Smart Waste Management System For Metropolitan Cities

Team ID: PNT2022TMID39384

Team Members:

1. 422719106006 A.JACOBSTANLY
2. 422719106007 M.JAYASHARMA
3. 422719106004 A.GOWTHAM
4. 422719106023 SK.SYED HUSSAIN

INDEX

- **INTRODUCTION**

Project Overview

The rate at which solid wastes are produced in most developing countries is becoming alarming. This increase may be due to recent population growth and rural-urban mi-gration [1]. Garbage is made up of non-renewable resources used daily to meet our needs then throw away. As increase in consumption of paper, clothing, bottles, and product packaging increases, the generation of garbage also increases significantly. The form and type of solid waste depends on a number of

factors which include the living standard and life style of the inhabitants of the region and the natural resourcesound in the region. There are two categories of Urban waste namely, organic and inorganic. The organic waste category can be further categorized into three units: non-fermentable, fermentable and putrescible [1]. The Putrescible wastes tend to decay faster, and if not cautiously managed, decomposition can lead to an offensive odour with an unpleasant view. Fermentable waste which also tends to decompose rapidly do so without the accompanying of offensive odour. Non-fermentable waste most times do not decompose or do so at a very slow rate. Unless organic waste is managed appropriately, the stricken negative effect it has will continue until full decomposition or stabilization occurs. Decomposed products which are poorly managed or uncon-trolled can and often times lead to contamination of air, water and soil resources [2]. One of the challenges a developing country faces due to rapid increase in popula-tion is proper solid waste management. A typical example is the garbage bins seen around which appear overfull to the point of spilling out, leading to environmental pollution. The effect of this is increase in the number of diseases because it gives room for insects to breed. Solid waste requires systematic management the content, origin or hazard potential notwithstanding as this will ensure environmental best prac-tices and living standard. Because solid waste management forms a very critical as-pect of our environmental hygiene, it is therefore necessary to incorporate it into environmental planning [3]. The recent advances in computers have led to the birth of new innovations and op-portunities like the Internet of Things where things (embedded systems) that are con-nected to the internet can also be controlled and interacted with via the internet. The term Internet of Things (IoTs) was first introduced by Kevin Ashton, a former direc-tor of the Auto-ID Centre of MIT in 1999 [4]. The idea of IoTs is to connect objects around us through wired and

wireless network with human intervention. Communica-tion and exchange of information are carried out by the object to provide advance intelligent service for the users. In the case of the proposed solid waste management system, the bins are connect-ed to the internet to relay real-time information of the status of the bin. The rapid growth in population in recent years has led to more waste disposals, necessitating the need for a proper waste management system to avoid unhygienic living conditions. Implementation of the system translates to the bin being interfaced with microcontrol-ler-based system with ultrasonic sensors and a Wi-Fi module. The data which would be sent from the bins would be received, analysed and processed in the ThingSpeak cloud that displays the level of the garbage in the bin on a graph in its web page. The main drive of solid waste management is the reduction and elimination of ad-verse effect of waste materials on human health and environment leading to im-provement in quality of life. In this work, an intelligent solid waste monitoring system is developed using Internet of Things (IoT) and cloud computing technologies. This is a recent innovation as cloud computing has been applied in other areas like [5][6][7][8]. Ultrasonic sensors are employed to detect the fill level of solid waste in each of the containers. The data obtained by the sensor is then transmitted to an IoT cloud platform, called ThingSpeak, using a Wi-Fi communication link. For each des-ignated fill level, the system sends appropriate notification message (in form of tweet) to alert relevant authorities and concerned citizen(s) for necessary action. Also,

the fill level is monitored on ThingSpeak in real-time.

- Purpose

Using technology and innovation to optimize current systems will enable cities to become smarter, more efficient and save resources. Due to the growing population, the amount of waste being produced is vast and rapidly increasing. The

management of this waste is therefore a significant area for much-needed improvement.

Currently, waste collection systems are in most cases outdated and result in pickups that are unnecessary or on the contrary – long-overdue. Unnecessary pickups result in 70% higher annual collection cost. When routes are planned inefficiently, congestion is created and more fuel is required to complete the collection. Overall, this contributes to a 50% higher carbon footprint.

With the use of IoT solutions for waste management, these issues can be solved by creating a more efficient pathway for garbage trucks. IoT sensor technology can be used to indicate when the emptying is actually needed. This customized and dynamic system for waste management can allow businesses, organizations, and citizens to all benefit.

#1 Time-saving

By having a more convenient route garbage trucks spend less time on the road, therefore, congestion in smart cities can be decreased. This means that truck

drivers and citizens are saving less time stuck in traffic jams. Additionally, using IoT technology for remote diagnostics also means not having to send staff all the way to monitor assets.

#2 Cost-saving

With the huge increase in waste, more resources are allocated to waste collection and handling. If unnecessary collections are eliminated, public spending on waste management can be reduced.

This frees up resources for municipalities to allocate to other initiatives. Moreover, waste is properly handled and sorted and turned into recyclable assets, this provides a further potential income stream.

#3 Sustainability

Overflowing bins will pollute the environment potentially contaminating areas and harming the general health of the public. An optimized route and system for waste collection will eliminate this risk as well as improving air quality and minimizing CO2 emissions. Smart cities can reduce their overall carbon footprint, bringing them closer to achieving the SDG goals.

#4 Improving efficiency

Smart cities are all about using resources efficiently – achieving more by using less input. One of the benefits includes the availability of real-time data, allowing for decisions to be made quickly. This means that action can be taken before having an overflow of containers. Smart cities can remain highly responsive and challenge the current waste hierarchy, breaking patterns of inefficiency and high costs.

#5 Transparency

By routes being monitored, the opportunity of the misuse of owned assets is eliminated.

Moreover, cities are encouraged to be transparent with their citizens by showing how waste is being managed. By making data publicly available, trust can be built with citizens. Smart cities should be encouraged by their operations and solutions to strive for more sustainable development, not limited by them.

#6 Meet the increasing demand for sustainable solutions

As society is becoming more aware of the environmental issues – there is an increasing demand for more sustainable and environmentally friendly solutions, when choosing between services and products. Smart cities infrastructure and intelligent waste solutions have the potential to lead the shift towards a more sustainable future.

- **LITERATURE SURVEY**

Swacch Bharat Abhiyan and digital India is a campaign by the government of India to keep infrastructure of the country clean and to make the cities smarter. Day by day the population of India is growing rapidly. At the same time, the garbage also is growing at the same rate. As a result the garbage management is a problem that is quite hectic issue to solve. All Citizens of India are aware about the process followed to collect the garbage in the society. The Brihan Mumbai Municipal Corporation (BMC) sometimes fails to collect the garbage in some area. It may cause pollution which leads to sanitary issues and disease. Therefore, some of the major steps have to be carried out to solve the management of waste. The existing system is collection of garbage arbitrarily. So, some of the areas get left sometimes which may lead to unodoured smell and hence public health gets affected. The smell of the garbage can also be fatal to some of the little ones in some areas. The proposed system describes the solution to the existing drawback. The proposed system monitors the garbage bin. While monitoring the garbage bin it sends the notification to the authority about the level of garbage filled. If the lower authority ignores the notification, the next notification goes to the higher authority. The proposed system will help them to actually know that where and when to go to collect the garbage. The proposed system manages

the effort to check the area by visiting there. The proposed project is quite helpful for both the Brihan Mumbai Municipal Corporation (BMC) and the citizens in that area by time to time interaction between Brihan Mumbai Municipal Corporation (BMC) and the proposed system. Hence the proposed system makes a better way to manage garbage.

In the project title “Smart Dustbin-An Efficient Garbage Monitoring System,” have proposed a ideaof between the existing dustbins and their population. This study and first part and the distribution of dustbins in some areas were taken palace of Dhaka city using averaging function and then the nearest neighbour functions of GIS . Remarkably, the procedure of the dustbins used today is one of the concern that is concentrated on the smart dustbin. The insufficient of the existing dustbins will have the number of number which will be calculated in it.It has measure the extent of pollution which is caused by the existing dustbins was also one of the level of the research part. It is found that the dustbins are burnt with wastes and has disturb and has cause pollution to the environment.

In the project titled "IoT Based Waste Management for Smart City" , it is being proposed in the project that has the introduction and the combination of the integrated system combined with an RFI, IGPS, GPRS, GIS and web camera which will solve the problem of waste They will help us to analyzed the actual performance of the system.To the study. In the project titled "IoT Based Waste Management for Smart City" , it is being proposed in the project that has the introduction and the combination of the integrated system combined with an RFI, IGPS, GPRS, GIS and web camera which will solve the problem of waste They will help us to analyzed the actual performance of the system.To the

study of the project we determine that the characterization of the waste and the current system of management activities. The project gives us the highlights and a overview of the municipal solid waste management (MSWM) system of Municipality and it concludes with a few suggestions, which may be beneficial to the authorities to work towards further improvement of the current management system.

In the project titled "Pawar2International Journal Of Engineering And Computer Science " is being proposed system in that novel prototype of solid waste bin monitoring system using network. The architecture will uses Zigbee and GSM(global service module) it has the communication technology as well and a set of chosen sensors it will monitor the status of garbage bins in real time. The project is divided into three parts lower tier, middle tier, upper tier. The low tier will have the sensor installed into the garbage bins, the middle tier will have the collection and then it that will give the information to the control station .In the upper tier we will store the data for future use. An algorithm of an energy is used in the first tier operation to collect the bin level.

In the project “A Survey on Smart Garbage Management in Cities using IoT” has proposed a dustbin which is interfaced with a microcontroller system in which we will having wireless systems along with central system showing the updated status of garbage, on web browser with html page by Wi-Fi Module. Hence the status will be updated on the html page. we have to reduce human efforts along with the enhancement of a smart city vision. Considering the modern technology, then we have a smart garbage bin can be costing but considering the amount of dustbin we all need in India, so there for then we have used sensors to reduce the cost and make it efficient in applications. And

they used only a Wi-Fi module to send and receive data. But the consideration of weight sensor will have the detection of garbage level which was there in the dustbin. It will only detect the weight of the waste. The message can be sent directly to the cleaning vehicle instead of the contractor's office.

In the project title “Garbage Monitoring System for Smart Cities,” proposed a model for which there is the collection of garbage in real time. There is a network which established using the sensors, which are placed into the garbage bin and have set at a level. Sensors will send a signal to the nearest vehicle driver if the level of garbage is crossed to empty the bin and hence the bin gets updated from time to time. Fig. 3 shows the Architectural Diagram, which consists major three modules; Sensor Module, in which sensors are used to sense the garbage levels once and connected to the Arduino board, Communication Module, in which Bluetooth is used for communication between the sensors and Arduino Uno board, and last module is Analysis and Monitoring Module, in which collected is sent to the admin for analysis.

In the project title “MATEC Web of Conferences 97, 01098,”
, the level of the garbage in each bin is measured by using the sensor. The information of the sensor is then received and processed by the Arduino Board. It will determine whether the garbage level has been reached to the threshold. For the research part, two marks have been made as a reference. The first is at the 70% and the second is at 90 %of the total bin height. If the garbage level in the bin is crossing the first. reference level, then the first warning message is generated and sent to the municipality. Besides, the green LEDs responds to alert all the residents at every floor. Next, if the garbage level in the bin is crossing the second reference level, then the second warning

message is generated and sent to the municipality. In that case, all the people will be alert when the red LEDs are at the high at their data pins.

In project title “Multipurpose Garbage Monitoring System Using IoT” which project have consideration of waste management issues been solved by smart bin, interface of GSM and ultrasonic sensor with the help of microcontroller based arduino people get best solution to management of waste this is replacement of traditional dust bin into smart bin one. ARM 7 have been used for controlling Zigbee and global communication, it gives the indication and sending the message using GSM. Sensors are placed in the bin. This bin made and wireless sensor node attached to dustbin send the signal to road side unit real time show status of the bin. Other same signal from RSU reaches the Garbage Collecting Vehicle (GCV) which arrives the particular place to collect garbage. Many technology uses to recycling the garbage. For unhygienic condition people face more problems regarding to health Such situation is control by providing unique ID to garbage bin and identify ID number is given to each can if bin is fill then send SMS to the server. In this project uses microcontroller ATMEGA 16 and certain sensors like PIR sensor, Hall Effect sensor, solar sensor, and LDR sensor. These sensors are connected to microcontroller through an interfacing circuit and an amplifier. The output could view in LCD display, sensor is sense the light and presence sensor sense car or human so light turn on . This project is based on efficient of automatic street lighting system based on low cost microcontroller controlling LED based on street automatically lighting levels control and light sensor, rain sensor, laser sensor and a set of the light emitting diode (LED) have been used brightness in of light will be directly proportional to number of traffic light

Operate like ON or OFF accordingly during night and heavy raining or bad weather.

In Hiransahi Akminand we are using two ultrasonic sensors which sense the level of garbage bin and two gas sensors which detect the harmful gases in the air .This sensors are connected to the avr family microcontroller which is interfaced with LCD display which shows the status of bins .We also used WiFi module which is used to transmit data for webpage applications .We are using one buzzer which gives beep whenever any dustbin is full. The whole system is powered by 12V transformer. Here, we are indicating Four levels Low, Medium, High and Full by using Embedded C programming. In first case when both the dustbins are empty webpage and LCD will display Low level.

Then according to the different levels of garbage it will show Medium, High or Full level on LCD as well as webpage. When any of the dustbin is full it gives beep and when both the dustbins are full it gives loud beep. Along with this the web page and LCD will display the level of harmful gases in the surrounding. In this way Authority can collect the garbage whenever dustbin is full.

In project titled “Bio-hydrogen, bio-methane, bioelectricity as crucial components of biorefinary of organic waste” , the aim of the work was to critically assess selected bioenergy alternatives from organic solid waste, such as biohydrogen and bioelectricity, to evaluate their relative advantages and disadvantages in the context of biorefineries, and finally to indicate the trends for future research and development. Biorefining is the sustainable processing of biomass into spectrum products viz. energy, materials, chemicals, food and

feed. Series systems show a better efficiency than one-stage process regarding substrate conversion to hydrogen and bioenergy

The dark fermentation also produces fermented by-products (fatty acids and solvents), so there is an opportunity for further combining with other processes that yield more bio energy. Photo heterotrophic fermentation is one of them: photosynthetic heterotrophs, such as non-sulphur purple bacteria, can thrive on the simple organic substances produced in dark fermentation and light, to give more H_2. Effluents from photo heterotrophic fermentation and digestives can be processed in microbial fuel cells for bioelectricity production and methanogenic digestion for methane generation, thus integrating a diverse block of bioenergies. Several digestates from bio energies could be used for bio products generation, such as cellulolytic enzymes and saccharification processes, leading to ethanol fermentation (another bioenergy), thus completing the inverse cascade. Finally, biohydrogen and similar products came up to contribute in the improvements for solid organic waste management worldwide.

In project titled, “Municipal Solid Waste Characterization and quantification as a measure towards effective waste management” the aim of the study was to generate a comprehensive data at the regional and national level for use in planning and implementation of relevant waste management activities in Ghana. The study will also assess how well households in three different socioeconomic areas are able to separate their wastes into organic and nonorganic wastes labeled on the bins as biodegradables, except project (food waste, yard waste, wood and manure) and other wastes (project, plastics,

metals, textiles, rubber and leather and any other waste). Sorting and separation of waste using a one way separation system which basically sorted into 'biodegradable (except project)' and all 'other wastes' was tested in this study and the outcome averaged for each study area. From the questionnaire administration results, out of 1000 respondents from all the study areas, 924 (92.4%) were willing to separate their waste while 4.8% were unwilling and 2.3% did not respond. The reason for their willingness to separate waste was because it had the potential for a cleaner environment, it was a good waste management practice and good for recycling but for those not willing to separate waste it was because there was no motivation to do it. Sorting and separation into the correct bins was effective in most of the areas as it averaged above 80% for the "biodegradables except project waste" and above 75% for the "other waste". In the municipality however all the sorting and separation were below 60%. A nationwide average of 84% was obtained for separation into the biodegradable waste bin/bag and 76% for the other waste bin/bag. The high separation efficiency is an indication that the one way separation system employed was convenient for the participating households. This simple sorting and separation system could be recommended for communities learning to separate waste. It is therefore imperative for the MMDAs or city authorities or planners to start rolling out a source sorting process in the various cities. The organic fraction in the waste was the highest in the waste stream and ranged from 48% to 69%.

- Existing problem the garbage shrinks and overflows the garbage bin and spread over the roads and pollutes the environment. The smell will be heavy and produces air

pollution and spreads disease. The street dogs and animals eat the waste food and spreads over the area and creates dirty environment to avoid such situation we are planning to design IOT Based Garbage Management For Smart Cities. Disadvantages of existing system

- Time consuming and less effective: trucks go and empty containers whether theyare full or not.
- High costs.
- Unhygienic Environment and look of the city.
- Bad smell spreads and may cause illness to human beings.
- More traffic and Noise.

- References

Smart City Waste Management System using Internet of Things and Cloud Computing

Aderemi A. Atayero1

, Segun I. Popoola1

, Rotimi Williams1

, Joke A. Badejo1

, and

Sanjay Misra1

1. Municipal Solid Waste Collection Problems: A Literature Review, Jeroen Beliën, Liesje

De Boeck, Jonas Van Ackere

2. Nuortio, T., Kytojoki, J., Niska, H., Braysy, O.: Improved route planning and scheduling

of waste collection and transport. Journal of Expert Systems with Applications 30(2), 223–
232 (2006) CrossRef
3. Zamorano, M., Molero, E., Grindlay, A., Rondriquez, M.L., Hurtado, A., Calvo, and F.J.:

A planning scenario for the application of geographical information systems in municipal waste collection: A case of Churriana de la Vega (Granada, Spain). Journal of Resources,

Conservation and Recycling 54(2), 123–133 (2009) CrossRef
4. Centre of Regional Science. Vienna University of Technology. Smart Cities. Ranking of
European Medium-Sized Cities (accessed on: December 23, 2014
5. Nguyen, T.D., Nguyen, T.T., Misra, S. Cloud-based ERP Solution for Modern Education

in Vietnam. International Conference on Future Data and Security Engineering, FDSE
(2014), pp 234-247.
6. Popoola, S.I., Popoola, O.A., Oluwaranti, A.I., Atayero, A.A., Badejo, J.A., Misra, S. A
Cloud-based Intelligent Toll Collection System for Smart Cities. 3rd International Conference on Next Generation Computing Technologies, NGCT (2017), Dehradun, India, pp 653-663.
7. Odun-Ayo, I., Misra, S., Omoregbe, N., Onibere, E., Bulama, Y., Damasevicius, R. CloudBased Security Driven Human Resource Management System. 8th International Conference on Applications of Digital Information and Web Technologies, ICADIWT (2017),

Mexico, pp 96-106

8. Oduh, I.U., Misra, S., Damasevicius, R., Maskeliunas, R. Cloud based simple employee management information system: A model for African small and medium enterprises. International Conference on Information Technology and Systems, ICITS18 (2017), Libertad, Equador, pp115-128

9. RFID and Integrated Technologies for Solid Waste Bin Monitoring System Maher Arebey,
M A Hannan, Hassan Basri, R A Begum and Huda Abdullah

10. Concept, Design and Implementation of Automatic Waste Management System, Adil
Bashir, Shoaib Amin Banday Ab. Rouf Khan, Mohammad Shafi

11. Smart Garbage Management System Vikrant Bhor1, Pankaj Morajkar2, Maheshwar
Gurav3, Dishant Pandya4

12. Kanchan Mahajan, Chitode, J.S: Waste Bin Monitoring System Using Integrated Technologies

13. Marshall, R.E., Farahbakhsh, K.: Systems approaches to integrated solid waste management in developing countries. Journal of Waste Management 33(4), 988–1003 (2013).

14. Atayero, A.A., Popoola, S.I., Williams, R., Badejo, J.A: Cloud based IoT-enabled solid
waste monitoring system for smart and connected communities.
International Journal of
Civil Engineering and Technology, 2019. 10(2): p. 2308-2315.

- Problem Statement Definition

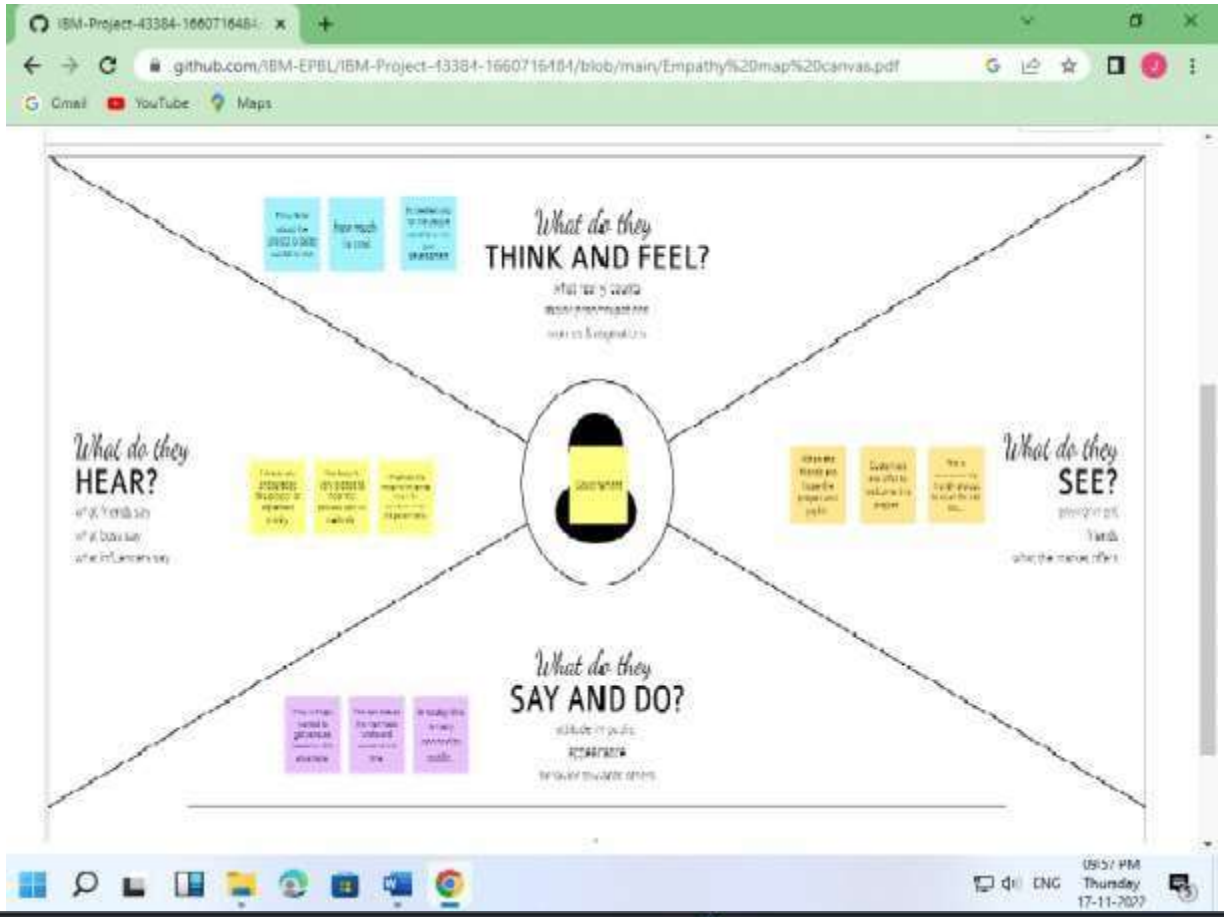
The current process of waste management starts with the waste being created by people in the cities and disposed in trash bins near its creation point. The disposed trash is collected by municipality or private company trucks at the predefined times and transferred to temporary collection centers. The trash at the collection centers is then sent for recycling.

This process in current city setting solves the waste problem partially while it creates other problems such as;

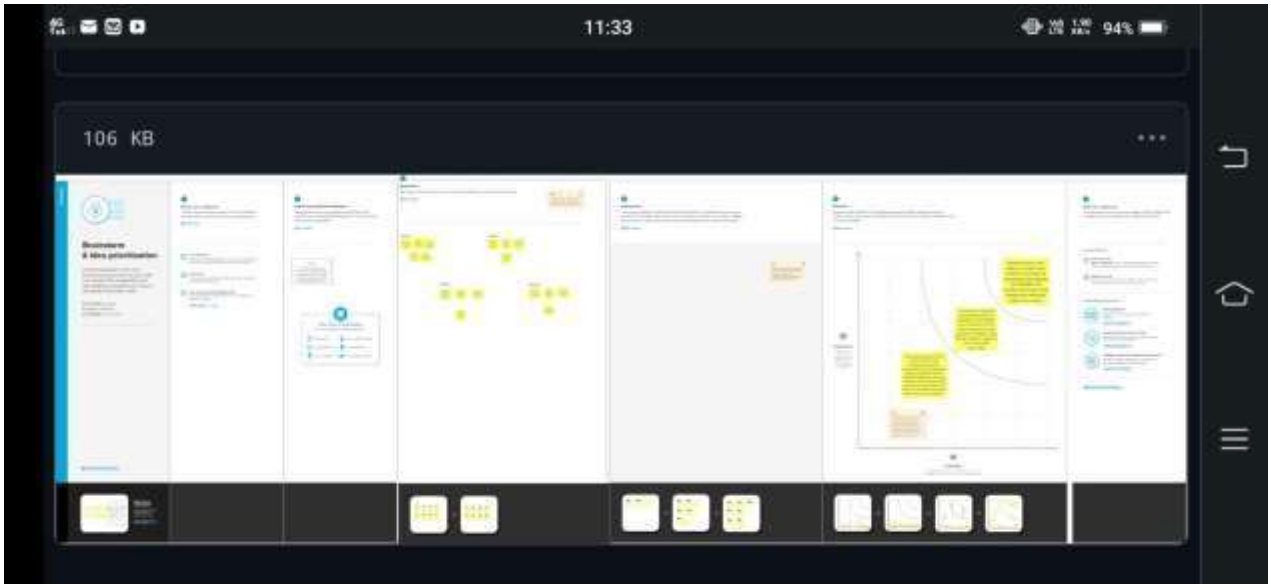
Some trash bins are overfilled while others are underfilled by the trash collection time, overfilled trash bins create unhygienic conditions, unoptimized truck routes result in excessive fuel usage and environmental pollution and all collected trash is combined which complicates sorting at the recycling facility.

Some of these problems can be mitigated by implementing smart waste management systems

- **IDEATION & PROPOSED SOLUTION**
- Empathy Map Canvas



- Ideation & Brainstorming



- Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	O Develop an efficient system and an application that can monitor and alert user.

2.	Idea / Solution description	• This product help the workers in monitoring the garbage. • In several areas the sensor will be integrated to sense the garbage. • If any area the garbage is detected the admins will be notified along with the location.
3.	Novelty / Uniqueness	O Fastest alerts to the workers. O User friendly.
4.	Social Impact / Customer Satisfaction	• Cost efficient. • East installation and provide efficient result.
5.	Business Model (Revenue Model)	O The product is advertised all over the platforms. Since it is economical, even health society from pollution.
6.	Scalability of the Solution	• Since the product is cost efficient, it can be placed in many places in the area. • Even when the garbage is more is more, the product sense the accurate values and alerts the workers effectively

- Problem Solution fit

The main problem of customers is improper maintenance of the garbage bins.

It leads to various problems like unhygienic environment , soil pollution and etc.

- **REQUIREMENT ANALYSIS**

- **Functional Requirements:**

Following are the functional requirements of the proposed solution. **FR No.**	**Functional Requirement (Epic)**	**Sub Requirement (Story / Sub-Task)**
FR-1	Detailed bin inventory.	All monitored bins and stands can be seen on the map, and
		you can visit them at any time via the Street View feature from Google. Bins or stands are visible on the map as green, orange or red circles. You can see bin details in the Dashboard – capacity, waste type, last measurement, GPS location and collection schedule or pick recognition.

FR-2	Real time bin monitoring.	The Dashboard displays realtime data on fill-levels of bins monitored by smart sensors. In addition to the % of filllevel, based on the historical data, the tool predicts when the bin will become full, one of the functionalities that are not included even in the best waste management software.. Sensors recognize picks as well; so you can check when the bin was last collected. With real-time data and predictions, you can eliminate the overflowing bins and stop collecting half-empty ones.
FR-3	Expensive bins.	The cost of the proposed bin is very efficient and it will not
		have any expensive many components of hardware in first place.

FR-4	Adjust bin distribution.	Ensure the most optimal distribution of bins. Identify areas with either dense or sparse bin distribution. Make sure all trash types are represented within a stand. Based on the historical data, you can adjust bin capacity or location where necessary.
FR-5	Eliminate unefficient picks.	Eliminate the collection of halfempty bins. The sensors recognize picks. By using real-time data on filllevels and pick recognition, we can show you how full the bins you collect are.

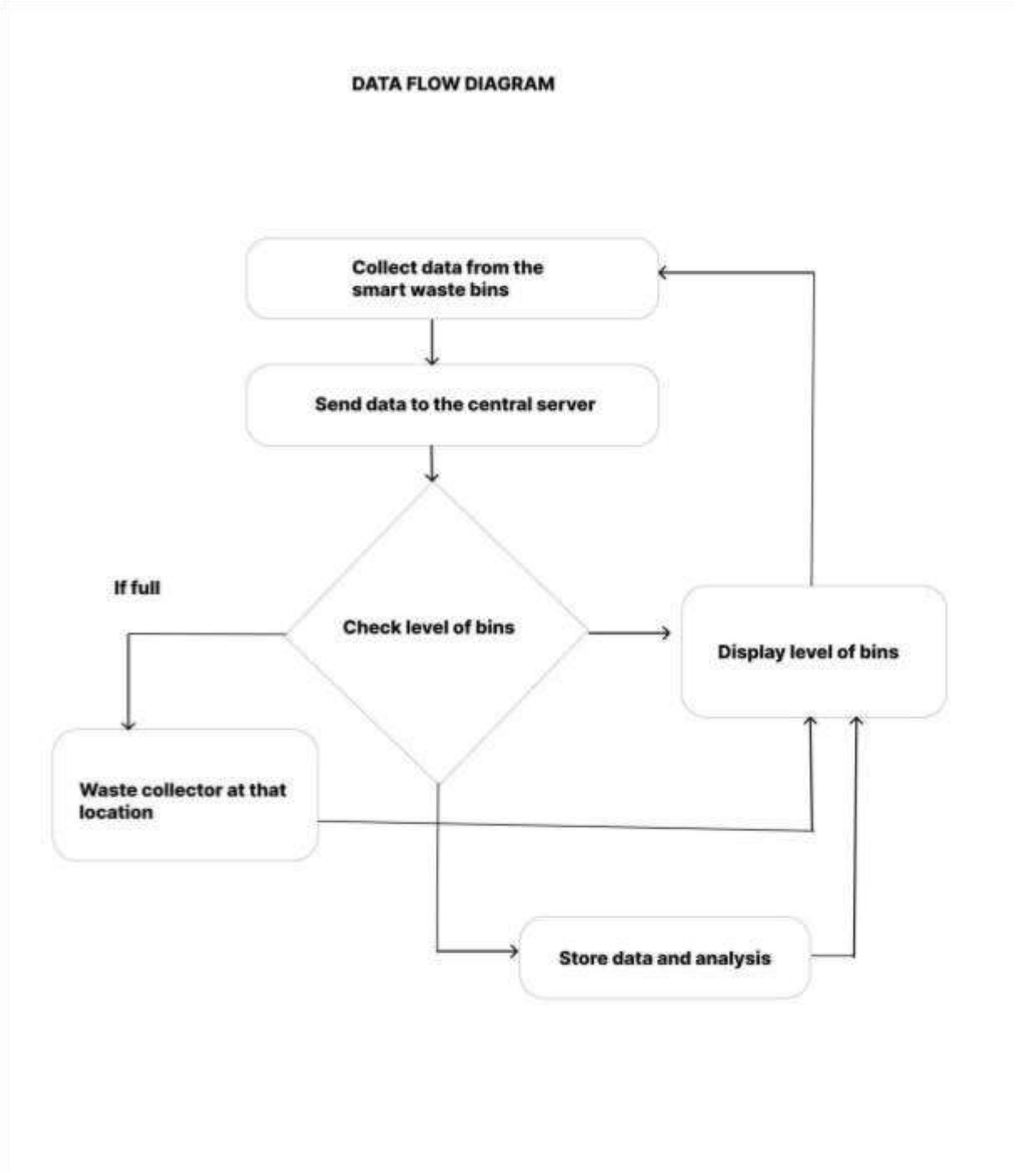
- Non-Functional requirements
- **Non-functional Requirements:**

Following are the nonfunctional requirements of the proposed solution. **FR No.**	**Non-Functional Requirement**	**Description**
NFR-1	**Usability**	IoT device verifies that usability is a special and important perspective to analyze user requirements, which can further improve the design quality. In the design

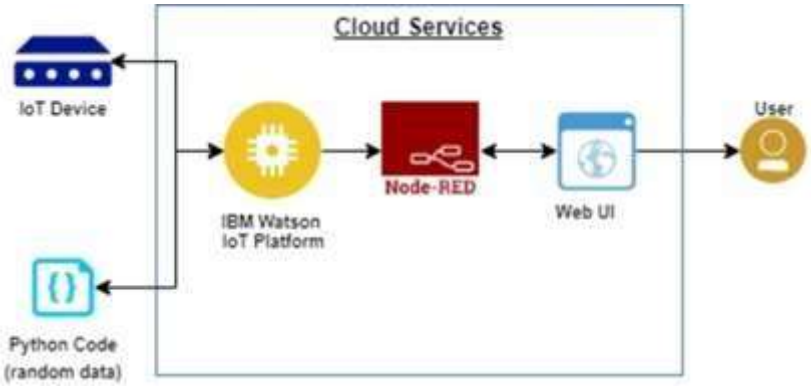
		process with user experience as the core, the analysis of users' product usability can indeed help designers better understand users' potential needs in waste management, behavior and experience.
NFR-2	**Security**	Use a reusable bottles Use reusable grocery bags Purchase wisely and recycle Avoid single use food and drink containers.
NFR-3	**Reliability**	Smart waste management is also about creating better working conditions for waste collectors and drivers. Instead of driving the same collection routes and servicing empty bins, waste collectors will spend their time more efficiently, taking care of bins that need servicing.
NFR-4	**Performance**	The Smart Sensors use ultrasound technology to measure the fill levels (along with other data) in bins several times a day. Using a variety of IoT networks ((NB-IoT,GPRS), the sensors send the data to

		Sensoneo's Smart Waste Management Software System, a powerful cloudbased platform, for data- driven daily operations, available also as a waste management app. Customers are hence provided data-driven decision making, and optimization of waste collection routes, frequencies, and vehicle loads resulting in route reduction by at least 30%.
NFR-5	**Availability**	By developing & deploying resilient hardware and beautiful software we empower cities, businesses, and countries to manage waste smarter.
NFR-6	**Scalability**	Using smart waste bins reduce the number of bins inside town , cities coz we able to monitor the

- **PROJECT DESIGN**
- Data Flow Diagrams

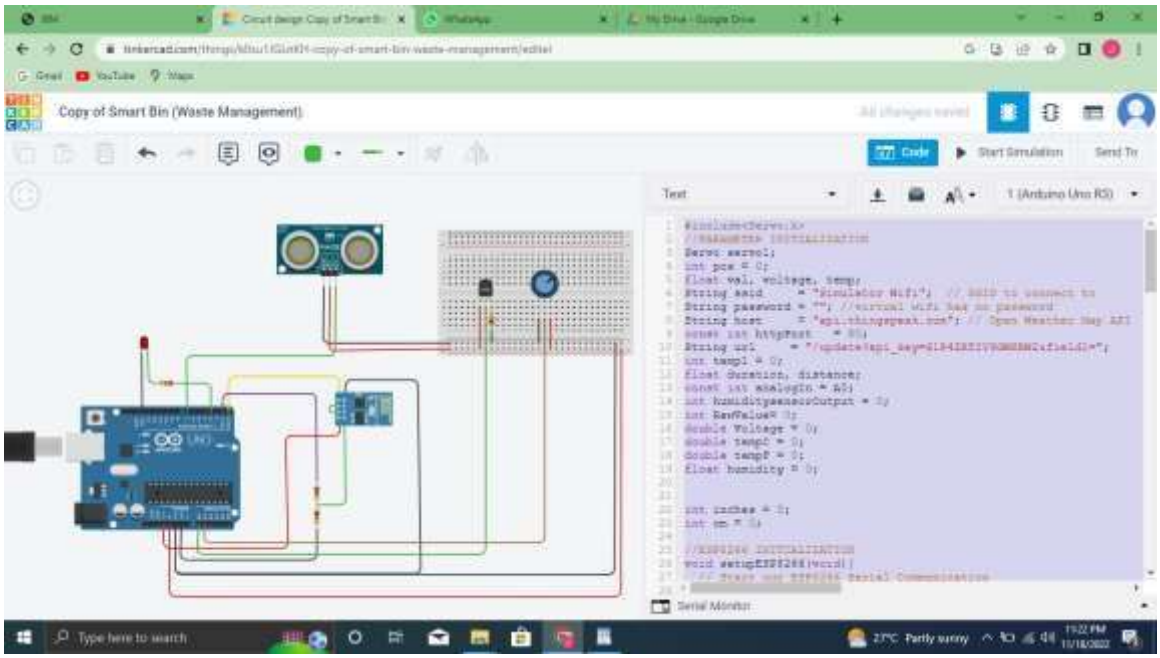


- Solution & Technical Architecture



- User Stories

- **PROJECT PLANNING & SCHEDULING**
- Sprint Planning



- 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	User Story	User Story / Task	Story Points	Priority	Team Members

	(Epic)	Number				
Sprint-1	Login	USN-1	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	Jacobstanly
Sprint-2	Dashboard	USN-2	As a Truck Driver, I'll follow Co-Admin's Instruction to reach the filling bin in short roots and save time	20	Low	Gowtham
Sprint-3	Dashboard	USN-3	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	Jayasharma
Sprint-4	Dashboard	USN-4	As a Municipality officer, I'll make sure everything is proceeding as planned and without any problems	20	High	Syed Hussain

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

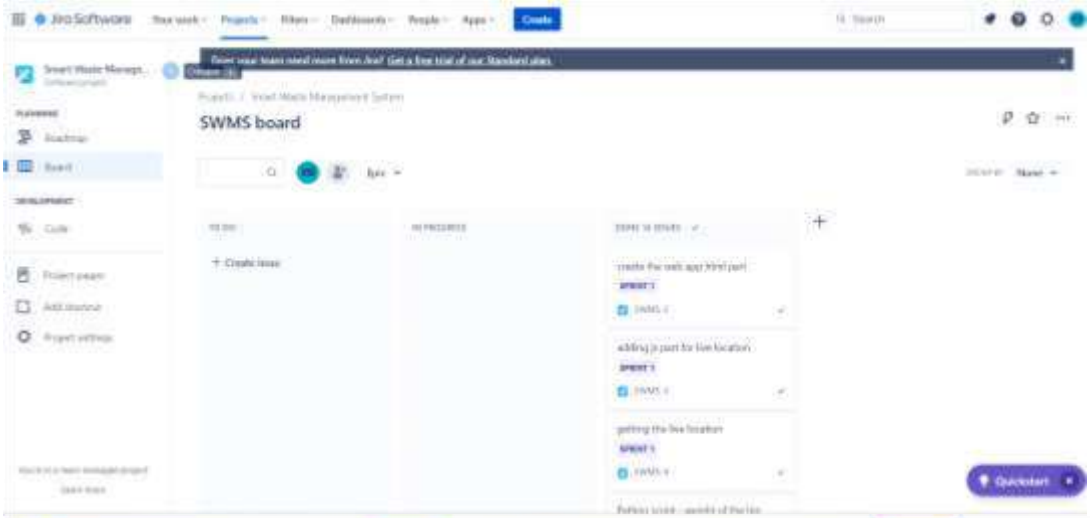
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022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

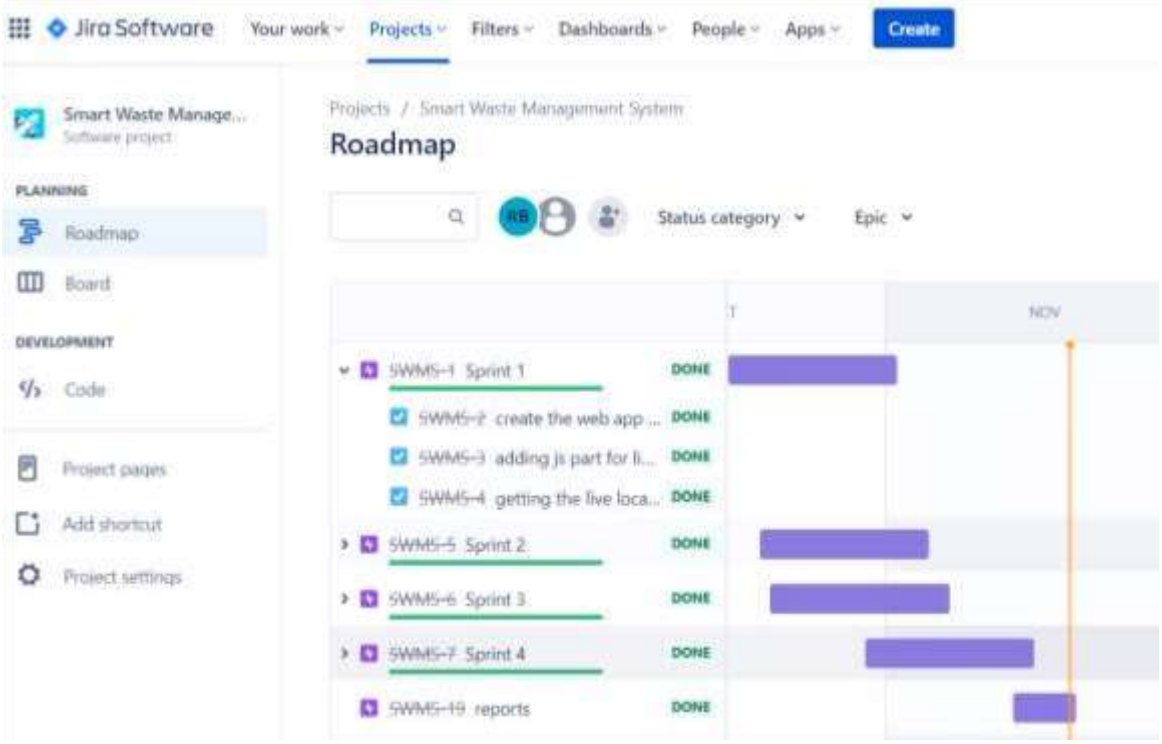
Velocity:

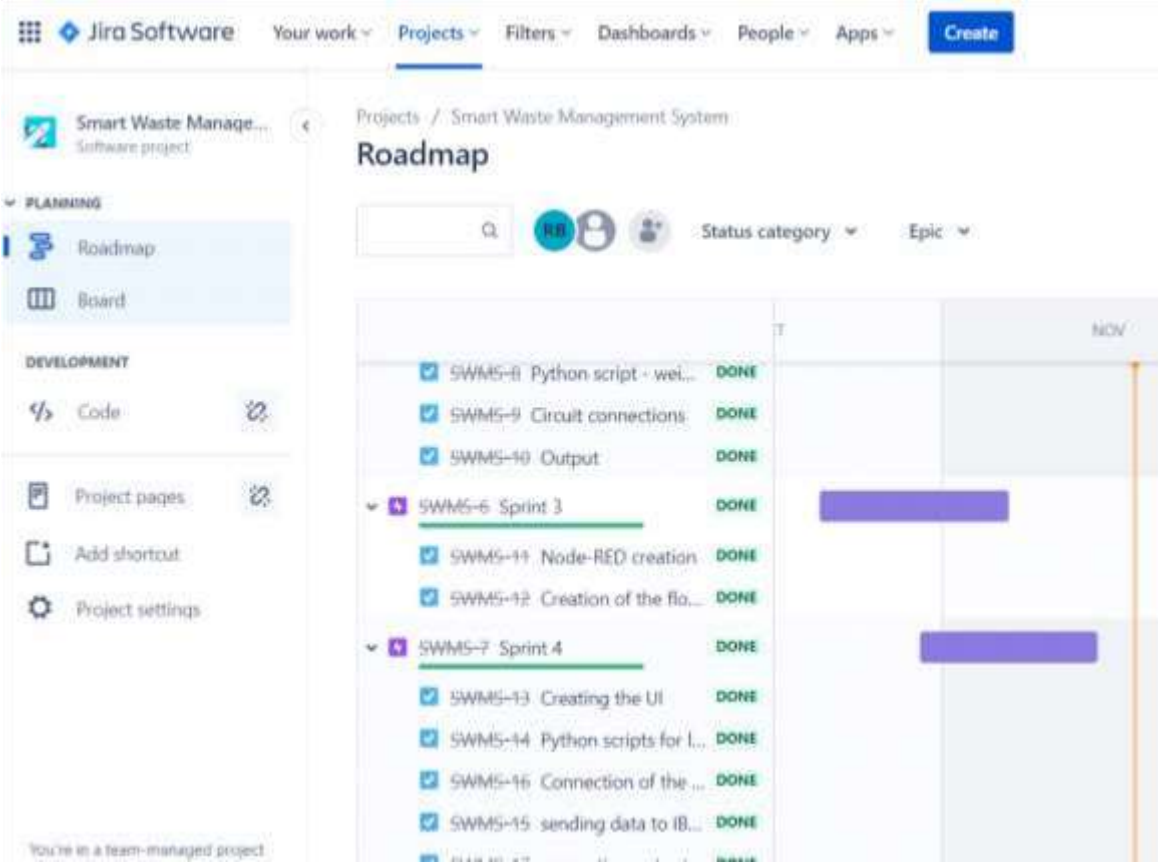
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

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

- Reports from JIRA


SWMS board


Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Smart Waste Manage...
Software project
PLANNING
Roadmap
Board
DEVELOPMENT
Code
Project pages
Add shortcut
Project settings
Projects / Smart Waste Management System
Roadmap
Status category
Epic
NOV
SWMS-1 Sprint 1 DONE
SWMS-2 create the web app ... DONE
SWMS-3 adding js part for li... DONE
SWMS-4 getting the live loca... DONE
SWMS-5 Sprint 2 DONE
SWMS-6 Sprint 3 DONE
SWMS-7 Sprint 4 DONE
SWMS-19 reports DONE



- **CODING & SOLUTIONING (Explain the features added in the project along with code) •** Feature 1

The main and first feature of the smart waste management is to get the live location of anyone who access the website for putting out a request for garbage collection in their locality. The live location is obtained as a result of the below code.

Web Application to get the Live location:

index.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/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
   <meta charset="utf-8">
   <meta name="viewport" content="width=device-width">
   <title>Smart Waste Management System</title>
   <link rel="icon" type="image/x-icon" href="/imgs/DUMPSTER.png">
   <link href="style.css" rel="stylesheet" type="text/css" />
   <script src="https://www.gstatic.com/firebasejs/9.14.0/firebase-app.js"></script>
   <script src="https://www.gstatic.com/firebasejs/9.14.0/firebase-analytics.js"></script>
<script src="https://www.gstatic.com/firebasejs/9.14.0/firebase-database.js"></script>

   <script>
      var firebaseConfig =
      {
         apiKey: "AIzaSyCcZk7b1CLOGviwUpthRDLotrmFX0MFuTs",
         authDomain: "swms-3840.firebaseapp.com",
projectId: "swms-3840",          storageBucket:
"swms-3840.appspot.com",
messagingSenderId: "479902726304",
         appId: "1:479902726304:web:3d822880d1275ee57a71c5",
measurementId: "G-MHP4N77MTP"
      };
      firebase.initializeApp(firebaseConfig)
   </script>
   <script defer src="db.js"></script>
</head>

<body style="background-color:#1F1B24;">
   <script src="maps.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
btndark">
         DUMPSTER
      </a>
   </center>
   <script

src="https://maps.googleapis.com/maps/api/js?key=AIzaSyBBLyWj3FWtCbCXGW3ysEiI2
fDfrv2v0Q&callback=myMap"></script></div>
</body>
```

```
</html>
```

db.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);
 }
);
```

maps.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);
      }
    );
  }
```

- Feature 2

In this part, the filled level of the bin is measured with the help of IBM IOT Watson platform devices, IBM Cloud interface and Node-RED is used for creating the dashboard nodes that helps us create a UI to display the distance, that is, the fill level of the bin. It also intimates the location of the bin with the fill level and alerts the collection authority if the fill level goes beyond a threshold value.

Code to evaluate the level of the garbage in bin:

bin1.py:

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

# watson device details
organization = "73ffyv"
devicType =    "BIN1"
deviceId =  "BIN1ID"
authMethod= "token"
authToken= "123456789"

#generate random values for randomo variables (temperature&humidity) def
myCommandCallback(cmd):
   global a
   print("command recieved is:%s" %cmd.data['command'])
control=cmd.data['command']
   print(control)

try:
   deviceOptions={"org":      organization,   "type":          devicType,"id":
         deviceId,"authmethod":authMethod,"auth-token":authToken}     deviceCli =
```

```
ibmiotf.device.Client(deviceOptions) except Exception as e:     print("Exception while
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:' 'Garbage level is high, collection time :) 90 %'            elif
   distance < 40 and distance >16:
        dist = 'Risk warning:' 'garbage 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 :' '
Garbage level is high, collection time :)'              elif load == "60 %"
or distance == "60 %":          warn = 'alert :' 'garbage is above 60%'
else :
        warn = 'alert :' 'Levels are low, collection not needed '

   def myOnPublishCallback(lat=11.035081,long=77.014616):
      print("Peelamedu, Coimbatore")
      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()
```

bin2.py:

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

# watson device details
organization  =  "73ffyv"
devicType   =     "BIN2"
deviceId    =    "BIN2ID"
authMethod= "token"
authToken= "123456789"

#generate random values for randomo variables (temperature&humidity) def
myCommandCallback(cmd):
   global a
   print("command recieved is:%s" %cmd.data['command'])
control=cmd.data['command']
   print(control)

try:
   deviceOptions={"org":       organization,    "type":            devicType,"id":
         deviceId,"authmethod":authMethod,"auth-token":authToken}      deviceCli =
ibmiotf.device.Client(deviceOptions) except Exception as e:      print("Exception while
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:' 'Garbage level is high, collection
time :) 90 %'               elif distance < 40 and distance >16:
        dist = 'Risk warning:' 'garbage 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 :' '
Garbage level is high, collection time :)'              elif load == "60 %"
or distance == "60 %":          warn = 'alert :' 'garbage is above 60%'
else :
        warn = 'alert :' 'Levels are low, collection not needed '

   def myOnPublishCallback(lat=11.068774,long=77.092978):
      print("PSG iTech, Coimbatore")
      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.py:

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

# watson device details organization
= "73ffyv" devicType =  "BIN3"
deviceId = "BIN3ID" authMethod=
"token"
authToken= "123456789"

#generate random values for randomo variables (temperature&humidity) def
myCommandCallback(cmd):
   global a
   print("command recieved is:%s" %cmd.data['command'])
control=cmd.data['command']
   print(control)

try:
   deviceOptions={"org":       organization,   "type":            devicType,"id":
         deviceId,"authmethod":authMethod,"auth-token":authToken}     deviceCli =
ibmiotf.device.Client(deviceOptions) except Exception as e:     print("Exception while
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:' 'Garbage level is high, collection
time :) 90 %'               elif distance < 40 and distance >16:
        dist = 'Risk warning:' 'garbage 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 :' '
Garbage level is high, collection time :)'              elif load == "60 %"
or distance == "60 %":          warn = 'alert :' 'garbage is above 60%'
else :
        warn = 'alert :' 'Levels are low, collection not needed '

   def myOnPublishCallback(lat=11.007403,long=76.963439):
      print("Kattoor, Coimbatore")
      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()
```

bin4.py:

```
import requests  import
json
```

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

# watson device details
organization = "73ffyv"
devicType = "BIN4"
deviceId = "BIN4ID"
authMethod= "token"
authToken= "123456789"

#generate random values for randomo variables (temperature&humidity) def
myCommandCallback(cmd):
   global a
   print("command recieved is:%s" %cmd.data['command'])
control=cmd.data['command']
   print(control)

try:
   deviceOptions={"org": organization, "type": devicType,"id":
        deviceId,"authmethod":authMethod,"auth-token":authToken}    deviceCli =
ibmiotf.device.Client(deviceOptions) except Exception as e:    print("Exception while
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:' 'Garbage level is high, collection time :) 90 %'
elif distance < 40 and distance >16:
        dist = 'Risk warning:' 'garbage 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 :' '
Garbage level is high, collection time :)'              elif load == "60 %"
or distance == "60 %":          warn = 'alert :' 'garbage is above 60%'
else :
        warn = 'alert :' 'Levels are low, collection not needed '

   def myOnPublishCallback(lat=11.453306,long=77.426024):
print("Seethammal Colony, Gobichittipalayam")
       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()
```

- **TESTING**
- Test Cases

Unit testing

Test case no.	Sensor/Stage	Input	Expected output	Obtained output	Status
1.	Ultrasonic	Garbage level in bin	Correct level or distance	As expected	Pass
		i)Null ii)Full iii)Range in %			
2.	AURDINO	Microcontroller to process the input data	To collect the data from sensor	As expected	Pass

- **ACCEPTANCE**

Acceptance testing - is the final phase of product testing prior to public launch. A level of the software testing process where a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery

- **ADVANTAGES AND DISADVANTAGES**

Benefits or advantages of Smart Waste Management

Following are the benefits or **advantages of Smart Waste Management**: ➡It saves time and money by using smart waste collection bins and systems equipped with fill level sensors. As smart transport vehicles go only to the filled

containers or bins. It reduces infrastructure, operating and maintenance costs by upto 30%.

➡It decreases traffic flow and consecutively noise due to less air pollution as result of less waste collection vehicles on the roads. This has become possible due to two way communication between smart dustbins and service operators.

➡It keeps our surroundings clean and green and free from bad odour of wastes, emphasizes on healthy environment and keep cities more beautiful.

➡It further reduces manpower requirements to handle the garbage collection process.

➡Applying smart waste management process to the city optimizes management, resources and costs which makes it a "smart city".

➡It helps administration to generate extra revenue by advertisements on smart devices.

Real time information on the fill level of the dustbin.

· Deployment of dustbin based on the actua|

needs.

· Cost Reduction and resource optimization.

· Improves Environment quality -Fewer smells -

Cleaner cities

· Intelligent management of the services in the

city.

· Effective u

Drawbacks or disadvantages of Smart Waste Management

Following are the drawbacks or **disadvantages of Smart Waste Management**:

➡System requires more number of waste bins for separate waste collection as per

population in the city. This results into high initial cost due to expensive smart

dustbins compare to other methods.

➡Sensor nodes used in the dustbins have limited memory size.

➡Wireless technologies used in the system such as zigbee and wifi have shorter range and lower data speed. In RFID based systems, RFID tags are affected by surrounding metal objects (if any).

➡It reduces man power requirements which results into increase in unemployments for unskilled people.

➡The trainining has to be provided to the people involved in the smart waste management system.

- **CONCLUSION**

Due to the absence of sustainable waste management technology, the current waste disposal situation is likely to worsen.This work presents an enhanced solution to the problem of waste management by the littering of the garbage bins once they are full.Littering of the environment and the health hazards are minimized as timely disposal of the wastes is ensured as the system automatically sends a message alert to the garbage collector or the management authority once the bin is full thereby ensuring that the bin is made empty to avoid dumping of refuse on the floor

- **FUTURE SCOPE**

Based on the real-time and historical data collected and stored in the cloud waste collection schedules and routes can be optimized. Predictive analytics could be used to make decisions ahead of time and offers insight into waste bin locations.

Graph theory optimization algorithms can be used to manage waste collection strategies dynamically and efficiently. Every day, the workers can receive the newly calculated routes in their navigation devices. The system can be designed to learn from experience and to make decisions not only on the daily waste level status but also on future state forecast, traffic congestion, balanced cost-efficiency functions, and other affecting factors that a priori humans cannot foresee.

Garbage collectors could access the application on their mobile phone/tablets using the internet. Real-time GPS assistance can be used to direct them to the pre-decided route. As they go collecting the garbage from the containers, the management is also aware of the progress as the vehicle, as well as the garbage containers, are traced in real-time. The management staff gets their own personalized administration panel over a computer/tablet which gives them a bird eye view over the entire operations.

An alternative solution using image processing and camera as a passive sensor could be used. But, the cost of those image processing cameras is higher as compared to the ultrasonic sensors, which leads to high solution implementation cost.

1. **Time can be reducing**
2. **very futurised technique**
3. **wanted method**

- **APPENDIX**

Source Code:

```
#include<Servo.h>
//PARAMETER INITIALIZATION
```

```
Servo servo1; int pos =
0; float val, voltage,
temp;

String ssid     = "Simulator Wifi";  // SSID to connect to
String password = ""; //virtual wifi has no password  String host
= "api.thingspeak.com"; // Open Weather Map API const int
httpPort   = 80;

String url     = "/update?api_key=S1P4ZR5IY9GWKRW2&field1=";
int temp1 = 0; float duration, distance; const int analogIn = A0; int
humiditysensorOutput = 0; int RawValue= 0; double Voltage = 0;
double tempC = 0; double tempF = 0; float humidity = 0;



int inches = 0; int
cm = 0;


//ESP8266 INITIALIZATION void
setupESP8266(void){

  // Start our ESP8266 Serial Communication
  Serial.begin(115200);   // Serial connection over USB to computer
Serial.println("AT");   // Serial connection on Tx / Rx port to ESP8266
delay(10);        // Wait a little for the ESP to respond   if
(Serial.find("OK"))     Serial.println("ESP8266 OK!!!");

    // Connect to Simulator Wifi
      Serial.println("AT+CWJAP=\"" + ssid + "\",\"" + password + "\"");
      delay(10);        // Wait a little for the ESP to respond   if
(Serial.find("OK"))
    Serial.println("Connected to WiFi!!!");
      // Open TCP connection to the host:

      //ESP8266 connects to the server as a TCP client.
```

```
    Serial.println("AT+CIPSTART=\"TCP\",\"" + host + "\"," + httpPort);
    delay(50);       // Wait a little for the ESP to respond   if
(Serial.find("OK"))
    Serial.println("ESP8266 Connected to server!!!") ;
}

// For Ultrasonics long readUltrasonicDistance(int
triggerPin, int echoPin)
{   pinMode(triggerPin, OUTPUT);  // Clear the
trigger   digitalWrite(triggerPin, LOW);
delayMicroseconds(2);
 // Sets the trigger pin to HIGH state for 10 microseconds
digitalWrite(triggerPin, HIGH);   delayMicroseconds(10);
digitalWrite(triggerPin, LOW);   pinMode(echoPin,
INPUT);
 // Reads the echo pin, and returns the sound wave travel time in microseconds
return pulseIn(echoPin, HIGH);
}

//VOID SETUP void
setup()
{
 Serial.begin(9600);
pinMode(A1, INPUT);
setupESP8266();
servo1.attach(13);   pinMode(3,
OUTPUT);
}

//VOID LOOP
void loop()  {
anydata();
delay(7000);
```

```
}


//SENSOR DATA PROCESS AND TRANSMIT
void anydata(void)  {
 //Ultrasonic Sensor   // measure the ping time
in cm   cm = 0.01723 *
readUltrasonicDistance(7, 7);   // convert to
inches by dividing by 2.54   inches = (cm /
2.54);   delay(100); // Wait for 100
millisecond(s)


 // LED    if
(inches <= 10)
 {
 // turn the LED on (HIGH is the voltage level)
digitalWrite(3, HIGH);
 }   if (inches >
10)
 {
 // turn the LED off (LOW is the voltage level)
digitalWrite(3, LOW);
 }


 // Humidity    humiditysensorOutput =
analogRead(A1);


 //Temperature Sensor
 val=analogRead(A0);
voltage=val*0.0048828125;    temp
= (voltage - 0.5) * 100.0;
```

```
  //Display the data
  Serial.print("\nDistance(inches): ");
  Serial.print(inches);
  Serial.println("in");
  Serial.print("Distance(cm): ");
  Serial.print(cm);
  Serial.println("cm");
  Serial.print("Temperature: ");
  Serial.println(temp);
  Serial.print("Humidity : ");
  Serial.print(map(humiditysensorOutput, 0, 1023, 10, 70));
  Serial.println("%");
  Serial.println(" ");


  // Construct our HTTP call
 if (temp1 == 15){
     temp1 = 0;

  }
  String httpPacket = "GET " + url + String(temp)
    + "&field2=" +
String(map(humiditysensorOutput, 0, 1023, 10, 70)) + "&field3=" + String(inches) + "
HTTP/1.1\r\nHost: " + host + "\r\n\r\n";
  temp1 = temp1 + 1;   int length =
  httpPacket.length();

  // Send our message length
  Serial.print(String(temp) +String(humidity) +String(humidity));
  Serial.print("AT+CIPSEND=");
 Serial.println(length);

  delay(10); // Wait a little for the ESP to respond if (!Serial.find(">")) return -1;
  // Send our http request
  Serial.print(httpPacket);
  delay(10); // Wait a little for the ESP to respond
 if (Serial.find("SEND OK\r\n"))
```

```
    Serial.println("ESP8266 sends data to the server");
      Serial.println(" ");
}
```

GitHub & Project Demo Link: https://drive.google.com/file/d/1nB3FkUBl4YsOu_gD8tEIzW1uDnSXxVL0/view?usp=drivesdk