display: flex;
	flex-direction: column;
	border-radius:20px;
}
#load_status
{
	background-image: url("/Images/KG.png");
	background-repeat: no-repeat;
	 background-size:     170px;
	background-position: left center;
}
#cap_status
{
	background-image: url("/Images/dust.png");
	background-repeat: no-repeat;
	 background-size: 150px;
	background-position: left center;
}
.status
{
	width: 80%;
	height: 40%;
	margin:5% 10%;
	background-color:#185adc;
	border-radius:20px;
	display: flex;
	justify-content: center;
	align-items: center;
	color: white;
	font-size: 60px;
```

```
}
.datas
{
        width:86%;
        margin:2.5% 7%;
        height:10%;
   background: url(water.png);
    background-repeat: repeat-x;
    animation: datas 10s linear infinite;

    box-shadow: 0 0 0 6px #98d7eb, 0  20px 35px rgba(0,0,0,1);

}
#map_container
{
        height: 100%;
        width: 100%;
        display: flex;
        flex-direction: column;
}
#live_location_heading
{
        margin-top:10%;
        text-align: center;
  color:  GREY;
}
#map
{
        height: 70%;
        width: 90%;
        margin-left: 4%;
        margin-right:4%;
        border: 10px solid white;
        border-radius: 25px;
}
#alert_msg
{
        width:92%;
        height:20%;
        margin:4%;
        background-color:grey;
        border-radius: 20px;
        display: flex;
        justify-content: center;
        align-items: center;
```

```
        color: #41af7f;
        font-size: 25px;
        font-weight: bold;
}
.lat
{
        margin: 0px;
        font-size:0px;
}



@keyframes datas{
   0%
   {
      background-position: -500px 100px;
   }
   40%
   {
      background-position: 1000px -10px;
   }

   80% {
      background-position: 2000px 40px;
   }
   100% {
      background-position: 2700px 95px;
   }

}
```

DATABASE.JS

```
const cap_status = document.getElementById('cap_status');
const alert_msg = document.getElementById('alert_msg');

var ref = firebase.database().ref();

ref.on("value", function(snapshot)
 {
   snapshot.forEach(function (childSnapshot) {
      var value = childSnapshot.val();

                const alert_msg_val = value.alert;
                const cap_status_val = value.distance_status;
```

```
                alert_msg.innerHTML= `${alert_msg_val}`;
    });
 }, function (error) {
    console.log("Error: " + error.code);
 });
```

MAP.JS:

```
const database = firebase.database();

function myMap()
{
        var ref1 = firebase.database().ref();

                ref1.on("value", function(snapshot)
                 {
                   snapshot.forEach(function (childSnapshot) {
                      var value = childSnapshot.val();
                                const latitude = value.latitude;
                                const longitude = value.longitude;

                                var latlong = { lat: latitude, lng: longitude}
                                var mapProp =
                                {
                                        center: new google.maps.LatLng(latlong),
                                        zoom: 10,
                                };
                                var map = new
google.maps.Map(document.getElementById("map"), mapProp);

                                var marker = new google.maps.Marker({ position: latlong });
                                marker.setMap(map);
                    });
                 }, function (error) {
                   console.log("Error: " + error.code);
                 });

}
```

BIN PYTHON CODE

BIN1

```
const database = firebase.database();

function myMap()
{
        var ref1 = firebase.database().ref();

                ref1.on("value", function(snapshot)
```

```
                {
                  snapshot.forEach(function (childSnapshot) {
                     var value = childSnapshot.val();
                              const latitude = value.latitude;
                              const longitude = value.longitude;

                              var latlong = { lat: latitude, lng: longitude}
                              var mapProp =
                              {
                                       center: new google.maps.LatLng(latlong),
                                       zoom: 10,
                              };
                              var map = new
google.maps.Map(document.getElementById("map"), mapProp);

                              var marker = new google.maps.Marker({ position: latlong });
                              marker.setMap(map);
                  });
                }, function (error) {
                  console.log("Error: " + error.code);
                });

}
```

BIN 2

```
import requests
import json
import ibmiotf.application
import ibmiotf.device
import time
import random
import sys

# watson device details

organization = "4yi0vc"
devicType =  "BIN2"
deviceId = "BIN2ID"
authMethod= "token"
authToken= "123456789"

#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,"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 "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=10.939091,long=78.135731):
        print("Bustand, 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()
```

BIN3

```
import requests
import json
import ibmiotf.application
import ibmiotf.device
import time
import random
import sys
```

```
# watson device details

organization = "4yi0vc"
devicType =  "BIN3"
deviceId = "BIN3ID"
authMethod= "token"
authToken= "123456789"

#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,"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 "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 :' 'Risk Warning: 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=10.939091,long=78.135731):
       print("Thirumanilayur, Karur")
       print("published distance = %s " %distance,"loadcell:%s " %loadcell,"lon = %s "
%long,"lat = %s" %lat)
       print(load)
       print(dist)
       print(warn)

    time.sleep(5)


    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(5)
```

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

BIN4

```
import requests
import json
import ibmiotf.application
import ibmiotf.device
import time
import random
import sys

# watson device details

organization = "4yi0vc"
devicType =  "BIN4"
deviceId = "BIN4ID"
authMethod= "token"
authToken= "123456789"

#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,"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 "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 :' 'Risk Warning: 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=10.939091,long=75.135731):
        print("Puliyur, Karur")
        print("published distance = %s " %distance,"loadcell:%s " %loadcell,"lon = %s "
%long,"lat = %s" %lat)
        print(load)
        print(dist)
```

```
        print(warn)

    time.sleep(5)


    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(5)



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