

TEAM ID	PNT2022TMID04297
PROJECT NAME	Smart Waste Management System For Metropolitan Cities

CODE:

register.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <link rel="stylesheet" href="register.css" type="text/css">
  <link rel="stylesheet" href="./Fonts,Icons/css/all.min.css">
  <link rel="icon" href="./Imgs/dustbin.png" type="image/icon">
  <title>REGISTER NOW</title>
</head>
<body>
  <div class="header">
    <h1>SMART WASTE MANAGEMENT</h1>
  </div>
  <div class="container">
    <form action="/action_page.php">
      <div class="formpage">
        <div class="detail1">
          <label for="fname">User name</label>
        </div>
        <div class="detail2">
          <input type="text" id="fname" name="firstname" placeholder="Your
name..">
        </div>
      </div>
      <div class="formpage">
        <div class="detail1">
          <label for="idno">Email ID</label>
        </div>
        <div class="detail2">

```

```
        <input type="text" id="idno" name="idno"
placeholder="abc@gmail.com">
    </div>
</div>
<div class="formpage">
    <div class="detail1">
        <label for="branch">Location</label>
    </div>
    <div class="detail2">
        <select id="area" name="area">
            <option value="nang">NANGANALLUR</option>
            <option value="nan">NANDANAM</option>
            <option value="ady">ADYAR</option>
            <option value="vel">VELACHERY</option>
            <option value="anna">ANNA NAGAR</option>
        </select>
    </div>
</div>
<div class="formpage">
    <div class="detail1">
        <label for="section">Contact</label>
    </div>
    <div class="detail2">
        <input type="text" id="section" name="section" placeholder="+91">
    </div>
</div>
<div class="formpage">
    <div class="detail1">
        <label for="cname">Zip Code</label>
    </div>
    <div class="detail2">
        <input type="text" id="cname" name="companyname"
placeholder="123456">
    </div>
</div>
<div class="formpage">
    <div class="detail1">
        <label for="sdate">Select date</label>
    </div>
    <div class="detail2">
        <input type="date" id="startdate" name="startdate"
placeholder="Select Date..">
    </div>
</div>
```

```

    </div>
</div>
<div class="formpage">
  <div class="detail1">
    <label for="address">Address</label>
  </div>
  <div class="detail2">
    <textarea id="address" name="subject" placeholder="Write
something.." style="height:200px"></textarea>
  </div>
</div>
<div class="formpage">
  <div class="detail1">
    <label for="address">Password</label>
  </div>
  <div class="detail2">
    <input type="text" id="section" name="section" placeholder="Enter
password">
  </div>
</div>
<div class="formpage">
  <div class="detail1">
    <label for="address">Confirm Password</label>
  </div>
  <div class="detail2">
    <input type="text" id="section" name="section" placeholder="Re-
enter password">
  </div>
</div>
<div class="formpage">
  <div class="detail1">
    <label for="file">Upload a file</label>
  </div>
  <div class="detail2">
    <input type="file" id="myFile" name="filename">
  </div>
</div>
<br>
<br>
<div class="formpage">
  <input type="submit" value="Submit">
</div>

```

```
</form>
</div>
</body>
</html>
```

register.css

```
@import
url('https://fonts.googleapis.com/css2?family=Fredoka:wght@400;600&fa
mily=Varela+Round&display=swap');
```

```
* {
padding-left: 110px;
padding-right: 110px;
}
body{
/*background-color: #49b5f1;*/
background-image:url("./Imgs/bg1.JPG") ;
font-family: 'Varela Round', sans-serif;
}
.header h1{
text-align: center;
align-items: center;
color: #076907;
font-family: 'Varela Round', sans-serif;
}
input[type=text], select , textarea {
width: 100%;
padding: 12px;
border: 1px solid #ccc;
border-radius: 5px;
resize: vertical;
font-family: 'Fredoka', sans-serif;
}

label {
padding: 12px 12px 12px 0;
display: inline-block;
font-family: 'Fredoka', sans-serif;
}

input[type=submit] {
```

```
background-color: #31cd31;
color: black;
padding: 12px 20px;
border: none;
border-radius: 4px;
margin-right: 150px;
cursor: pointer;
float: right;
font-family: 'Fredoka', sans-serif;
}
```

```
input[type=submit]:hover {
  background-color: gray;
}
```

```
input[type=file] {
  background-color: #31cd31;;
  color: black;
  padding: 12px 20px;
  border: none;
  margin-left: 7px;
  border-radius: 4px;
  width: 95%;
  font-family: 'Fredoka', sans-serif;
  cursor: pointer;
  float: left;
}
```

```
input[type=file]:hover {
  background-color: grey;
}
```

```
input[type=date] {
  background-color: #31cd31;
  color: black;
  padding: 12px 20px;
  border: none;
  margin-left: 7px;
  border-radius: 4px;
  width: 95%;
  font-family: 'Fredoka', sans-serif;
  cursor: pointer;
  float: left;
}
```

```
input[type=date]:hover {  
  background-color: grey;  
}
```

```
.container {  
  border-radius: 5px;  
  background-color: #cbe6f4ea;  
  padding: 20px;  
  margin-right: 10px;  
  padding-right: 110px;  
}
```

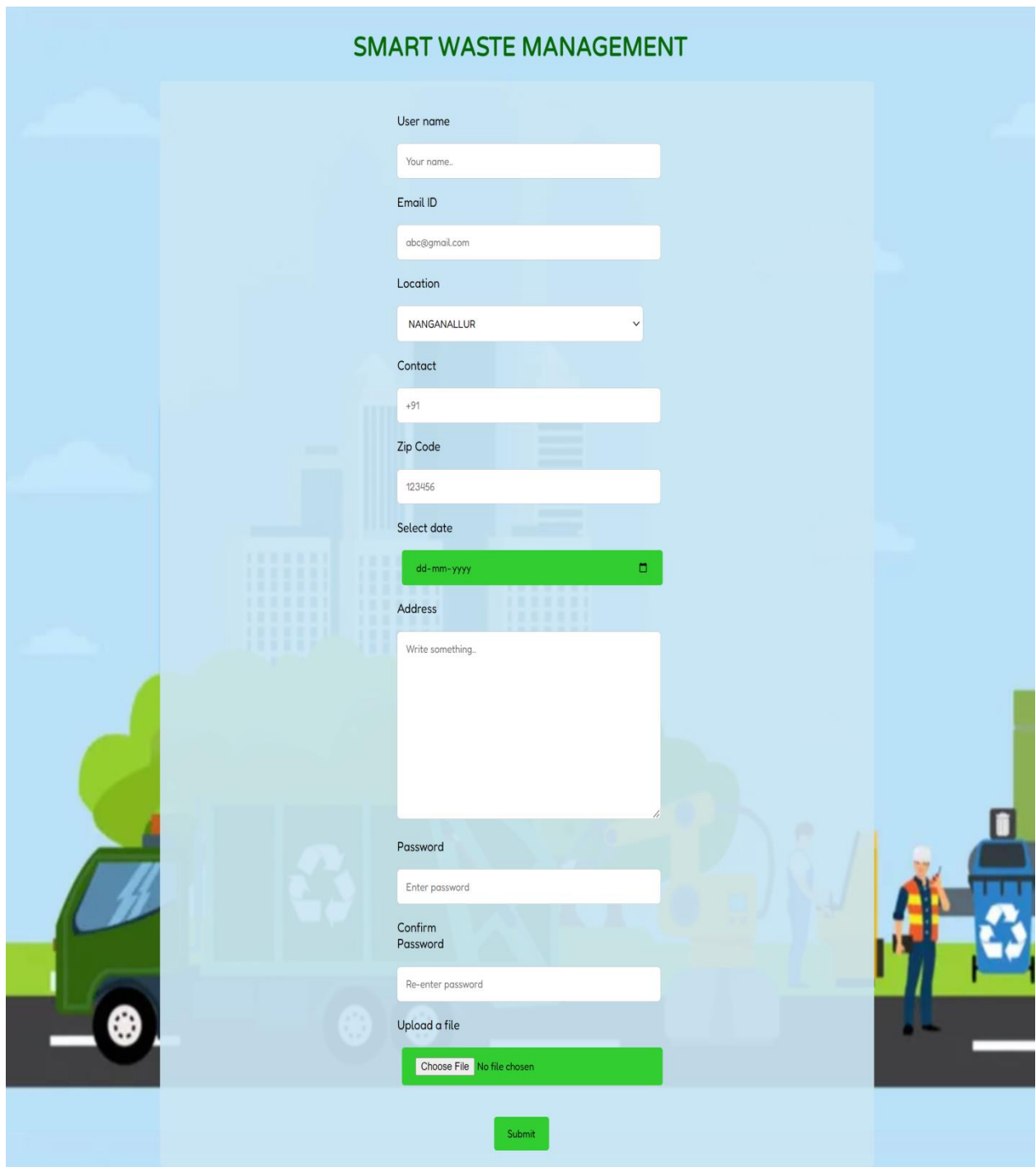
```
.detail1 {  
  float: left;  
  width: 25%;  
  margin-top: 6px;  
  font-family: 'Varela Round', sans-serif;  
}
```

```
.detail2 {  
  float: left;  
  width: 75%;  
  margin-top: 6px;  
  font-family: 'Varela Round', sans-serif;  
}
```

```
.formpage:after {  
  content: "";  
  display: table;  
  clear: both;  
  text-align: center;  
}
```

```
@media screen and (max-width: 600px) {  
  .detail1, .detail2, input {  
    width: 100%;  
    margin-top: 0;  
  }  
}
```

OUTPUT:



The image shows a web form titled "SMART WASTE MANAGEMENT" overlaid on a background illustration of a city street with a green waste management truck, a recycling bin, and a worker. The form contains the following fields:

- User name:
- Email ID:
- Location:
- Contact:
- Zip Code:
- Select date:
- Address:
- Password:
- Confirm Password:
- Upload a file: No file chosen
-

CODE:

```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQTT
#include "DHT.h"// Library for dht11
#define DHTPIN 15 // what pin we're connected to
#define DHTTYPE DHT22 // define type of sensor DHT 11
```

```
#define LED 2
```

```
DHT dht (DHTPIN, DHTTYPE); //creating the instance by passing pin and type of  
dht connected
```

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

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

```
#define ORG "w5zj5y" //IBM ORGANIZATION ID
```

```
#define DEVICE_TYPE "abcdef" //Device type mentioned in IBM Watson IOT  
Platform
```

```
#define DEVICE_ID "123456" //Device ID mentioned in IBM Watson IOT  
Platform
```

```
#define TOKEN "?nIvmVuhCTf9JaTSuu" //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/command/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 wifi client
```

```
PubSubClient client(server, 1883, callback, wifiClient); //calling the predefined  
client id by passing parameter like server id, port and wifi credential
```

```
void setup() // configuring the ESP32
```

```
{
```

```
  Serial.begin(115200);
```

```
  dht.begin();
```

```
  pinMode(LED, OUTPUT);
```



```

    delay(10);
    Serial.println();
    wificonnect();
    mqttconnect();
}

void loop()// Recursive Function
{

    h = dht.readHumidity();
    t = dht.readTemperature();
    Serial.print("temp:");
    Serial.println(t);
    Serial.print("Humid:");
    Serial.println(h);

    PublishData(t, h);
    delay(1000);
    if (!client.loop()) {
        mqttconnect();
    }
}

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

void PublishData(float temp, float humid) {
    mqttconnect();//function call for connecting to ibm
    /*
        creating the String in in form JSon to update the data to ibm cloud
    */
    String payload = "{\"temp\":";
    payload += temp;
    payload += ", \"Humid\":";
    payload += humid;
    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);
  if(data3=="lighton")
  {
    Serial.println(data3);
    digitalWrite(LED,HIGH);
  }
  else
  {
    Serial.println(data3);
    digitalWrite(LED,LOW);
  }
  data3="";
}

```

OUTPUT:

WOKWI

SAVE

SHARE

Docs

phase1.ino

diagram.json

libraries.txt

Library Manager

```

1 #include <WiFi.h> //library for wifi
2 #include <PubSubClient.h> //library for MQTT
3 #include <LiquidCrystal_I2C.h>
4 LiquidCrystal_I2C lcd(0x27, 20, 4);
5 // credentials of IBM Accounts -
6 #define ORG "w5zj5y" //IBM organisation id
7 #define DEVICE_TYPE "abcdef" // Device type mentioned in ibm watson iot platfo
8 #define DEVICE_ID "123456" // Device ID mentioned in ibm watson iot platform
9 #define TOKEN "?nIvmVuhCTf9JaTSuu" // Token
10 // customise above values -
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // server name
12 char publishTopic[] = "iot-2/evt/data/fmt/json";
13 char topic[] = "iot-2/cmd/led/fmt/String"; // cmd Represent type and command i
14 char authMethod[] = "use-token-auth"; // authentication method
15 char token[] = TOKEN;
16 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //Client id
17 //
18 WiFiClient wifiClient; // creating instance for wificlient
19 PubSubClient client(server, 1883, wifiClient);
20 #define ECHO_PIN 12
21 #define TRIG_PIN 13
22 float dist;
23 void setup()
24 {
25   Serial.begin(115200);
26   pinMode(LED_BUILTIN, OUTPUT);
27   pinMode(TRIG_PIN, OUTPUT);
28   pinMode(ECHO_PIN, INPUT);
29   //pir pin
30   pinMode(4, INPUT);

```

Simulation

01:25.782 99%

Reconnecting MQTT client to
w5zj5y.messaging.internetofthings.ibmcloud.com
IBM subscribe to cmd OK

No motion detected
No motion detected
No motion detected

WOKWI

SAVE

SHARE

Docs

phase1.ino

diagram.json

libraries.txt

Library Manager

```

1 #include <WiFi.h> //library for wifi
2 #include <PubSubClient.h> //library for MQTT
3 #include <LiquidCrystal_I2C.h>
4 LiquidCrystal_I2C lcd(0x27, 20, 4);
5 // credentials of IBM Accounts -
6 #define ORG "w5zj5y" //IBM organisation id
7 #define DEVICE_TYPE "abcdef" // Device type mentioned in ibm watson iot platfo
8 #define DEVICE_ID "123456" // Device ID mentioned in ibm watson iot platform
9 #define TOKEN "?nIvmVuhCTf9JaTSuu" // Token
10 // customise above values -
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // server name
12 char publishTopic[] = "iot-2/evt/data/fmt/json";
13 char topic[] = "iot-2/cmd/led/fmt/String"; // cmd Represent type and command i
14 char authMethod[] = "use-token-auth"; // authentication method
15 char token[] = TOKEN;
16 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //Client id
17 //
18 WiFiClient wifiClient; // creating instance for wificlient
19 PubSubClient client(server, 1883, wifiClient);
20 #define ECHO_PIN 12
21 #define TRIG_PIN 13
22 float dist;
23 void setup()
24 {
25   Serial.begin(115200);
26   pinMode(LED_BUILTIN, OUTPUT);
27   pinMode(TRIG_PIN, OUTPUT);
28   pinMode(ECHO_PIN, INPUT);
29   //pir pin
30   pinMode(4, INPUT);

```

Simulation

01:51.820 69%

Sending distance: 184.96
Publish OK
Motion Detected
Lid Opened
Warning!! Trash is about to cross 50% of bin level
Lid Closed

IBM Watson IoT Platform

412519205153@smartinternz.com
ID: w6zj6y

Browse Action Device Types Interfaces

Search by Device ID

Device Simulator

Device ID	Status	Device Type	Class ID	Date Added
123456	Connected	abcdef	Device	Nov 18, 2022 1:11 PM

Identity Device Information Recent Events State Logs

Device ID: 123456
Device Type: abcdef
Date Added: Nov 18, 2022 1:11 PM
Added By: 412519205153@smartinternz.com
Connection Status: Connected
Connection Time: Nov 18, 2022 3:11 PM
Client Address: 50.31.197.64 Insecure

Items per page 50 | 1-1 of 1 item

1 Simulation running

IBM Watson IoT Platform

412519205153@smartinternz.com
ID: w6zj6y

Browse Action Device Types Interfaces

Search by Device ID

Device Simulator

Device ID	Status	Device Type	Class ID	Date Added
123456	Disconnected	abcdef	Device	Nov 18, 2022 1:11 PM

Identity Device Information Recent Events State Logs

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
event_1	{"randomNumber":64}	json	a few seconds ago
event_1	{"randomNumber":12}	json	a few seconds ago
event_1	{"randomNumber":62}	json	a few seconds ago
event_1	{"randomNumber":95}	json	a few seconds ago
event_1	{"randomNumber":44}	json	a few seconds ago

1 Simulation running

Activate Windows
Go to PC settings to activate Windows.

CODE:

```
import wiotp.sdk.device
import time
import random
import collections.abc
try:
    from collections.abc import MutableMapping
except ImportError:
    from collections import MutableMapping
myConfig = {
```

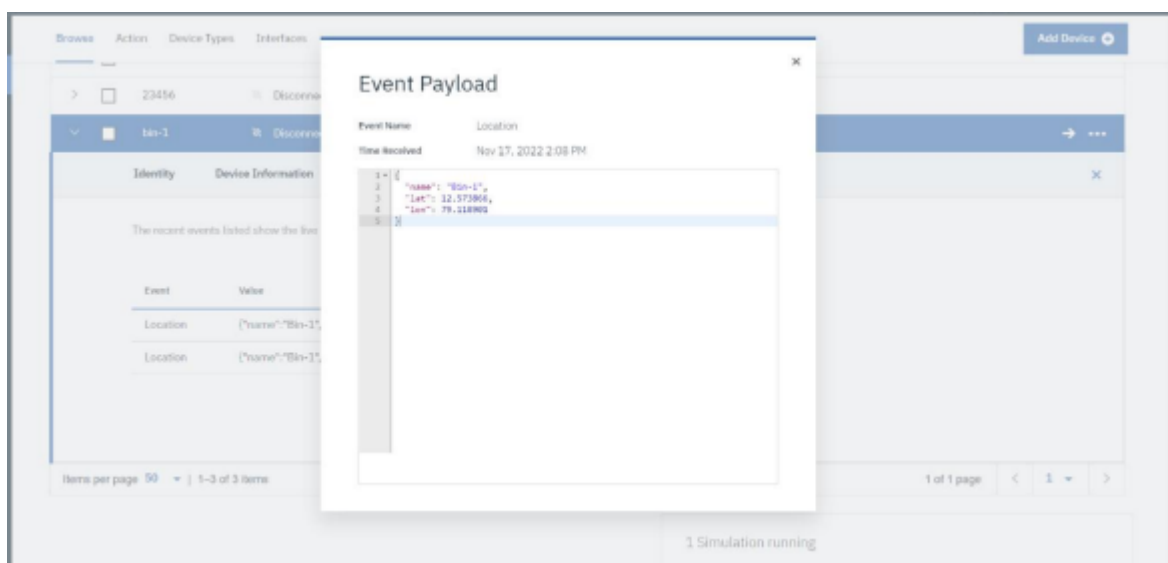
```

"identity": {
  "orgId": "gx76pd",
  "typeId": "SmartBin",
  "deviceId": "bin-1"
},
"auth": {
  "token": "ZeskE9*BHtQSqNIICL"
}
}

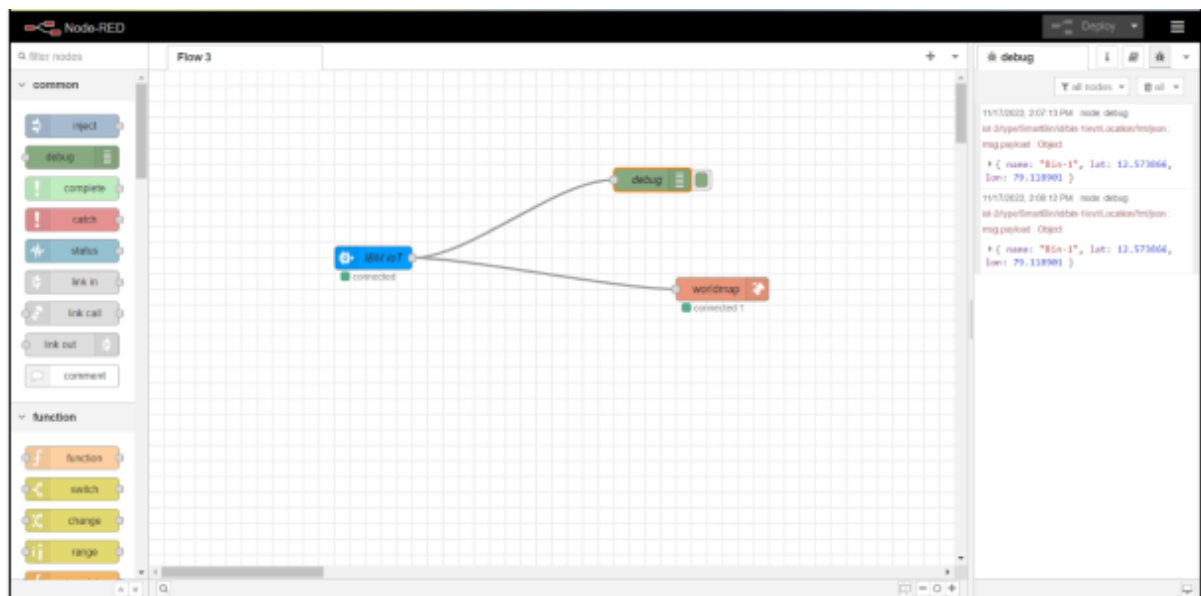
def myCommandCallback (cmd):
print ("Message received from IBM IoT Platform: %s" % cmd.data['command'])
m=cmd.data['command']
client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()
def pub (data):
client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
onPublish=None)
print ("Published data Successfully: %s", myData)
while True:
myData={'name': 'Bin1', 'lat': 13.092677, 'lon': 80.188314}
pub (myData)
time.sleep (3)
client.commandCallback = myCommandCallback
client.disconnect ()

```

OUTPUT:



NODE-RED:



BIN LOCATION ON WORLD MAP:

