**PROJECT**

**SMART WASTE MANAGEMENT SYSTEM IN METROPOLITAN CITIES USING IOT**

**Team members :-**

**P. Pooja Sree - 19R11A04H8**

**P. Sruthi - 19R11A04H9**

**P. Kathyayani - 19R11A04J1**

**R. Prerna - 19R11A04J2**

**T. Vasavi sri lakshmi - 19R11A04J9**

**DOCUMENTATION**

**INTRODUCTION**

**OVERVIEWWaste management is all the activities and actions required to manage waste from its inception to its final disposal . This includes collection, transportation, treatment and disposal of waste together with monitoring and regulation. Waste collection methods vary widely among different countries and regions. Domestic waste collection services are often provided by local government authorities. Curbside collection is the most common method of disposal in most countries, in which waste is collected at regular intervals by specialized trucks. Waste collected is then transported to an appropriate disposal area. Nowadays, cities with developing economies experience exhausted waste collection services, inadequately managed and uncontrolled dumpsites and the problems are worsening. Waste collection method in such countries is an on-going challenge and many struggle due to weak institutions and rapid urbanization.**

**PURPOSE**

**1. By 2030, almost two-third of the world’s population will be living in cities. This fact requires the development of sustainable solutions for urban life, managing waste is a key issue for the health.**

**2. Efficient and energy-saving waste management, reducing CO2,air pollution and vehicle exhaust emissions—these are just a few examples for the demands of future cities. In views of that, the efficient use and responsible handling of resources become more important.**

**3. Effectively managing waste is important in developed countries. Waste management may swallow upto 50% of a city’s budget, but only serve a small part of the population.**

**4. Sometimes, upto 60%of waste is not being collected, it is often simply burned by the roadside. It can pollute drinking water, it can spread disease to people living nearby.**

**5. Even with great route optimization, the worker must still physically go to the dustbin to check waste levels. Because of this, trucks often visit containers that do not need emptying, which wastes both time and fuel.**

**6. Waste management prevents harm to human health and the environment by reducing the volume and hazardous character of residential and industrial waste.**

**7. Improving proper waste management will reduce pollution, recycle useful materials and create more green energy.**

**LITERATURE SURVEY**

**INTRODUCTIONAn inevitable consequence of development and industrial progress is generation of waste. Therefore, efficient waste management is a matter of international concern and countries have setup robust regulatory waste management regimes for balancing the objectives of development and environment sustainability. In India, the national environment policy, 2006 while suggesting measures for collection of wastes and safe disposal of residues .The metro cities and major economic hubs generate the maximum volume of waste, but a survey of 20 smaller cities selected to be developed as smart cities show that most are struggling to manage waste. So, there should be an improvement in the waste management techniques.**

**HISTORYFollowing the onset of industrialisation and the sustained urban growth of large population centres, the build-up of waste in the cities caused a rapid deterioration in levels of sanitation and the general quality of urban life. The streets became choked with filth due to the lack of waste clearance regulations .In the UK, London, The Metropolitan Board of Works was the first city-wide authority that centralized sanitation regulation for the rapidly expanding city and the Public Health Act 1875 made it compulsory for every household to deposit their weekly waste in "moveable receptacles: for disposal—the first concept for a dust-bin .Early garbage removal trucks were simply open bodied dump trucks pulled by a team of horses. They became motorized in the early part of the 20th century and the first closed body trucks to eliminate odours with a dumping lever mechanism were introduced in the 1920s in Britain.**

**EXISTING PROBLEM**

**Solid waste is a great threat not only to the economy of any country but for the environment too. The public through various sources generate tons of solid waste regularly. In the era of globalization, one of the rising issues of developing and under developed countries is handling such huge masses of solid waste. Hyderabad is the 2nd largest city of Sindh and 6th in Pakistan. Unfortunately, it does not poses proper solid waste management system right from collection up to its proper disposal. Most of those uncollected wastage poses a high risk to the public through blockage of drains and formation of stagnant ponds which provide a breeding ground for flies and mosquitoes with a high risk of diseases**

**ABOUT HYDERABAD CITY**

**The total waste produced in Hyderabad city per day is 3000 ton and per year 10950000 tons. As per recent estimates, the municipal waste generation in metro cities varies between 0.2-0.6 kg/capita/day and urban MSW generation is estimated to be approximately 0.49kg per capita per day.**

**PROPOSED SOLUTION**

**1. The smart, sensor based dustbin will judge the level of waste in it and send the message directly to the municipal corporation.**

**2. According to the filled level of the dustbin, the vehicles from the municipal corporation will choose the shortest path with the help of the “TRANSPORTATION SOFTWARE”, which will save their time.**

**THEORETICAL ANALYSIS**

**BLOCK DIAGRAM**

Garbage Container

Arduino Board

Ultrasonic and Weight sensor

**FLOWCHART**

Municipal Corporation

Software application

Mobile communication Network

**No**

**Send message to the control room**

**Send cleaning vehicle for cleaning**

**Update status of dustbin**

**Check whether the dustbin is full**

**HARDWARE AND SOFTWARE DESIGNING**

**Yes**

**Arduino Uno:**

**Arduino Uno is a micro controller board. It has 14 digital input/ output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the micro controller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.. You can tinker with your UNO without worrying too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again.**

**Ultrasonic Sensor:**

**The Ultrasonic Sensor sends out a high-frequency sound pulse and then times how long it takes for the echo of the sound to reflect back. The sensor has 2 openings on its front. One opening transmits ultrasonic waves, (like a tiny speaker), the other receives them, (like a tiny microphone).The speed of sound is approximately 341 meters (1100 feet) per second in air. The ultrasonic sensor uses this information along with the time difference between sending and receiving the sound pulse to determine the distance to an object.**

**Breadboard:**

**A breadboard is a construction base for prototyping of electronics. In the 1970’s the solder less breadboard (AKA plug board, a terminal array board) became available and nowadays the term “breadboard” is commonly used to refer to these. “Breadboard” is also a synonym for “prototype”. Because the solder less breadboard does not require soldering, it is reusable. This makes it easy to use for creating temporary prototypes and experimenting with circuit design.**

**JUMPER WIRES:**

**A jump wire is an electrical wire or group of them in a cable with a connector or pin at each end (or sometimes without them – simply “tinned”), which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering**

**Arduino IDE:**

**The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board. It contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus.**

**Web Server:**

**A Web server is a program that uses Hypertext Transfer Protocol to serve the files that form Web pages to users, in response to their requests, which are forwarded by their computers’ HTTP clients. Dedicated computers and appliances may be referred to as Web servers as well.**

**Load cell ( Weight Sensor):**

**A load cell is a force transducer. It converts a force such as tension, compression, pressure, or torque into an electrical signal that can be measured and standardized. As the force applied to the load cell increases, the electrical signal changes proportionally.**

**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.**

**MIT APP:**

**MIT App Inventor is a web application integrated development environment originally provided by Google, and now maintained by the Massachusetts Institute of Technology (MIT). It allows newcomers to computer programming to create application software(apps) for two operating systems (OS): Android, and iOS, which, as of 8 July 2019, is in final beta testing.**

**ESP32 Wi-Fi Module:**

**ESP32 is a series of low-cost, low-power system on a chip microcontrollers with integrated Wi-Fi and dual-mode Bluetooth. The ESP32 series employs a Tensilica Xtensa LX6 microprocessor in both dual-core and single-core variations and includes built-in antenna switches, RF balun, power amplifier, low-noise receive amplifier, filters, and power-management modules. ESP32 is created and developed by Espressif Systems, a Shanghai-based Chinese company, and is manufactured by TSMC using their 40 nm process.[2] It is a successor to the ESP8266 microcontroller.**

**IBM Cloud:**

**The IBM SmartCloud brand includes infrastructure as a service, software as a service and platform as a service offered through public, private and hybrid cloud delivery models. IBM places these offerings under three umbrellas: SmartCloud Foundation, SmartCloud Services and SmartCloud Solutions.KEYWORDS/ REQUIREMENTS:-**

**ESP32 microcontroller, GPS, WEIGHT SENSOR MODULE OR LOAD CELLS., ULTRASONIC SENSOR.**

**SOFTWARE REQUIRED;-**

**Arduino IDE , Arduino Language, IBM cloud, MIT APP inventor.**

**RESULT**

* **When the dustbins are filled the details of these dustbins can be accessed by the concerned authorities by using INTERNET and the required action can be taken as soon as possible.**

**ADVANTAGES**

**1. Less time and fuel consumption as the trucks go only to the filled containers.**

**2. Decreased noise, traffic flow and air pollution as a result of less trucks on the roads.**

**3. Our smart operating system enable two way communication between the dustbin deployed in the city and service operator. Therefore the focus is only on collection of route based fill level of the containers.**

**4. The sensors installed in the containers provide real time information on the fill level. 5. This information helps determine when and where to prioritise collection.**

**6. In this way both service providers and citizens benefit from an optimized system which results in major cost savings and less urban pollution.7. Reduces the infrastructure (trucks, containers), operating (fuel) and maintenance costs of the service by upto 30%. 8. Applying this technology to the city optimises management, resources and costs, and makes it a “SMART CITY”.9. Historical information on collections helps adapt the deployment of containers to the actual needs of the city, therefore reducing the number of containers that clutter up the road and increasing public parking spaces.10. It keeps the surroundings clean and green, free from bad odour of wastes, emphasizes on healthy environment and keep cities more beautiful.11. Reducing manpower required to handle the garbage collection.**

**DISADVANTAGES**

**1. The trucks collecting garbage take rounds around a locality even when the dustbins are empty.**

**2. High costs**

**3. Pollution caused to mismanagement of waste also influencing factors like the look , odor and others.**

**4. Accumulation of different insects and organisms which spread illness around.**

**5. Sometimes this also leads to more traffic .**

**APPLICATIONS**

**1. This can be best used by municipal corporation for their betterment of management**

**regarding collection of wastes.**

**2. With the help of proper technology (GPS & SOFTWARE APPLICATIONS) we can guide the trucks to choose the shortest path.**

**3. It also favors the “SMART CITY” project and “DIGITAL INDIA”.**

**CONCLUSION**

* **By using this method the collection of waste in the city becomes more easier. It helps in reducing air pollution, traffic flow, man power, time and money. With the help of proper technology (GPS & SOFTWARE APPLICATIONS) we can guide the trucks in selecting the shortest path for garbage collection. This project can add an edge to the cities aiming to get smart and people-friendly.**

**FUTURE SCOPE**

**1. Smart Waste Management System include a rise in smart city initiatives across different regions and stringent regulations and compliance requirements for environment protection & waste management.**

**2. The odor control mechanism to get rid of foul smell of organic garbage. Also, realizing the requirement of an autonomous dustbin, GPS module can be implemented for path planning combined with ultrasonic sensor for obstacle avoidance.**

**BIBLIOGRAPHY**

**http://www.environmentalistseveryday.org/publications-solid-waste-industry-research/information/history-of-solid-waste-management/early-america-industrial-revolution.php**

**https://www.researchgate.net/publication/326105589\_Solid\_Waste\_Management\_Issues\_in\_Hyderabad\_City**

**https://www.openpr.com/news/1067653/future-scope-of-smart-waste-management-system-market-estimated-to-grow-cagr-rapidly-global-forecast-2018-2025.html**

**APPENDIX**

**SOURCE CODE:-**

NOTE:- THE WEIGHT SENSOR IS REPLACED BY AN LDR FOR TEMPORARY VALUES.

**SERIAL MONITOR OUTPUT**

CODE

#include <WiFi.h>

#include <PubSubClient.h>

int distance;

int a;

String command;

String data="";

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

// CHANGE TO YOUR WIFI CREDENTIALS

const char\* ssid = "dlink-299C";//your wifi ssid

const char\* password = "njpa2410";//your password

// CHANGE TO YOUR DEVICE CREDENTIALS AS PER IN IBM BLUMIX

#define ORG "vrwhb5"

#define DEVICE\_TYPE "ESP32"

#define DEVICE\_ID "13579"

#define TOKEN "Pre12345@" // Authentication Token OF THE DEVICE

// PIN DECLARATIONS

int echopin=4;

int trigpin=16;

const int ldrin = 34;

int ldrvalue;

float longitude=17.5208;

float latitude =78.6307;

//-------- Customise the above values --------

const char publishTopic[] = "iot-2/evt/Data/fmt/json";

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";

char topic[] = "iot-2/cmd/home/fmt/String";// cmd REPRESENT command type AND COMMAND IS TEST

char authMethod[] = "use-token-auth";

char token[] = TOKEN;

char clientId[] = "d:" ORG ":" DEVICE\_TYPE ":" DEVICE\_ID;

WiFiClient wifiClient;

PubSubClient client(server, 1883, callback, wifiClient);int publishInterval = 5000; // 30 seconds

long lastPublishMillis;

void publishData();

void setup() {

pinMode(echopin,INPUT);

pinMode(trigpin,OUTPUT);

Serial.begin(115200);

Serial.println();

wifiConnect();

mqttConnect();

}

void loop() {

digitalWrite(trigpin,HIGH);

delay(1000);

digitalWrite(trigpin,LOW);

int duration=pulseIn(echopin,HIGH);

distance=(duration\*0.034)/2;

delay(1000);

int ldrvalue= analogRead(ldrin);

delay(1000);

if (millis() - lastPublishMillis > publishInterval)

{

publishData();

lastPublishMillis = millis();

}

if (!client.loop()) {

mqttConnect();

}

}

void wifiConnect() {

Serial.print("Connecting to "); Serial.print(ssid);

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

}

Serial.print("WiFi connected, IP address: ");

Serial.println(WiFi.localIP());

}void mqttConnect() {

if (!client.connected())

{

Serial.print("Reconnecting MQTT client to "); Serial.println(server);

while (!client.connect(clientId, authMethod, token)) {

Serial.print(".");

delay(500);

}

initManagedDevice();

Serial.println();

}

}

void initManagedDevice() {

if (client.subscribe(topic)) {

Serial.println("subscribe to cmd OK");

} else {

Serial.println("subscribe to cmd FAILED");

}

}

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

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

Serial.println(topic);

for (int i = 0; i < payloadLength; i++) {

command+= (char)payload[i];

}

Serial.print("data: "+ command);

command= "";

}

void publishData()

{

String payload = "{\"d\":{\"distance\":";

payload += distance;

payload += ",""\"ldrvalue\":";

payload += ldrvalue;

payload += ",""\"longitude\":";

payload += longitude;

payload += ",""\"latitude\":";

payload += latitude;

payload += "}}";

Serial.print("\n");

Serial.print("Sending payload: "); Serial.println(payload);

if (client.publish(publishTopic, (char\*) payload.c\_str())) {

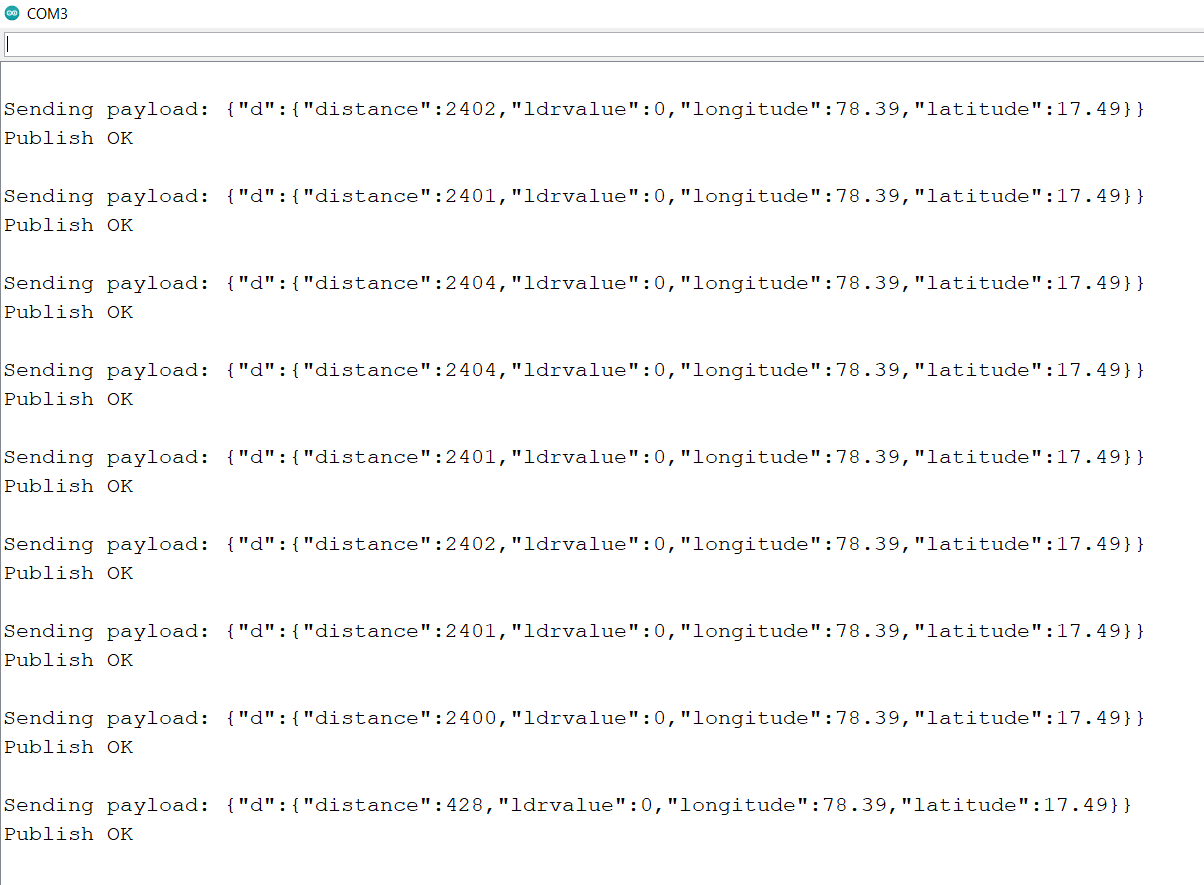
Serial.println("Publish OK");

} else {

Serial.println("Publish FAILED");

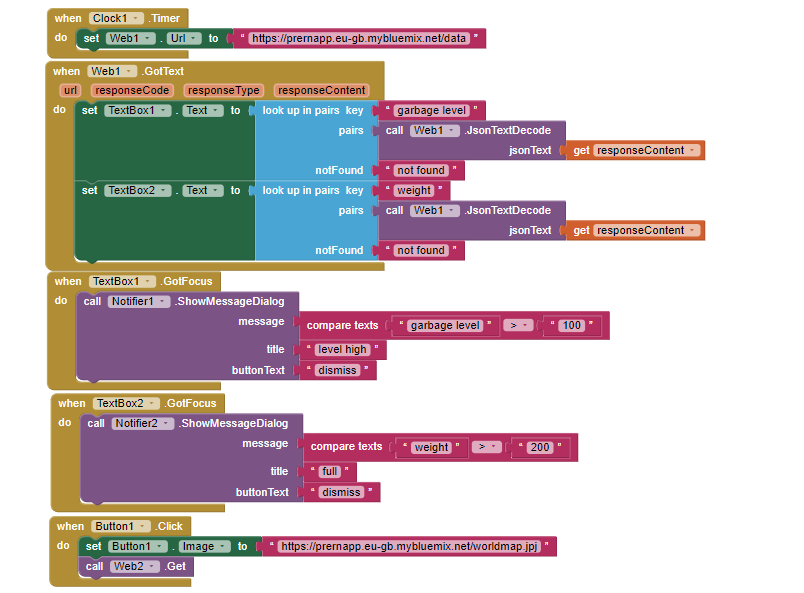
}

}

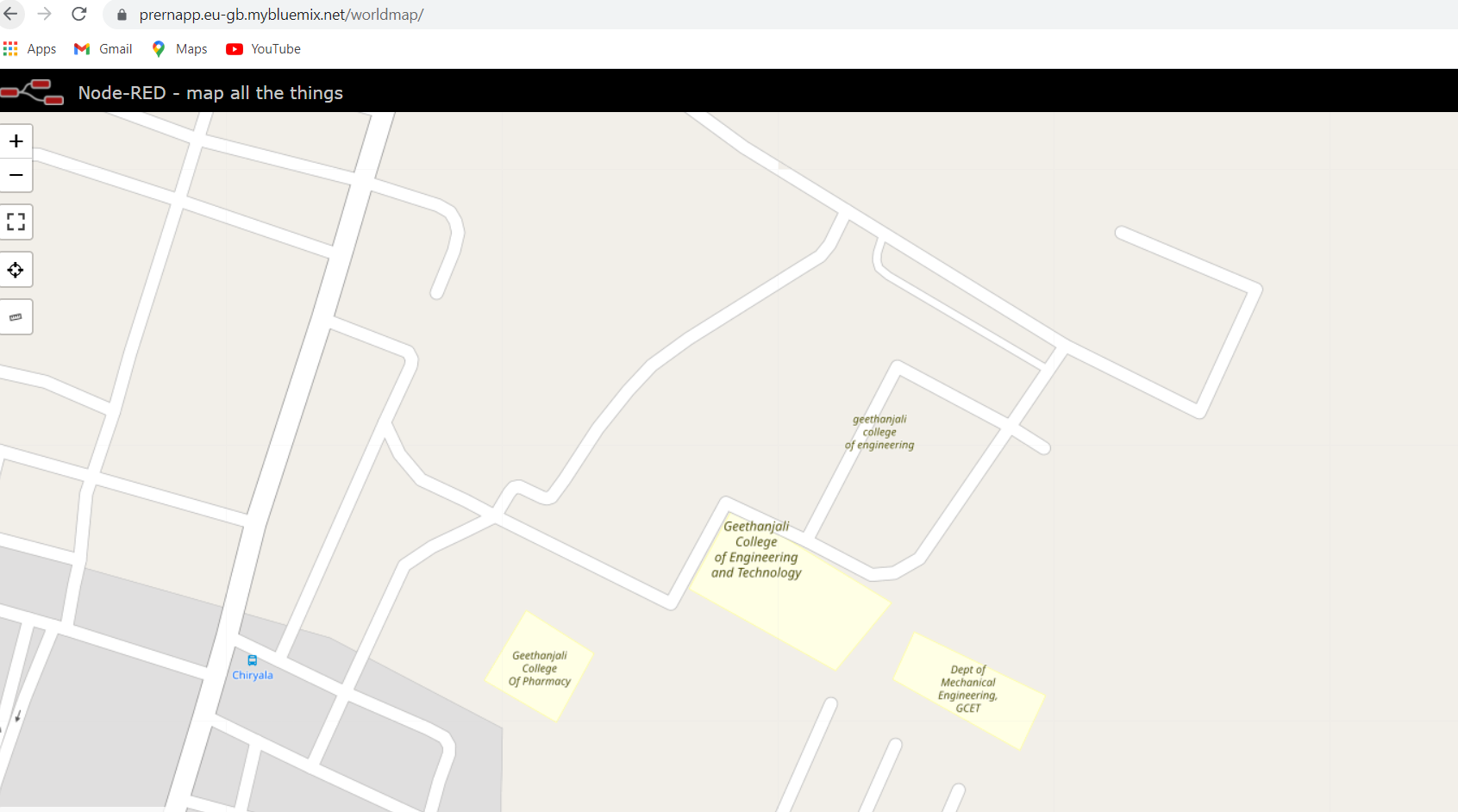
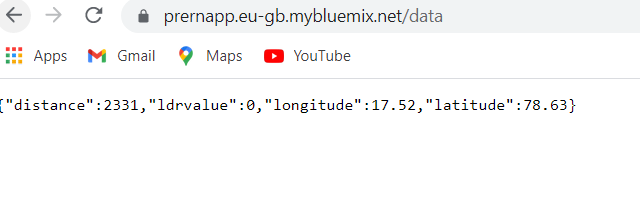
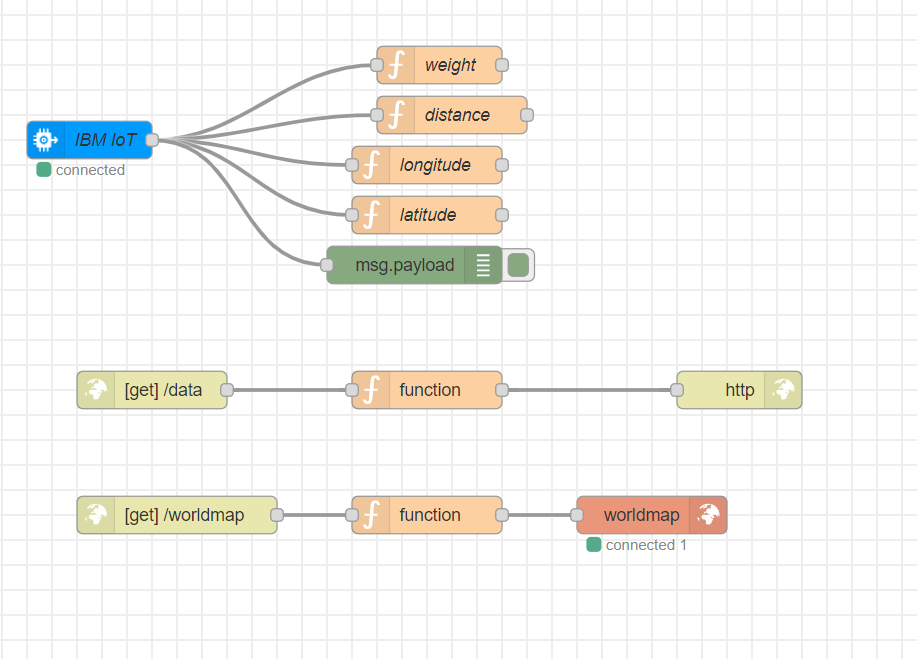


MIT APP INPUTS

BLOCKS:-



NODE RED INPUTS:-



OUPUTS ON MOBILE PHONE:-

