

Sprint Delivery – 1

Date	09 November 2022
Team ID	PNT2022TMID43416
Project Name	Project - SMART WASTE MANAGEMENT SYSTEM

1. Introduction

Today big cities around the world are facing a common problem, managing the city waste effectively without making the city unclean. Today's waste management systems involve a large number of employees being appointed to attend a certain number of dumpsters; this is done every day periodically. This leads to a very inefficient and unclean system in which some dumpsters will be overflowing and some dumpsters might not be even half full. This is caused by variation in population density in the city or some other random factor that makes it impossible to determine which part needs immediate attention. Here a waste management system is introduced in which each dumpster is embedded in a monitoring system that will notify the corresponding personnel if the dumpster is full.

2. Problem Statement

An inefficient waste management may create serious environmental impacts like infectious diseases, land and water pollution, and climate changes. In this design smart waste management is used for proper disposal and efficient collection of waste by using a Web application.

- Garbage level detection in bins.
- Getting the weight of the garbage in the bin.
- Alerts the authorized person to empty the bin whenever the bins are full.
- Garbage level of the bins can be monitored through a web App.
- We can view the location of every bin in the web application by sending GPS location from the device.

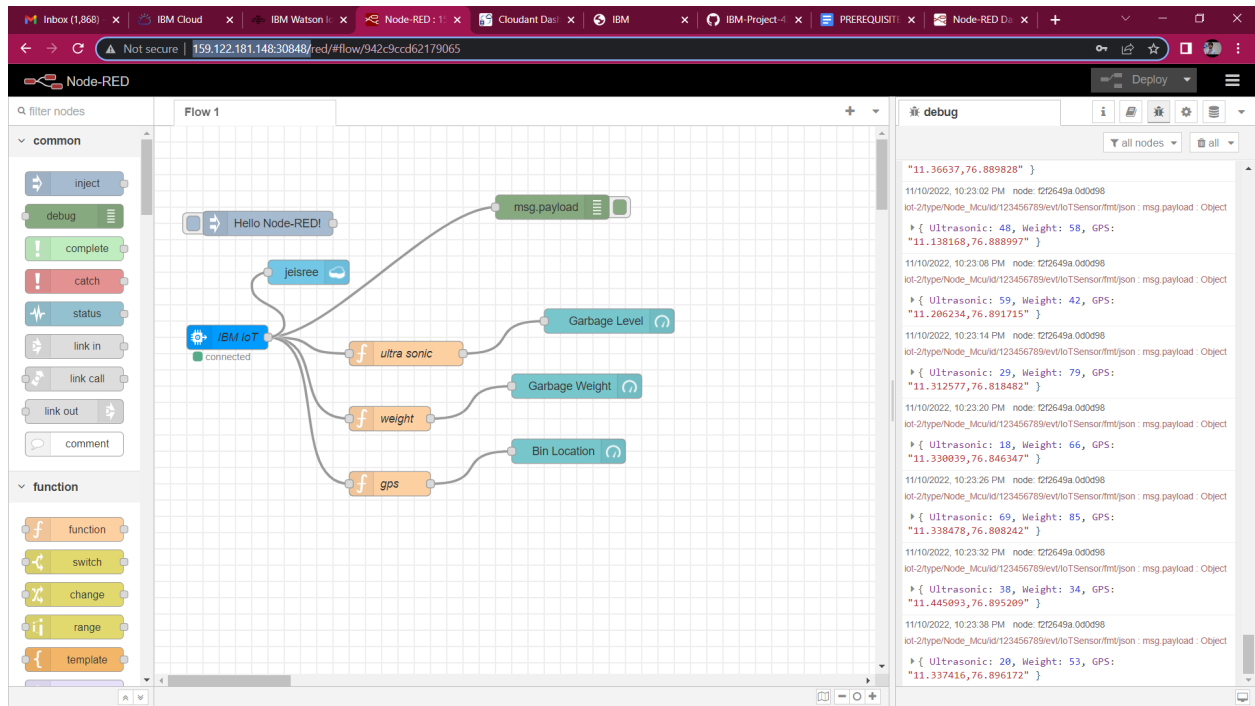
3. Proposed Solution

The main aim of this project is to create a web application to monitor the status of any bin and view its location through a Web App. Once the garbage bin is full. The alert message is sent to the authorized person to empty the bin.

4. Required Software Installation

Node-Red

Node-RED is a flow-based development tool for visual programming developed originally by IBM for wiring together hardware devices, APIs and online services as part of the Internet of Things. Node-RED provides a web browser-based flow editor, which can be used to create JavaScript functions.



Installation of IBM IoT and Dashboard nodes for Node-Red :

IBM Watson IoT Platform

Browse Action Device Types Interfaces

Add Device +

Browse Devices

All Devices Diagnose

This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.

Search by Device ID

Device Simulator

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
123456789	Disconnected	Node_Mcu	Device	Nov 9, 2022 8:21 PM	

Items per page 50 | 1-1 of 1 item

1 of 1 page

Python IDE:

```
iot python.py - D:\IBM\iot python.py (3.7.0)
File Edit Format Run Options Window Help

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

#Provide your IBM Watson Device Credentials
organization = "8wd932"
deviceType = "Node_Mcu"
deviceId = "123456789"
authMethod = "token"
authToken = "123456789"

# Initialize GPIO
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="lighton":
        print("led in on")
    else :
        print ("led is off")

try:
    deviceOptions = {"org": organization, "type": deviceType, "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 "hello" with value "world" into the cloud as an event of type "greeting" 10 times
deviceCli.connect()

while True:
    #Get Sensor Data from DHT11
    time.sleep(5)
    ult_son=random.randint(0,80)
    weight=random.randint(0,100)
    lat = round(random.uniform(11.03, 11.50), 6)
```