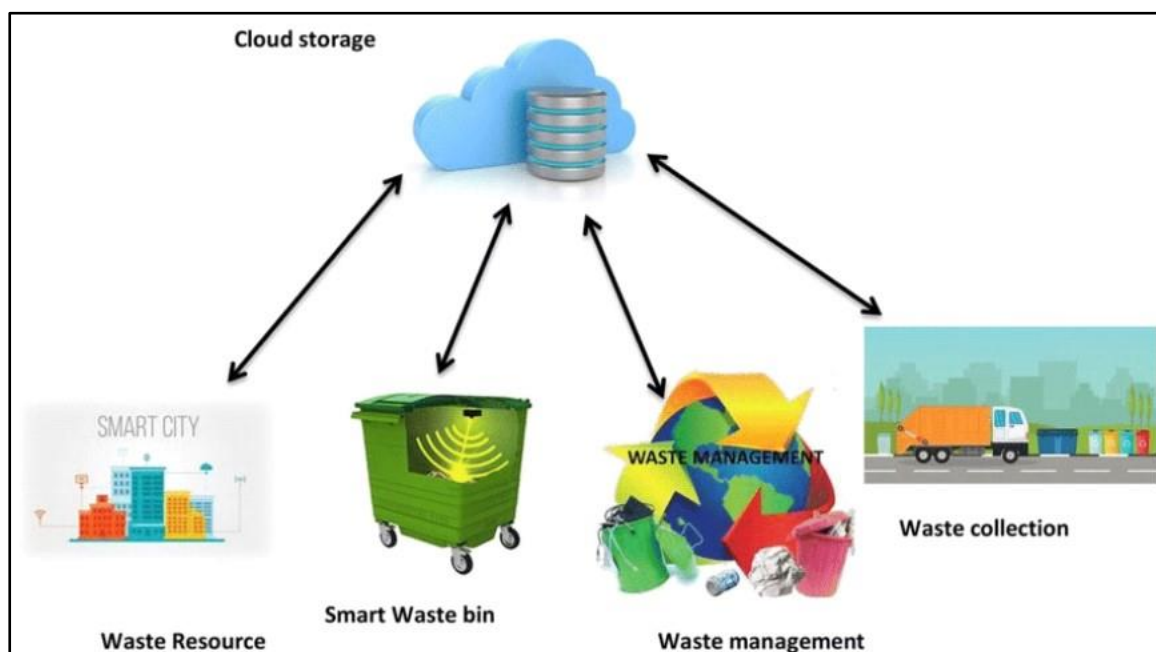


# SMART WASTE MANAGEMENT FOR METROPOLITAN CITIZENS

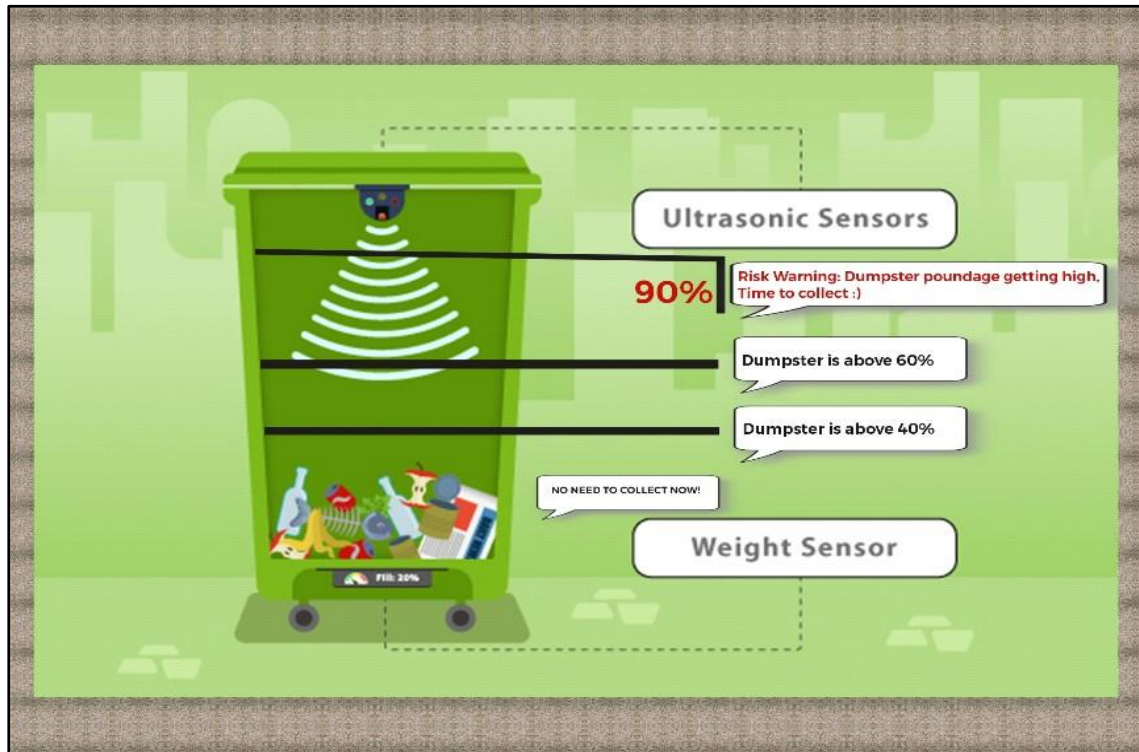
Team ID:PNT2022TMID47006

## PROJECT SOFTWARE

BLOCK DIAGRAMATIC REPRESENTATION OF SMART  
WASTE MANAGEMENT  
SYSTEMATIC REPRESENTATION



SYSTEMATIC REPRESENTATION:



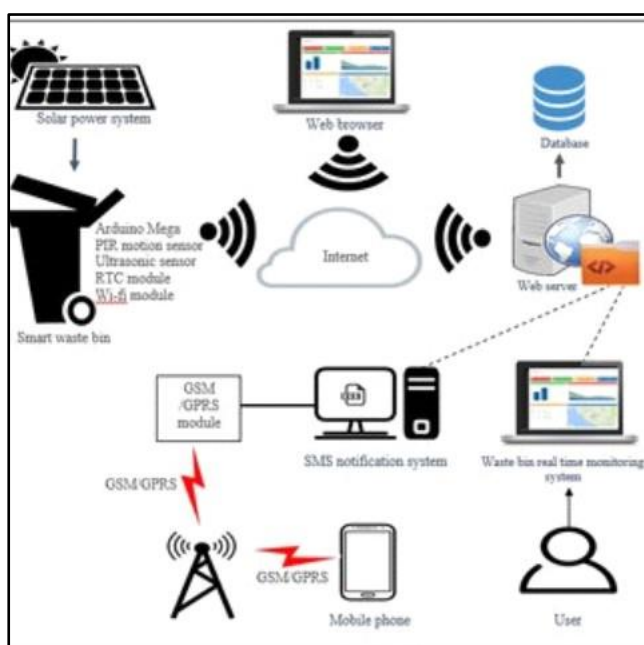
LOGO OF SYSTEMATIC VIEW



## SCAN CODER OF SMART WASTE WEB APPLICATION



## OVERALL VIEW OF THE ENTIRE SMART WASTE MANAGEMENT NETWORK





```

{
  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);
});

```

## index.html

```
<!DOCTYPE html> <html>
```

```
<head>
```

```

  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.c ss"
integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZ w1T"
crossorigin="anonymous">
    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width">

    <title>Garbage Management System</title>

```

```

<link rel="icon" type="image/x-icon" href="/Images/DUMPSTER.png">

<link href="style.css" rel="stylesheet" type="text/css" />

<script
src="https://www.gstatic.com/firebasejs/8.10.1/firebaseapp.js"></script>

<script
src="https://www.gstatic.com/firebasejs/8.10.1/firebasefirebase.js"></script>

<script> var
    firebaseConfig =
    {
        apiKey: "AlzaSyB9ysbnaWc3IyeCioh-aJQT_UCMd5CBFeU",
        authDomain: "fir-test-923b4.firebaseio.com",
        databaseURL: "https://fir-test-923b4-default-rtdb.firebaseio.com",
        projectId: "fir-test-923b4",
        storageBucket: "fir-test-923b4.appspot.com",
        messagingSenderId: "943542145393",
        appId: "1:943542145393:web:9b5ec7593e6a3cbd7966d0",
        measurementId: "G-BN7JNX1Q7B"
    };
    firebase.initializeApp(firebaseConfig)
</script>

<script defer src="database.js"></script>
</head>

<body style="background-color:#1F1B24;">

    <script src="map.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 btn-dark">DUMPSTER</a></center>

    <script

        src="https://maps.googleapis.com/maps/api/js?key=AlzaSyBBLyWj-
        3FWtCbCXGW3ysEil2fDfrv2v0Q&callback=myMap"></script></div>

</body>

</html>
```

## map.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);
});
}

```

## replix.nix.txt

```

{ pkgs }: { deps
= [
    pkgs.nodePac
    kages.vscode-
    langservers-
    extracted
    pkgs.nodePac

```



```
kages.typescri  
pt-language-  
server  
];  
}
```

## style.css

```
html, body  
{  
  height: 100%; margin:  
    0px; padding:0px;  
}  
  
#container  
{ display: flex; flex-  
  direction: row;  
  height: 100%;  
  width: 100%;  
  position: relative;  
}  
  
#logo_container  
{  
  height: 100%; width: 12%;  
  background-color: #C5C6D0;  
  display: flex;
```

```
        flex-direction: column; vertical-align:
        text-bottom;
    }
```

```
.logo
```

```
{
    width:70%; margin:
    5% 15%;
```

```
/* border-radius: 50%; */
```

```
}
```

```
#logo_3
```

```
{
    vertical-align: text-bottom;
```

```
}
```

```
#data_container
```

```
{
    height: 100%; width:
    20%; margin-left: 1%;
    margin-right: 1%;
    display: flex; flex-
    direction: column;
}
```

```
#data_status
```

```
{
    height:60%; width:8%;
    margin:7%;
    background-color:
```

```
#691F6E; display: flex;

flex-direction:

column; border-

radius:20px;

}

#load_status

{

    background-image: url("/Images/KG.png");

    background-repeat: no-repeat; background-size:

        170px;

    background-position: left center;

}

#cap_status

{

    background-image: url("/Images/dust.png");

    background-repeat: no-repeat; background-size:

        150px;

    background-position: left center;

}

.status

{

    width: 80%; height: 40%;

    margin:5% 10%; background-

    color:#185adc; border-

    radius:20px; display: flex;

    justify-content: center;

    align-items: center;
```

```
    color: white; font-size:
    60px;

}

.datas
{
    width:86%; margin:2.5%
    7%; height:10%;
    background: url(water.png);
    background-repeat: repeat-x; animation:
    datas 10s linear infinite;

    box-shadow: 0 0 0 6px #98d7eb, 0 20px 35px rgba(0,0,0,1);

}

#map_container
{
    height: 100%; width:
    100%; display: flex;
    flex-direction: column;
}

#live_location_heading
{
    margin-top:10%; text-align:
    center;
    color: GREY;
}
```

```
#map
{
    height: 70%; width: 90%;
    margin-left: 4%; margin-
    right:4%; border: 10px
    solid white; border-
    radius: 25px;
}

#alert_msg
{
    width:92%; height:20%;
    margin:4%; background-
    color:grey; border-
    radius: 20px; display:
    flex; justify-content:
    center; align-items:
    center; color: #41af7f;
    font-size: 25px; font-
    weight: bold;
}

.lat
{
    margin: 0px; font-size:0px;
}
```

```

@keyframes datas{
  0%
  {
    background-position: -500px 100px;
  }
  40%
  {
    background-position: 1000px -10px;
  }

  80% {
    background-position: 2000px 40px;
  }
  100% { background-position: 2700px
    95px;
  }
}

```

## CODE FOR DATA TRANSFER SENSOR

```

#include <WiFi.h> //library for wifi
#include <PubSubClient.h> //library for MQTT #include
<LiquidCrystal_I2C.h> LiquidCrystal_I2C lcd(0x27, 20, 4);
// credentials of IBM Accounts #define ORG
"ktymlx" //IBM organisation id
#define DEVICE_TYPE "new" // Device type mentioned in ibm watson iot platform

```

```

#define DEVICE_ID "09874" // Device ID mentioned in ibm watson iot platform

#define TOKEN "hariwignesh123" // Token

// customise above values -

char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // server name
char publishTopic[] = "iot-2/evt/data/fmt/json"; char

topic[] = "iot-2/cmd/led/fmt/String"; // cmd Represent type and command is test
format of strings char authMethod[] = "usetoken auth"; // authentication method
char token[] = TOKEN;

char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //Client id

//

WiFiClient wifiClient; // creating instance for wificlient

PubSubClient client(server, 1883, wifiClient);

#define ECHO_PIN 12 #define

TRIG_PIN 13 float

dist;

void setup()

{

Serial.begin(115200);

pinMode(LED_BUILTIN, OUTPUT);

pinMode(TRIG_PIN, OUTPUT);

pinMode(ECHO_PIN, INPUT); //pir

pin pinMode(4, INPUT);

//ledpins pinMode(23,

OUTPUT); pinMode(2,

OUTPUT); pinMode(4,

OUTPUT); pinMode(15, OUTPUT);

lcd.init();

lcd.backlight();

lcd.setCursor(1, 0);

```

```

lcd.print("");
wifiConnect();
mqttConnect();
}

float readcmCM()
{
digitalWrite(TRIG_PIN, LOW);
delayMicroseconds(2); digitalWrite(TRIG_PIN,
HIGH); delayMicroseconds(10);
digitalWrite(TRIG_PIN, LOW); int duration =
pulseIn(ECHO_PIN, HIGH); return duration *
0.034 / 2;
}

void loop()
{
lcd.clear();
publishData();
delay(500);
if (!client.loop())
{
mqttConnect(); //function call to connect to IBM
}
}

/* -retrieving to cloud */ void
wifiConnect()
{
Serial.print("Connecting to ");
Serial.print("Wifi");

```



```

WiFi.begin("Wokwi-GUEST", "", 6); 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("IBM subscribe to cmd OK");
}
else
{

```

```
Serial.println("subscribe to cmd FAILED");  
  
}  
  
}  
  
void publishData()  
{  
float cm = readcmCM(); if(digitalRead(34))  
//PIR motion detection  
{  
Serial.println("Motion  
Detected"); Serial.println("Lid  
Opened"); digitalWrite(15, HIGH);  
}  
else  
{  
digitalWrite(15, LOW);  
}  
if(digitalRead(34)== true)  
{  
if(cm <= 100) //Bin level detection  
{  
digitalWrite(2, HIGH);  
Serial.println("High Alert!!!,Trash bin is about to be full");  
Serial.println("Lid Closed"); lcd.print("Full! Don't use"); delay(2000);  
lcd.clear(); digitalWrite(4, LOW); digitalWrite(23, LOW);  
}  
else if(cm > 150 && cm < 250)  
{  
digitalWrite(4, HIGH);
```

```

Serial.println("Warning!!,Trash is about to cross 50% of bin level"); digitalWrite(2,
LOW); digitalWrite(23,
LOW);
}
else if(cm > 250 && cm <=400)
{
digitalWrite(23, HIGH); Serial.println("Bin is
available"); digitalWrite(2,LOW); digitalWrite(4,
LOW);
}
delay(10000); Serial.println("Lid Closed");
}
else
{
Serial.println("No motion detected");
}
if(cm <= 100)
{
digitalWrite(21,HIGH);
String payload = "{\"High Alert!!\":\":"; payload
+= cm; payload
+= "left\" }";
Serial.print("\n");
Serial.print("Sending payload: ");
Serial.println(payload); if (client.publish(publishTopic, (char*) payload.c_str()))
// if data is uploaded to cloud successfully,prints publish ok or
prints publish failed
{

```

```

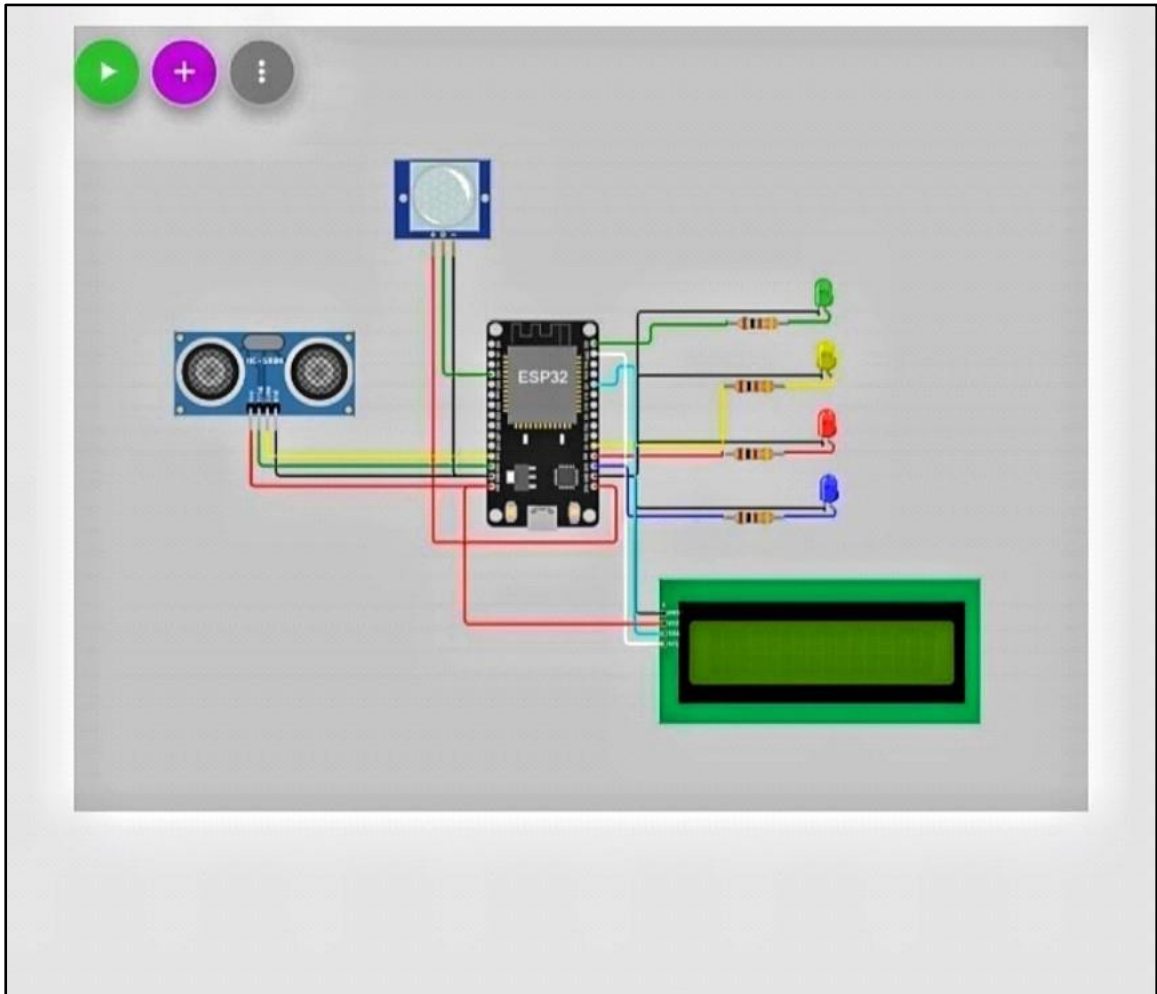
Serial.println("Publish OK");
}
}
if(cm <= 250)
{
digitalWrite(22,HIGH);
String payload = "{\"Warning!!\":\"\""; payload
+= dist; payload += "left\" }";
Serial.print("\n");
Serial.print("Sending distance: "); Serial.println(cm); if(client.publish(publishTopic,
(char*) payload.c_str()))
{
Serial.println("Publish OK");
}
else
{
Serial.println("Publish FAILED");
}
}

float inches = (cm / 2.54); //print on LCD lcd.setCursor(0,0); lcd.print("Inches");
lcd.setCursor(4,0);

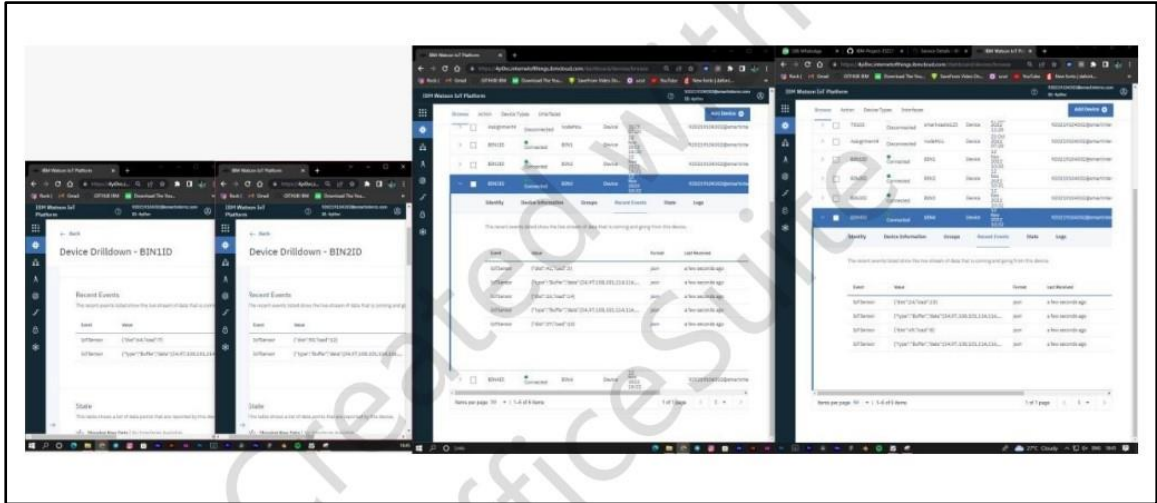
lcd.setCursor(12,0); lcd.print("cm"); lcd.setCursor(1,1); lcd.print(inches, 1);
lcd.setCursor(11,1); lcd.print(cm, 1); lcd.setCursor(14,1); delay(1000);
lcd.clear();
}

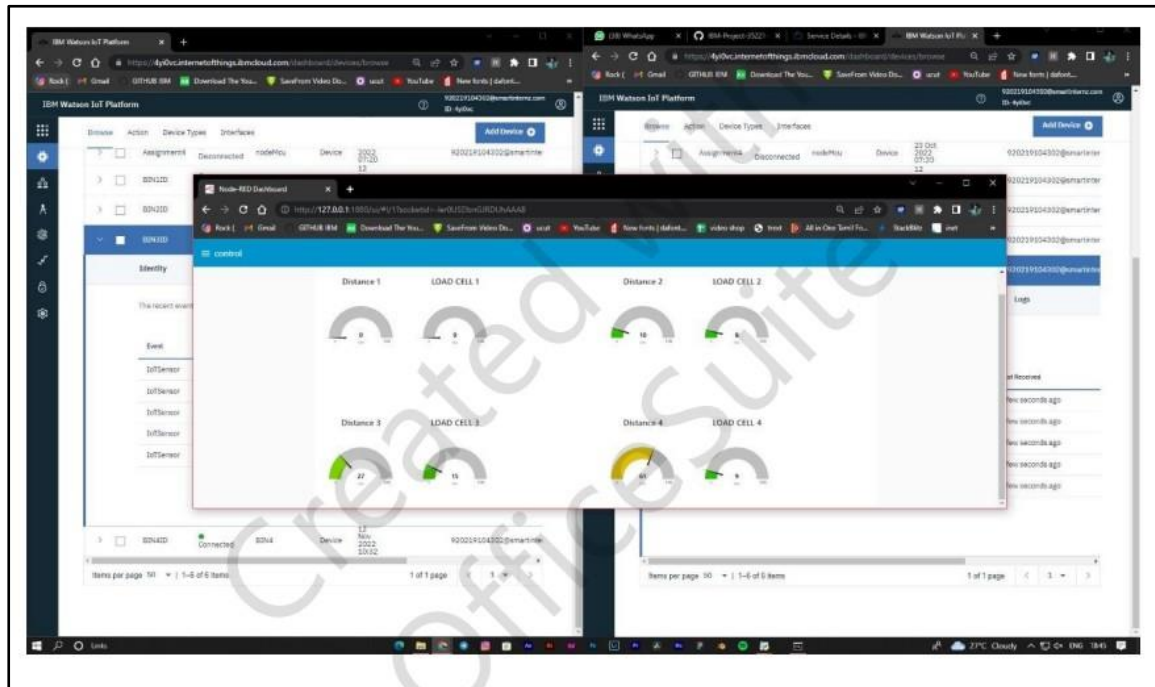
```

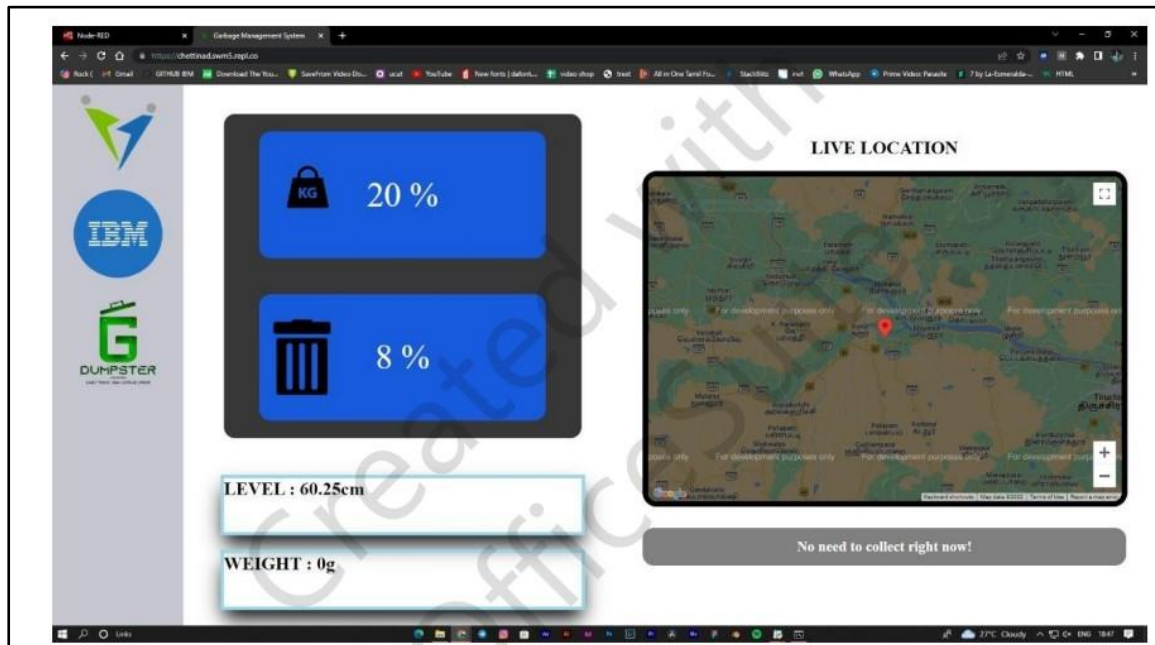
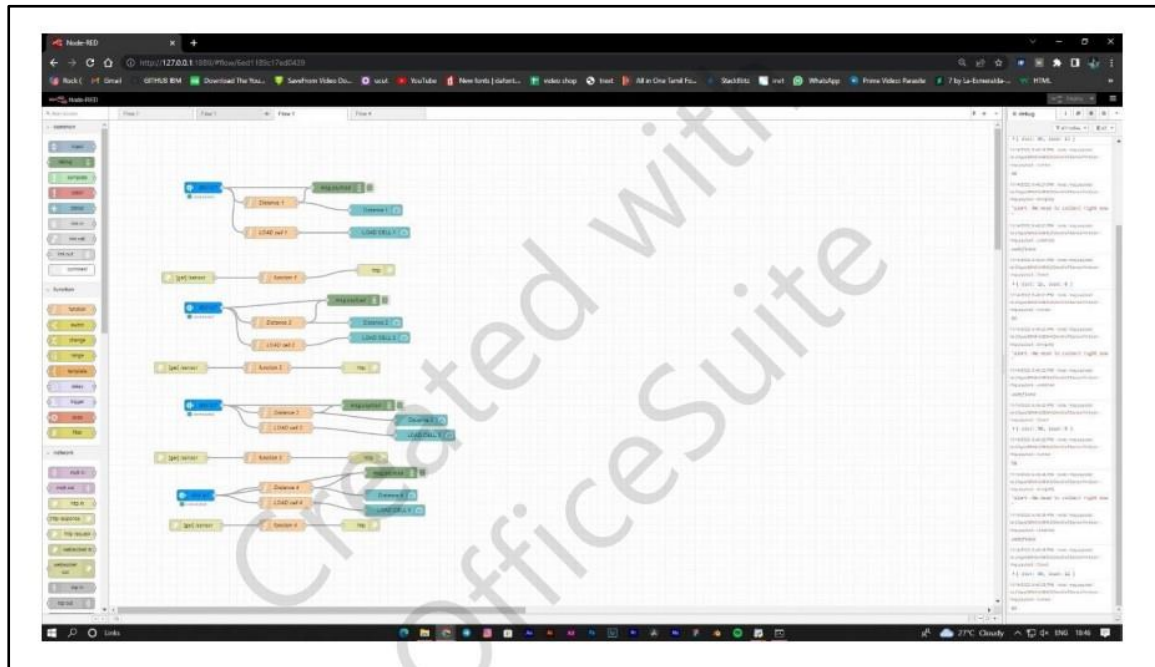
# CONNECTION DIAGRAM



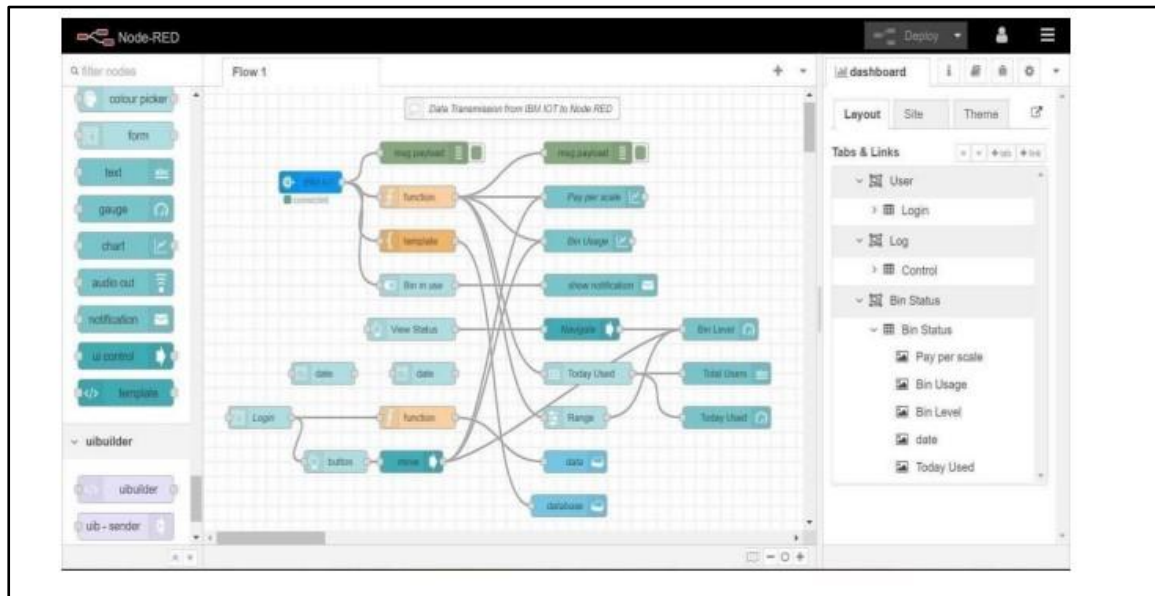
# OUTPUT IMAGES







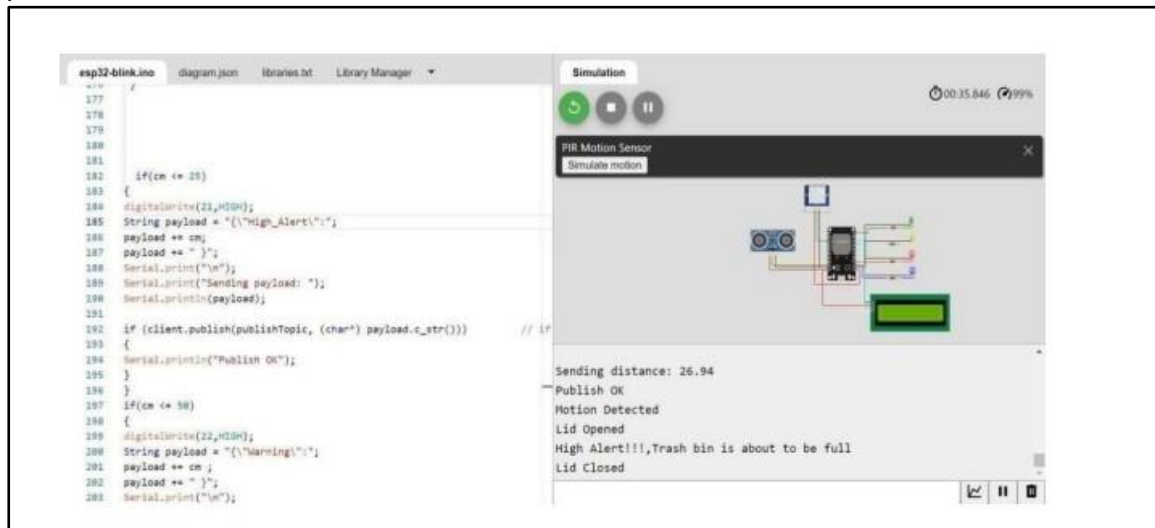
Node-RED Connection setup for data transmission from IBM Watson IOT platform to Node-RED dashboard.



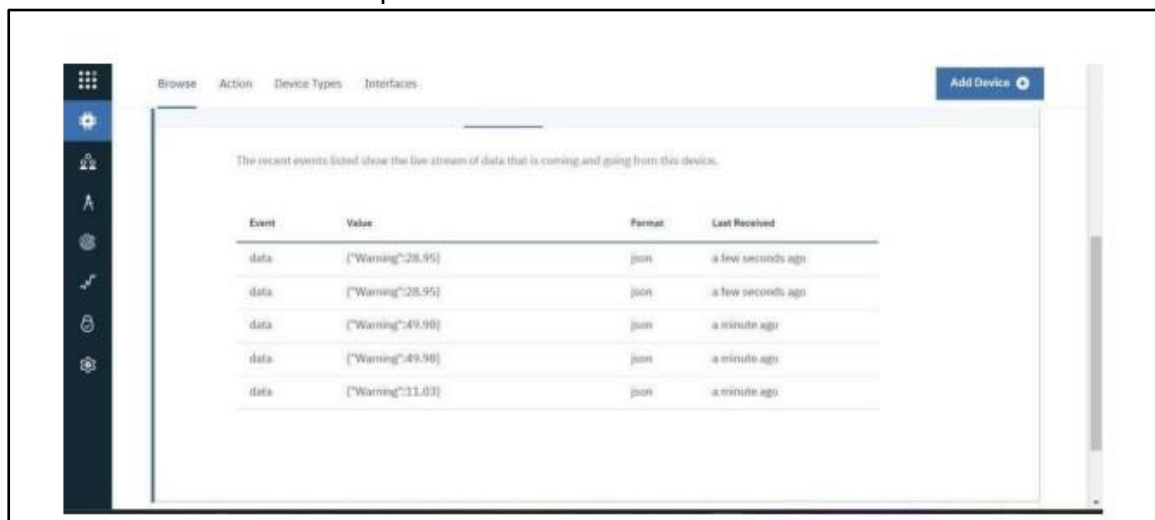


Simulate Wokwi connection to transmit data from wokwi account to IBM WatsonIoT

platform and then to Node Red dashboard

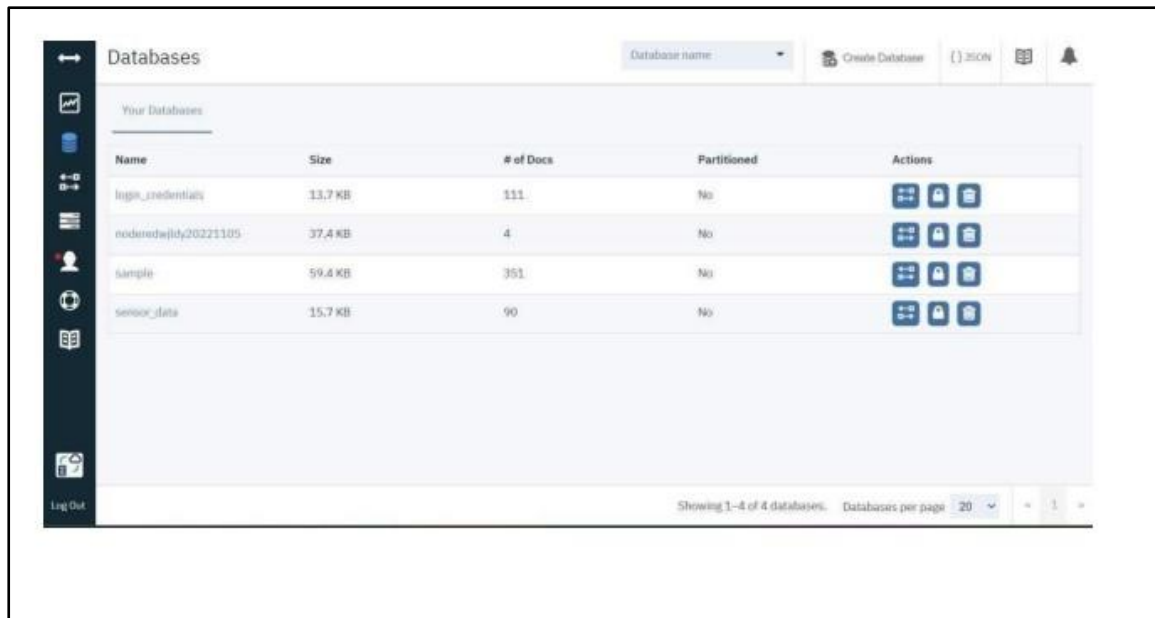


Data transfer to Watson IOT platform.



Data transfer from IBM Watson IOT platform and wokwi to Node red





Data is stored in JSON format



Web UI

