PROJECT REPORT

Date	18 May 2023
Team ID	NM2023TMID08877
Project Name	Smart City Waste Management System with Connected Trash Cans
Team Members	Arunachalam K (Team Leader) Viknesh S (Team Member) Sivakumar K (Team Member) Sudharson S (Team Member) Kalidass S (Team Member)

INTRODUCTION

1.1 Project Overview

The Smart City Waste Management System with Connected Trash Cans is a new approach to waste management that uses sensor technology, cloud computing, and data analysis to improve efficiency and sustainability. The system works by installing sensors in trash cans that measure the fill level and temperature of the waste. This data is then transmitted to a cloud server, where it is analyzed to identify waste generation patterns and optimize collection routes and schedules. The system also provides real-time data on waste levels to waste management companies and city officials. This information can be used to improve waste management practices and prevent overflowing trash cans. The Smart City Waste Management System with Connected Trash Cans is a valuable tool for improving waste management in urban areas. It can help to reduce the amount of waste sent to landfills, improve air quality, and create a cleaner, more sustainable environment.

1.2 Purpose

The purpose of the Smart City Waste Management System with Connected Trash Cans project is to improve waste management in urban areas by:

- Ensuring timely trash collection through real-time monitoring of fill levels.
- Optimizing resource allocation and routes for efficient waste collection.
- Promoting sustainability by preventing overflowing trash cans and maintaining cleanliness.
- Providing real-time monitoring and alerts for waste management staff.
- Engaging citizens in waste management efforts through a user-friendly app or web interface.

1.3 LITERATURE SURVEY

Existing problem:

- Manual systems in which employees clear the dumpsters periodically.
- No systematic approach towards clearing the dumpsters.
- Unclear about the status of a particular location
- Employees are unaware of the need for a particular location.
- Very less effective in cleaning city.

References:

S.NO	TITLE	AUTHOR & YEAR	DESCRIPTION
1.	Smart Waste Management using WSN and IOT	Sivasankari, Bhanu Shri, Y. BevishJinila 2017	In this paper, they use Wireless Sensor Networks and IOT. The garbage bins are deployed with sensors and are networked together using WSN. The sensorsdeployed in the garbage bins collect the data for every determined interval. Once the threshold is reached, it raises a requestto the GCA (garbage collector agent). This agent collects the requests of all the filled vehicles and communicates using the IoT framework.
2.	Smart Waste Management System using IOT	Tejashree Kadus, Pawan kumar Nirmal,Kartikee Kulkarni 2020	In this paper, they use an Arduino board interface with a load sensor, an IR sensor, and a Wi-Fi module instead of a PIR sensor and an ultrasonic sensor. In addition to electrical components, they use mechanical components like the load sensing plate and shredder to crash the trash and then measure the load.
3.	Smart Waste Management System	Bindu Shree, Manasa, Sanjana Rao, Vidhya Shree, Gowra PS 2021	In this paper, they use sensors, which include an IR sensor for detecting the presence of any waste and a soil moisture sensor to detect whether the waste is dry or wet. The emphasis is primarily on waste segregation, followed by analysis via the website.

IDEATION & PROPOSED SOLUTION

2.1 Problem Statement Definition

Customer Problem Statement:

Fig 2.1.1 Customer Problem Statement

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	Busy city worker	Dispose of my waste	Hard to find trash cans	Trash cans are often overflowing	Guilty and frustrated
PS-2	Business owner	Maintain a clean and welcoming storefront	Traditional trash cans attract pests	Trash cans often overflow	Worried about the impact on my customers
PS-3	Local	Reduce	It is hard to	Traditional	Helpless and

	environmental activist	waste and promote sustainability	make a meaningful impact on peoples	waste management systems are inefficient	frustrated
PS-4	City official	Manage waste and maintain a clean and safe environment	It is challenging to provide adequate resources	Traditional waste management systems are expensive and inefficient	Overwhelmed and like I am failing to meet the needs of the community

Table 2.1.1 Customer Problem Statement

2.2 Empathy Map Canvas

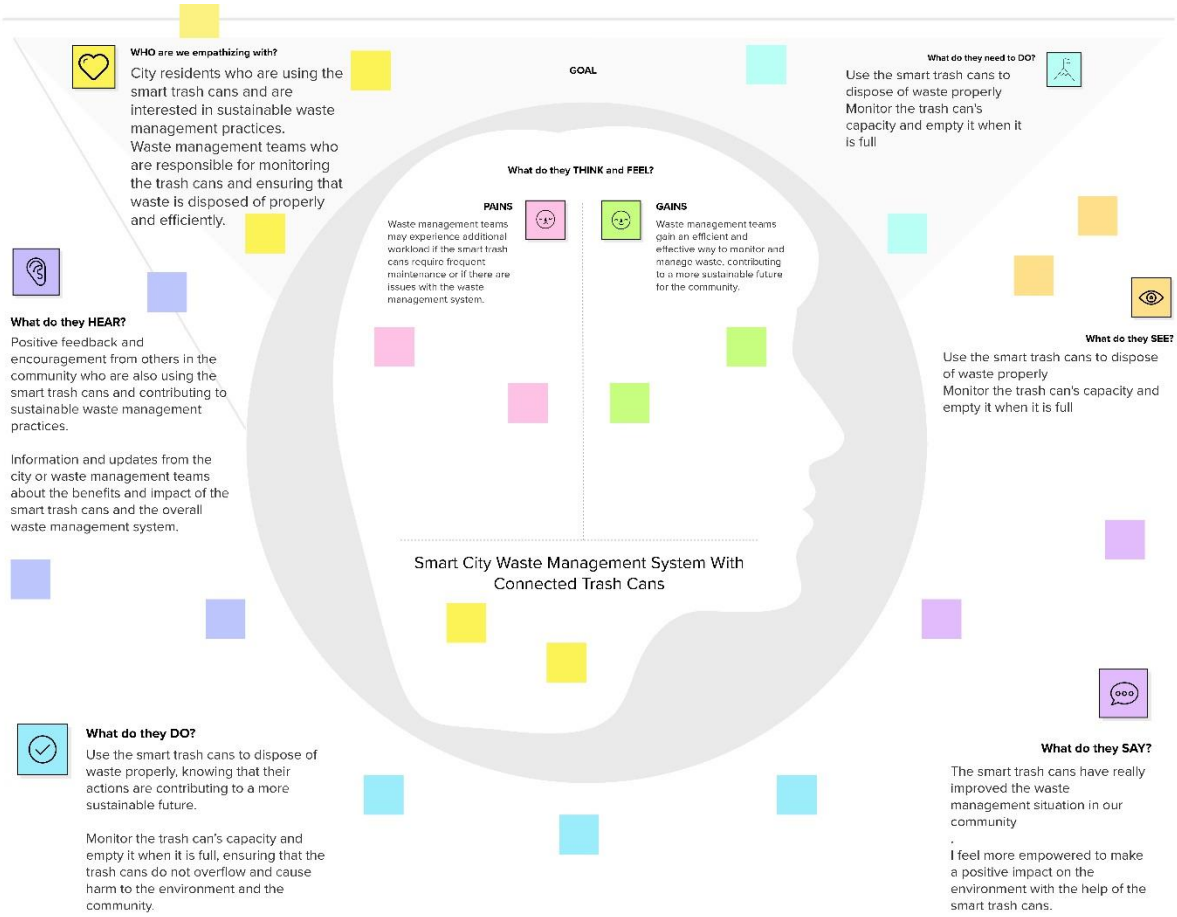

Fig 2.2.1 Empathy Map Canvas

2.3 Ideation & Brainstorming

Team Gathering, Collaboration and Select the Problem Statement

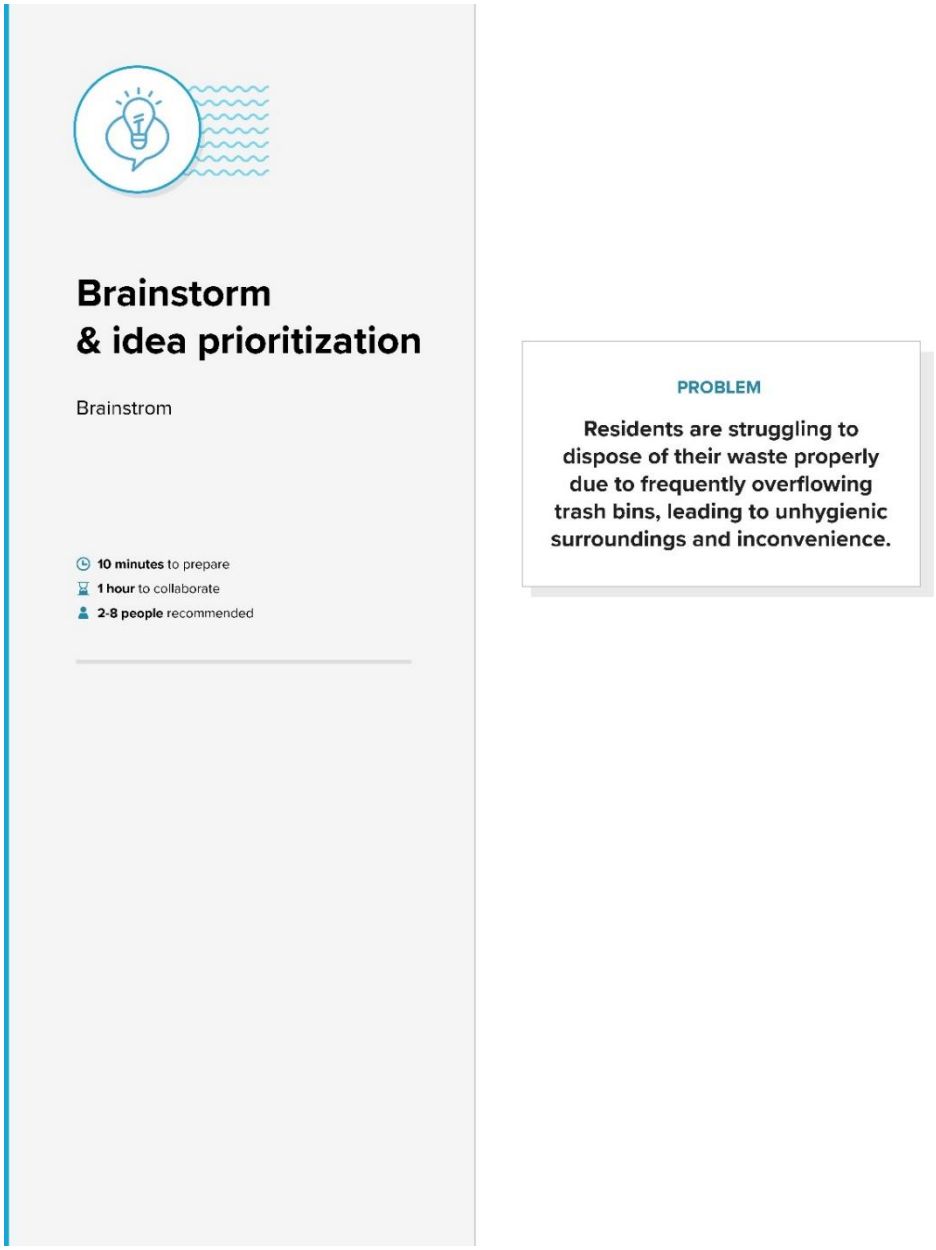

Fig 2.3.1 Brainstorm

Brainstorm, Idea Listing and Grouping

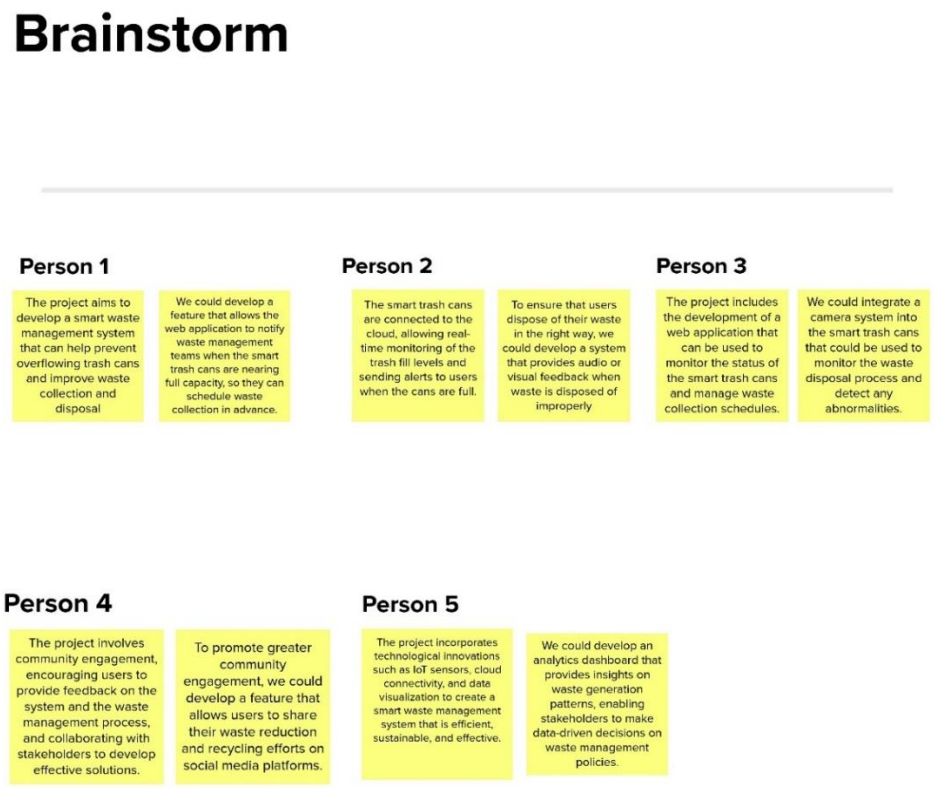

Fig 2.3.2 Brainstorm individual idea

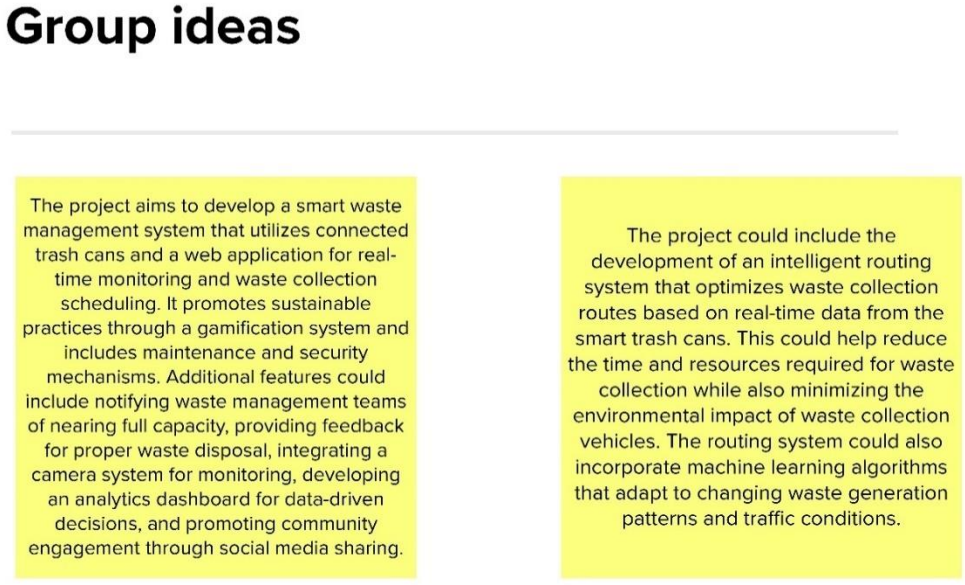

Fig 2.3.3 Brainstorm Group idea

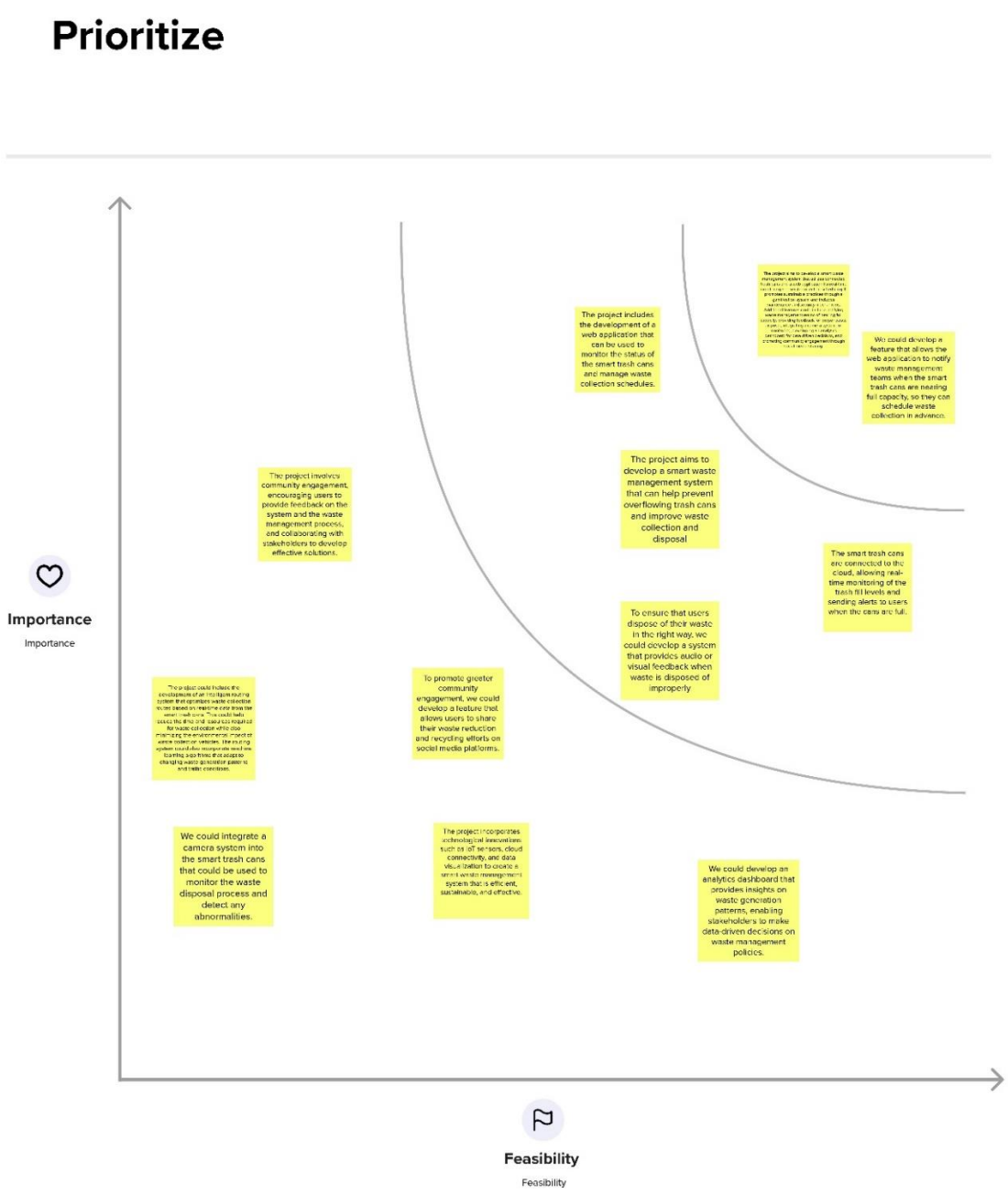

Fig 2.3.1 Idea Prioritization

2.4 Proposed Solution

S. No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Traditional waste management systems in cities are often inefficient and result in overflowing trash cans and littered streets. Additionally, waste management personnel must manually check and empty trash cans, resulting in additional costs and time. There is a need for a more efficient waste management system that can automatically monitor and manage trash cans in urban environments.
2.	Idea / Solution description	We propose developing a smart city waste management system with connected trash cans that use IoT technology to enable real-time monitoring and management of waste in urban environments. The system will include sensors and connectivity features that can detect when a trash can is full or needs maintenance, enabling waste management personnel to act quickly and efficiently.
3.	Novelty / Uniqueness	Our solution is unique because it leverages IoT technology to enable real-time monitoring and management of trash cans in urban environments. The system uses advanced sensors and connectivity features to detect when a trash can is full or needs maintenance, improving the efficiency and effectiveness of waste management operations.
4.	Social Impact / Customer Satisfaction	By providing a more efficient and effective waste management solution, our system will have a positive impact on the environment and reduce costs for municipalities and businesses. The system's real-time monitoring and management features will also improve the cleanliness and overall aesthetic of urban environments, leading to increased satisfaction among residents and visitors.
5.	Business Model (Revenue Model)	Our business model is based on a subscription-based revenue model, where clients pay a monthly fee for access to the system's features and capabilities. We will also offer additional customization and integration services for an additional fee.
6.	Scalability of the Solution	Our solution is highly scalable and can be easily adapted to other urban environments. As the system is based on IoT technology, it can handle large volumes of data and can be integrated with other systems and technologies as needed. Additionally, the system's modular design allows for easy installation and maintenance.

REQUIREMENT ANALYSIS

3.1 Functional requirement

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIn
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Authentication and Access Control	User login using username and password. User roles and permissions management
FR-4	Trash Can Monitoring	Real-time monitoring of trash can status Detection of trash can fill level. Tracking of location and identification of each trash can
FR-5	Notification and Alerts	Alert users when a trash can is full or nearing capacity. Notification via email, SMS, or mobile app push notification
FR-6	Web Application Interface	Display real-time trash can statuses on the web app. Provide a user-friendly interface for managing trash cans. Support filtering and sorting of trash can data
FR-7	Cloud Integration	Connect trash cans to the cloud platform. Transmit trash can data securely to the web app. Ensure data integrity and reliability
FR-8	Data Storage and Management	Store trash can data securely in a database. Retain historical data for reporting and analysis. Enable data backup and recovery mechanisms
FR-9	Reporting and Analytics	Generate reports on trash can fill levels and trends. Provide insights for waste management planning. Analyze data to identify high-traffic areas or

		problem spots
FR-10	Geolocation and Mapping	Map trash can locations on an interactive map | | Display nearby trash cans based on user's current location
FR-11	IoT Device Integration	Integrate IoT-enabled trash cans for data collection. Connect trash cans to IoT platforms (e.g., Watson IoT). Exchange sensor data between devices and the cloud platform
FR-12	Integration with Wokwi Platform	Explore Wokwi platform for device simulation | | Utilize Wokwi libraries for IoT device development | | Code to connect devices across the cloud platform
FR-13	Node-RED Application Development	Gain knowledge of web application development through Node-RED. Develop Node-RED flows for data processing and visualization. Integrate Node-RED with the web app for real-time monitoring

3.2 Non-Functional requirements

FR No.	**Non-Functional Requirement**	**Description**
NFR-1	Usability	This has a user-friendly interface and intuitive navigation for ease of use by both administrators and end-users.
NFR-2	Security	This ensures the confidentiality, integrity, and availability of data and has proper authentication and access control mechanisms to prevent unauthorized access.
NFR-3	Reliability	This is highly reliable, ensuring that the trash can monitoring, notification, and data transmission functions work consistently without failures or disruptions.
NFR-4	Performance	This can handle many connected trash cans and users, with minimal latency in data processing and real-time monitoring.

NFR-5	Availability	This is available and accessible to users 24/7, with minimal downtime for maintenance or system updates.
NFR-6	Scalability	This is designed to be scalable, supporting the addition of new trash cans and users without impacting performance or functionality.

PROJECT DESIGN

4.1 Data Flow Diagrams

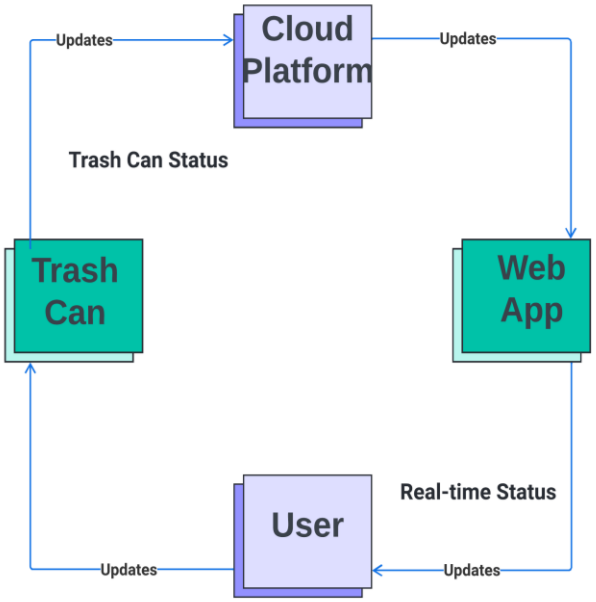

Fig 4.1.1 Data flow diagram

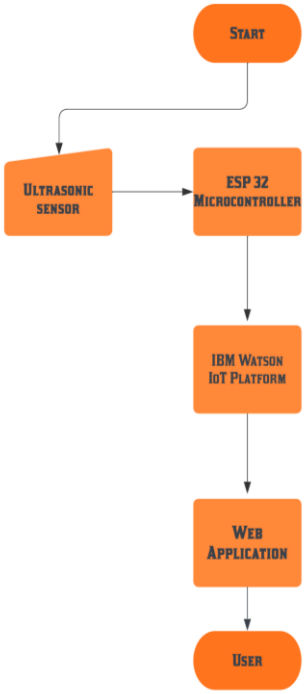

Fig 4.1.2 Data flow flowchart

4.2 Solution & Technical Architecture

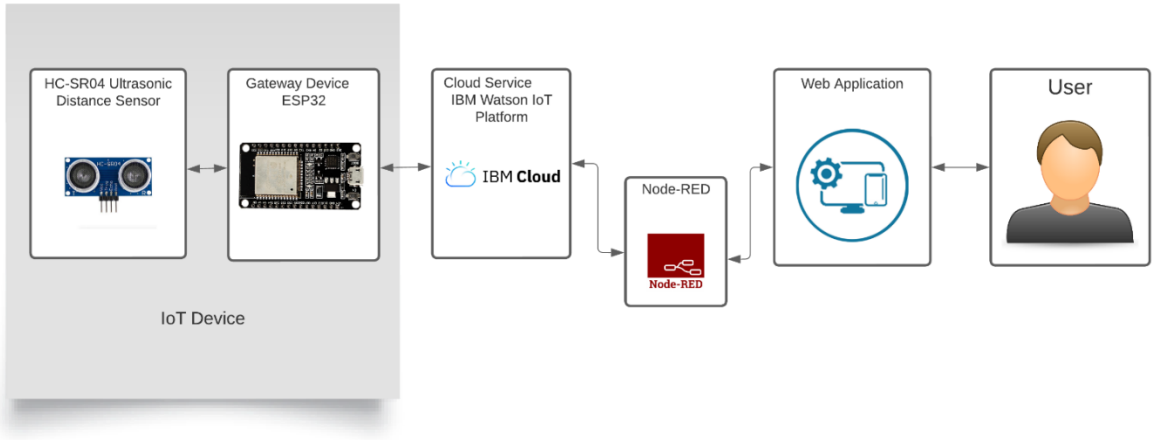

Fig 4.2.1 Solution Architecture

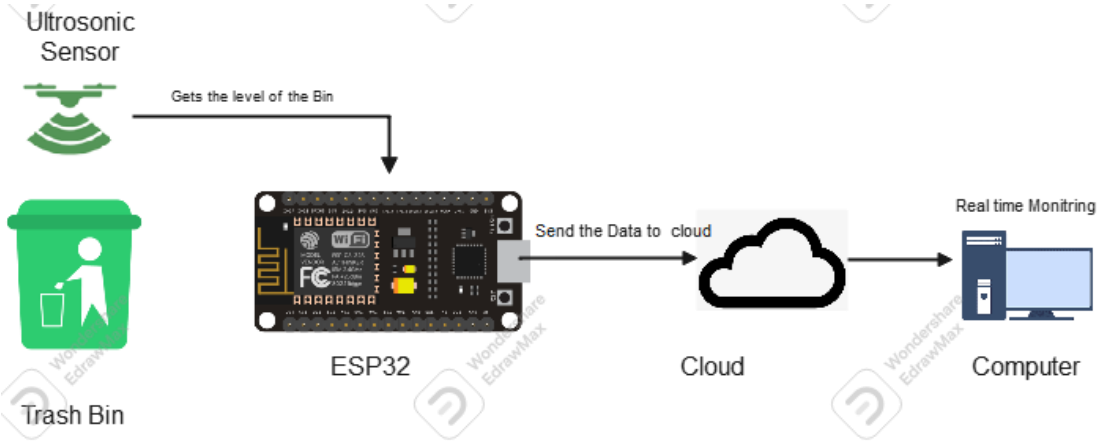

Fig 4.2.2 Technical Architecture

4.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Team Member
Customer (Mobile user)	Registration	USN-1	As a mobile user, I can register on the app, so that I can receive notifications about the status of the trash cans.	The app's registration page collects user information and sends a confirmation message upon successful registration.	High	Arunachalam
		USN-2	As a mobile user, I can receive real-time alerts when the trash cans are full, so that I can take necessary actions.	The app notifies users when trash cans are full and provides a map view of their locations.	High	Sudharson
		USN-3	As a mobile user, I can view the fill level of the trash cans in real-time, so that I can plan my waste disposal activities accordingly.	The app displays a real-time list of connected trash cans with fill levels and allows users to filter based on location, fill level, or waste type.	Medium	Kalidass
		USN-4	As a mobile user, I can report any issues with the trash cans, such as damage or overflowing, so that the authorities can take necessary actions.	The app allows users to report trash can issues, including the option to attach photos or videos, which are then sent to the authorities for action.	Medium	Viknesh
		USN-5	As a waste management personnel, I can track the maintenance status of the trash cans, so that I can schedule timely repairs and replacements.	It facilitates tracking of trash can statuses, reporting of issues, and monitoring the progress of maintenance requests.	High	SivaKumar

CODING & SOLUTIONING

5.1 Feature 1

- A device is designed to detect the status of the bin in the streets of the city.
- The device is embedded with Esp32 board and ultrasonic sensor.
- The device is designed in using wokwi platform for simulation.

5.2 Feature 2

A user interface is developed using the following services:

- MIT App Invertor
- IBM IoT Watson Platform
- Node-Red service
- Wokwi simulator Web Application

RESULTS

6.1 Performance Metrics

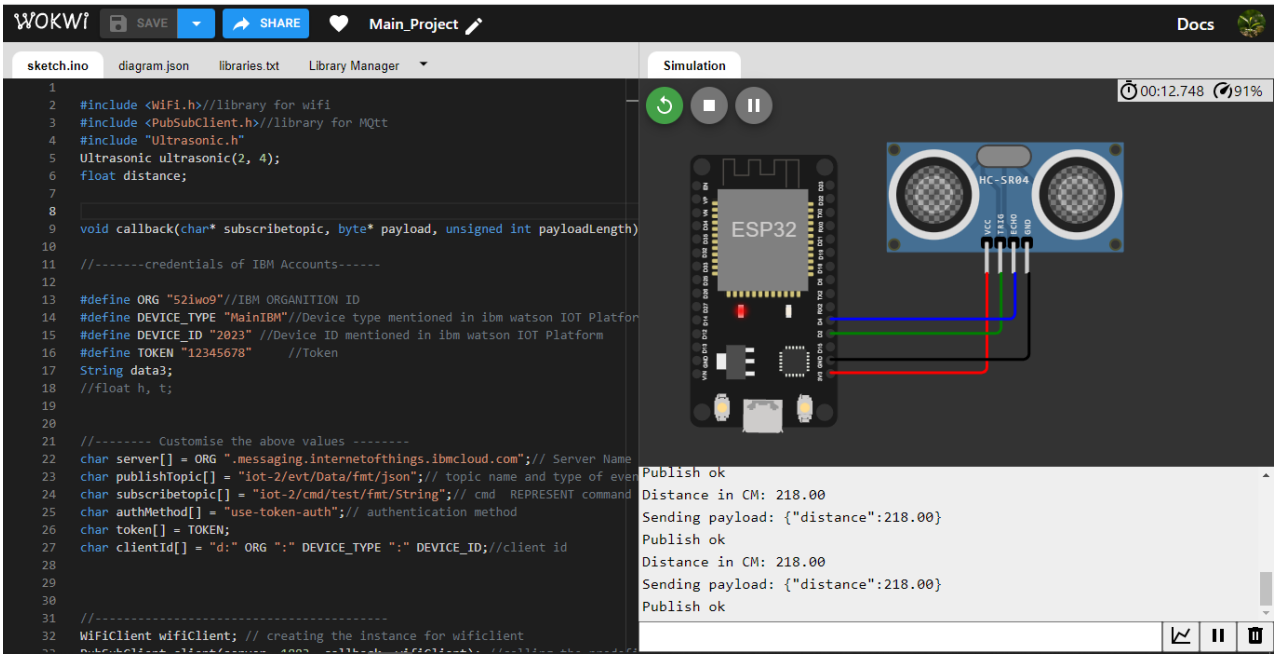

Fig 6.1.1 Running Simulation

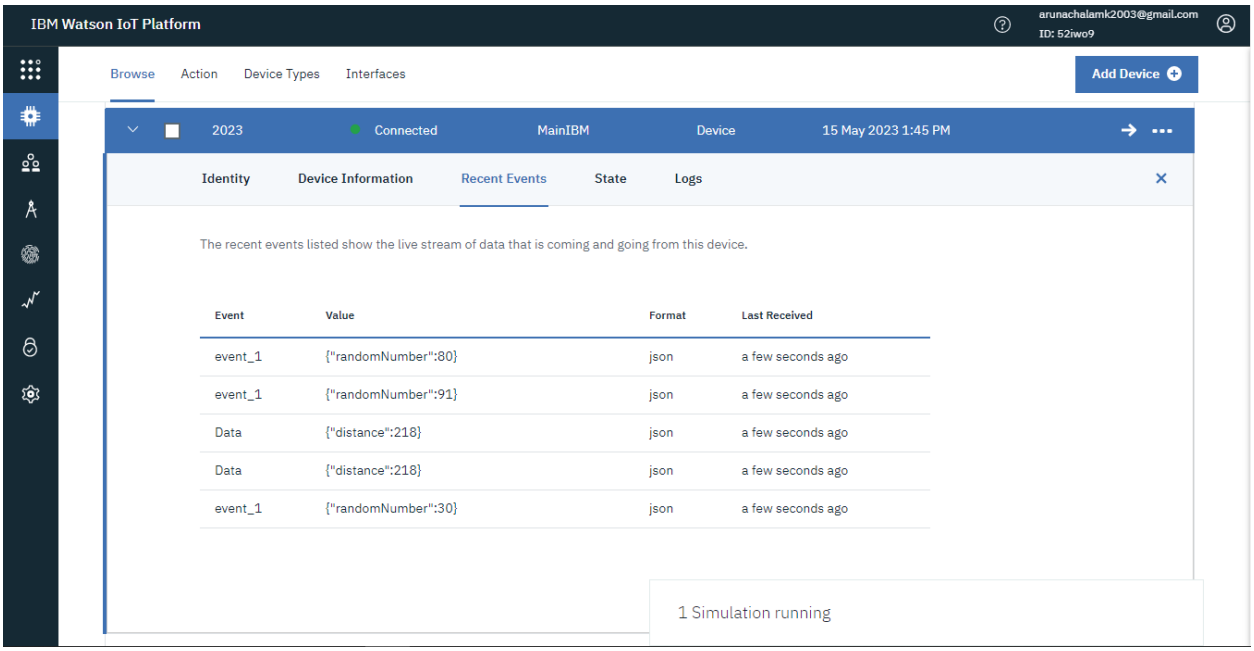

Fig 6.1.2 Iot Watson Platform

Node-red flow and Dashboard:

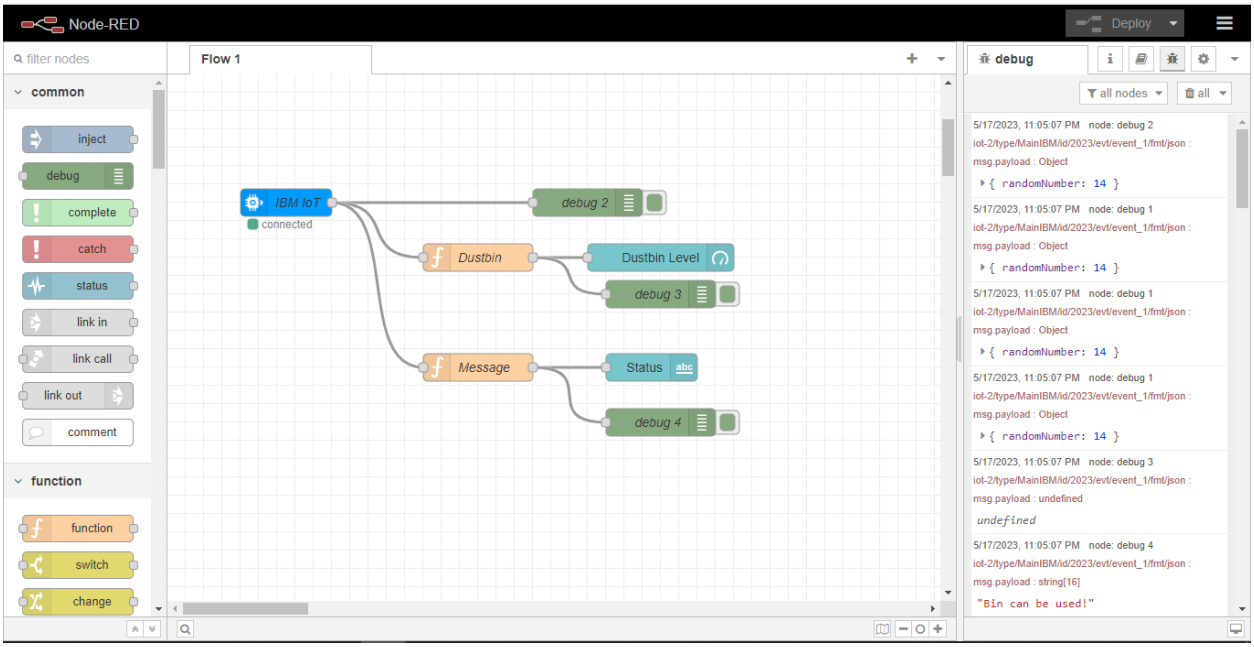

Fig 6.1.3 Node-red flow

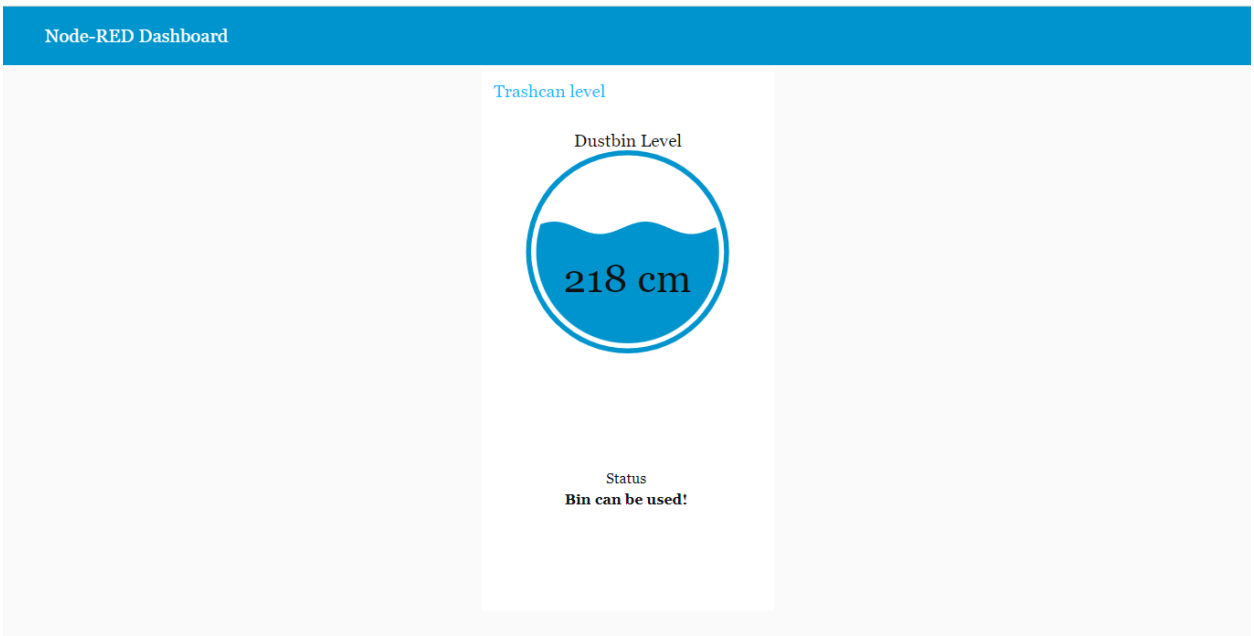

Fig 6.1.4 Node-red Dashboard

ADVANTAGES

Reduced overflowing trash cans: The system can detect when a trash can is full and send an alert to waste management crews. This prevents overflowing trash cans, which can be unsightly and attract pests.

Improved efficiency of waste management services: The system can track the fill levels of trash cans in real time. This information can be used to optimize waste collection routes and schedules. This can save waste management companies time and money.

Real-time data on waste levels: The system can provide real-time data on waste levels to waste management companies and city officials. This information can be used to improve waste management practices and prevent overflowing trash cans.

Reduced environmental impact: The system can help to reduce the amount of waste sent to landfills. This is because the system can help to optimize waste collection routes and schedules. This can reduce the amount of fuel used by waste management vehicles, which can reduce greenhouse gas emissions.

Increased sustainability: The system can help to increase the sustainability of waste management. This is because the system can help to reduce the amount of waste sent to landfills. Landfills are a major source of methane, a greenhouse gas that is more potent than carbon dioxide.

DISADVANTAGES

Cost: The system is expensive to install and maintain. The cost of the system will vary depending on the size of the city and the number of trash cans that are equipped with sensors.

Privacy: The system collects data on waste levels and other information. This data could be used to track people's movements and habits.

Security: The system is vulnerable to hacking. If the system is hacked, it could be used to disrupt waste management services or to steal data.

Acceptance: The system may not be accepted by the public. Some people may be concerned about the privacy implications of the system or the cost of the system.

Weather conditions: In some areas, extreme weather conditions can damage the sensors or make the system unreliable.

Power outages: In some areas, power outages can disrupt the system. This can lead to overflowing trash cans and other problems.

CONCLUSION

In conclusion, the Smart City Waste Management System is a valuable tool for improving waste management in urban areas. It has the potential to reduce overflowing trash cans, improve the efficiency of waste management services, and reduce the environmental impact of waste management. However, it is important to be aware of the disadvantages of the system before implementing it. These disadvantages include cost, privacy, security, acceptance, trash can vandalism, trash can theft, weather conditions, and power outages. It is important to consider these factors when planning to implement a Smart City Waste Management System with Connected Trash Cans. Overall, the Smart City Waste Management System with Connected Trash Cans is a promising new technology that has the potential to revolutionize waste management in urban areas. It is a valuable tool for improving efficiency, sustainability, and environmental impact.

FUTURE SCOPE

The Smart City Waste Management System with Connected Trash Cans has the potential to revolutionize waste management by improving efficiency, sustainability, and environmental impact. In the future, the system can be used to develop intelligent waste management systems, promote sustainable waste management practices, engage the public in waste management, make data-driven decisions about waste management, and integrate with other smart city technologies. These developments have the potential to make a significant impact on the environment and improve the quality of life in urban areas.

APPENDIX

Source Code

```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt
#include "Ultrasonic.h"
Ultrasonic ultrasonic(2, 4);
float distance;

void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);

//-------credentials of IBM Accounts------

#define ORG "52iwo9"//IBM ORGANITION ID
#define DEVICE_TYPE "MainIBM"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "2023" //Device ID mentioned in ibm watson IOT Platform
#define TOKEN "12345678"     //Token
String data3;
//float h, t;

//-------- Customise the above values --------
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name
char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of event perform and format in which data to be send
char subscribetopic[] = "iot-2/cmd/test/fmt/String";// cmd  REPRESENT command type AND COMMAND IS TEST OF FORMAT STRING
char authMethod[] = "use-token-auth";// authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id

//-----------------------------------------
WiFiClient wifiClient; // creating the instance for wificlient
PubSubClient client(server, 1883, callback ,wifiClient); //calling the predefined client id by passing parameter like server id,portand wificredential
void setup()// configureing the ESP32
{
  Serial.begin(115200);
  delay(10);
  Serial.println();
  wificonnect();
  mqttconnect();
}
```

```
void loop()// Recursive Function
{

  distance = ultrasonic.read(CM);

  Serial.print("Distance in CM: ");
  Serial.println(distance);
  delay(1000);

  PublishData(distance);
  delay(1000);
  if (!client.loop()) {
    mqttconnect();
  }
}

/*.....retrieving to Cloud.....*/

void PublishData(float distance) {
  mqttconnect();//function call for connecting to ibm
  /*
     creating the String in in form JSon to update the data to ibm cloud
  */
  String payload = "{\"distance\":";
  payload += distance;

  payload += "}";
  Serial.print("Sending payload: ");
  Serial.println(payload);
  if (client.publish(publishTopic, (char*) payload.c_str())) {
    Serial.println("Publish ok");// if it sucessfully upload data on the cloud then it will print
publish ok in Serial monitor or else it will print publish failed
  } else {
    Serial.println("Publish failed");
  }

}
void mqttconnect() {
  if (!client.connected()) {
    Serial.print("Reconnecting client to ");
    Serial.println(server);
    while (!!!client.connect(clientId, authMethod, token)) {
      Serial.print(".");
      delay(500);
    }
```

```
    initManagedDevice();
    Serial.println();
  }
}
void wificonnect() //function defination for wificonnect
{
  Serial.println();
  Serial.print("Connecting to ");

  WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to establish the
connection
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void initManagedDevice() {
  if (client.subscribe(subscribetopic)) {
    Serial.println((subscribetopic));
    Serial.println("subscribe to cmd OK");
  } else {
    Serial.println("subscribe to cmd FAILED");
  }
}

void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
{

  Serial.print("callback invoked for topic: ");
  Serial.println(subscribetopic);

  for (int i = 0; i < payloadLength; i++) {
    //Serial.print((char)payload[i]);
    data3 += (char)payload[i];
  }

  Serial.println("data: "+ data3);

data3="";

}
```