

SPRINT_2

Team ID	PNT2022TMID27888
Project Name	Smart Waste Management System for Metropolitan Cities

AIM:

To create a device for the Cloudant database and IoT Watson Cloud platform. Publish data on the IoT Watson Cloud platform, such as bin weight and bin status values(level). Store user information in the Cloudant database, including user name, password, and Gmail. Using a local Node-RED application, to design the workflow for IoT contexts.

COMPONENTS:

S.NO	COMPONENTS	QUANTITY
1	Ultrasonic sensor	2
2	Servo motor	1
3	ESP32 Microcontroller	1
4	HX711 Load Cell	1
5	LCD 16*2 Display	1

DESCRIPTION:

The below code will display the status of both bin level and bin load for a single smart bin. The status of the bin level is shown in the 16*2 LCD display. The data obtained from the sensors is then published to the IoT Watson Cloud Platform. The sensor data can be visually depicted using node-red application.

CODE:

```
#include <ESP32Servo.h>
#include <LiquidCrystal_I2C.h>
#include <HX711.h>
#define DATA_PIN 12
#define CLOCK_PIN 14
#include <WiFi.h>
#include <PubSubClient.h>
WiFiClient wifiClient;
#define ORG "uuyxja"
#define DEVICE_TYPE "NodeMcu"
```

```

#define DEVICE_ID "12345"
#define TOKEN "23323850"
#define speed 0.034
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Data/fmt/json";
char topic[] = "iot-2/cmd/home/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
PubSubClient client(server, 1883, wifiClient);
void publishData();
const int trigpin=2;
const int echopin=15;
String command;
String data="";
long duration;
float dist;
LiquidCrystal_I2C LCD = LiquidCrystal_I2C(0x27, 16, 2);
Servo servo;
int trigPin1 = 2;
int echoPin1 = 15;
int trigPin2 = 18;
int echoPin2 = 5;
int duration1;
int distance1;
int duration2;
int distance2;
void setup()
{
  Serial.begin(115200);
  LCD.begin(16,2);
  LCD.init();
  LCD.backlight();
  LCD.clear();
  servo.attach(23);
  Serial.begin(115200);
  pinMode(trigPin1, OUTPUT);
  pinMode(echoPin1, INPUT);
  pinMode(trigPin2, OUTPUT);
  pinMode(echoPin2, INPUT);
  wifiConnect();
  mqttConnect();
}
void loop() {
  publishData();
  delay(500);
  if (!client.loop()) {
    mqttConnect();
  }
}

```

```

}
}
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(client.subscribe(topic));
    Serial.println("subscribe to cmd OK");
  }
  else {
    Serial.println("subscribe to cmd FAILED"); }
}
void publishData()
{
  digitalWrite(trigPin1, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin1, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin1, LOW);
  duration1 = pulseIn(echoPin1, HIGH);
  distance1= duration1*0.034/2;
  delay(100);
  digitalWrite(trigPin2, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin2, HIGH);
  delayMicroseconds(10);
}

```

```

digitalWrite(trigPin2, LOW);
duration2 = pulseIn(echoPin2, HIGH);
distance2= duration2*0.034/2;
delay(100);

LCD.setCursor(0,1);
LCD.print("Fill Status ");
if(distance2>300 && distance2<=400){
    LCD.setCursor(12,1);
    LCD.print("25% ");
    String payload = "{\"Bin_Level\":\"";payload += "25";payload += "\",\"Bin_Load
\": \"";
    payload += "12.5";
    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");
    }
}
else if(distance2 > 200 && distance2<= 299){
    LCD.setCursor(12,1);
    LCD.print("50%");
    String payload = "{\"Bin_Level\":\"";payload += "50";payload += "\",\"Bin_Load
\": \"";
    payload += "25";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");
    }
}
else if(distance2 >50 && distance2<=199){
    LCD.setCursor(12,1);
    LCD.print("75%");
    String payload = "{\"Bin_Level\":\"";payload += "75";payload += "\",\"Bin_Load
\": \"";
    payload += "37.5";payload += "\"}";
    Serial.print("\n");
    Serial.print("Sending payload: ");
    Serial.println(payload);
    if (client.publish(publishTopic, (char*) payload.c_str())) {

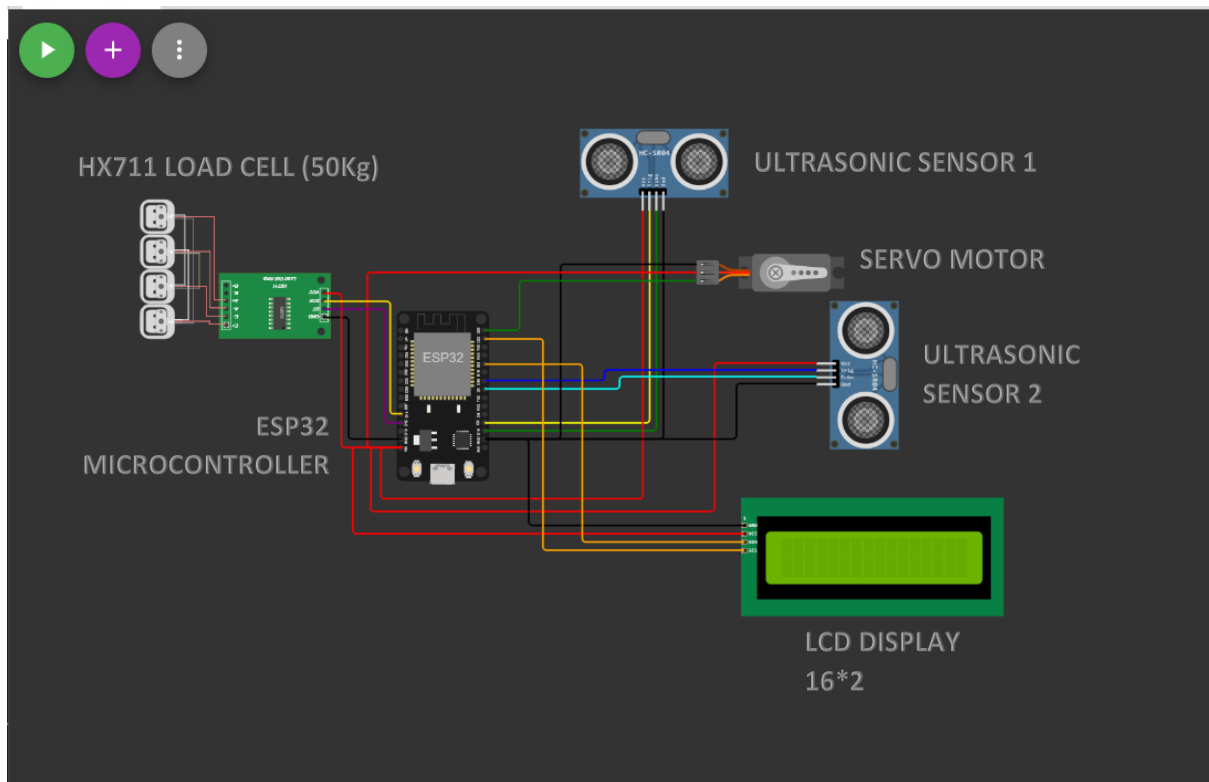
```

```

        Serial.println("Publish OK");
    } else {
        Serial.println("Publish FAILED");
    }
}
else{
    LCD.setCursor(12,1);
    LCD.print("100%");
    String payload = "{\"Bin_Level\":\"";
    payload += "100";payload += "\",\"Weight \":\""; payload += "50";payload +=
    "\"}";
    Serial.print("Sending payload: ");
    Serial.println(payload);
    if (client.publish(publishTopic, (char*) payload.c_str())) {
        Serial.println("Publish OK");
    } else {
        Serial.println("Publish FAILED");
    }
}
if(distance1<=50){
    LCD.setCursor(0,0);
    LCD.print("Dustbin is open ");
    servo.write(90);
}
else{
    LCD.setCursor(0,0);
    LCD.print("Dustbin is close ");
    servo.write(0);
}
}
}

```

CIRCUIT DIAGRAM:



OUTPUT:

The screenshot shows a simulation of the circuit. The LCD display shows the text: "Dustbin is open" and "Fill Status 25%". Below the simulation window, a log shows the following messages:

```
Connecting to Wifi...WiFi connected, IP address: 10.10.0.2
Reconnecting MQTT client to dluuhi.messaging.internetofthings.ibmcloud.com
subscribe to cmd OK

Sending payload: {"Bin_Level":50,"Bin_Load ":25}
Publish OK

Sending payload: {"Bin_Level":50,"Bin_Load ":25}
Publish OK
```

IBM WATSON IoT PLATFORM:

The screenshot shows the IBM Watson IoT Platform interface. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A search bar is present for finding devices by ID. The main content area displays a table of devices, with one device (ID: 12345) highlighted. Below the table, the 'Recent Events' tab is active, showing a live stream of data. The data points are as follows:

Event	Value	Format	Last Received
Data	{"Bin_Level":75,"Weight":37.5}	json	a few seconds ago
Data	{"Bin_Level":75,"Weight":37.5}	json	a few seconds ago
Data	{"Bin_Level":75,"Weight":37.5}	json	a few seconds ago
Data	{"Bin_Level":75,"Weight":37.5}	json	a few seconds ago
Data	{"Bin_Level":75,"Weight":37.5}	json	a few seconds ago

Node-red workflow:

The screenshot shows a Node-RED workflow within the IBM Watson IoT Platform. The workflow consists of the following components:

- IBM IoT Node:** A blue node labeled 'connected' that receives data from the IoT platform.
- Function Nodes:** Two orange nodes labeled 'Bin Level' and 'Weight' that process the incoming data.
- Output Node:** A green node labeled 'msg.payload' that outputs the processed data.

The workflow is configured to receive data from the IoT platform and process it using the 'Bin Level' and 'Weight' function nodes. The output is displayed in the 'debug' console on the right, showing the following data points:

```
11/18/2022, 4:25:38 PM node:7b95e1e07482b41b  
iot-2/type:NodeMcu/12345/evn/Data/fmt/json :  
msg.payload : Object  
{ Bin_Level: 25, Weight: 12.5 }  
11/18/2022, 4:25:39 PM node:7b95e1e07482b41b  
iot-2/type:NodeMcu/12345/evn/Data/fmt/json :  
msg.payload : number  
25  
11/18/2022, 4:25:40 PM node:7b95e1e07482b41b  
iot-2/type:NodeMcu/12345/evn/Data/fmt/json :  
msg.payload : number  
12.5  
11/18/2022, 4:25:41 PM node:7b95e1e07482b41b  
iot-2/type:NodeMcu/12345/evn/Data/fmt/json :  
msg.payload : Object  
{ Bin_Level: 25, Weight: 12.5 }  
11/18/2022, 4:25:42 PM node:7b95e1e07482b41b  
iot-2/type:NodeMcu/12345/evn/Data/fmt/json :  
msg.payload : number  
25  
11/18/2022, 4:25:42 PM node:7b95e1e07482b41b  
iot-2/type:NodeMcu/12345/evn/Data/fmt/json :  
msg.payload : number  
12.5
```

Node-red WEB UI:

Form

Log in

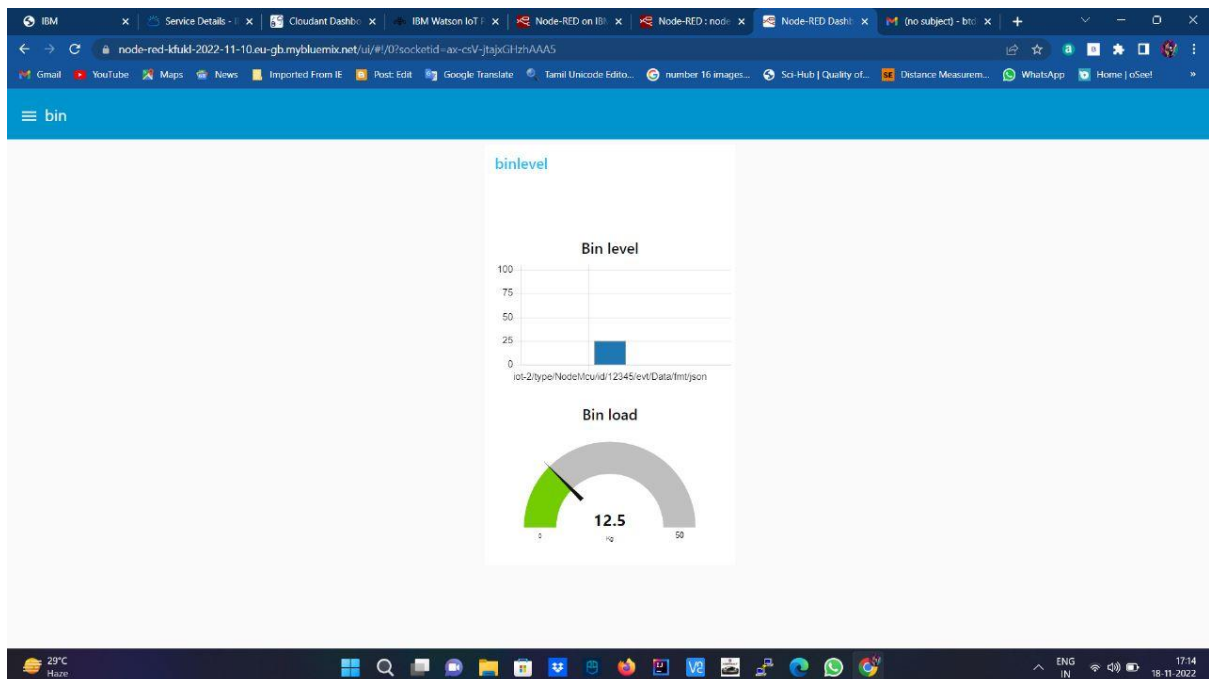
First name *
Loganath

Last name *
M

E-MAIL *
logan50@gmail.com

Password *

SUBMIT CANCEL



CLOUDANT DATABASE:

The screenshot shows the Cloudant Databases dashboard. The left sidebar contains navigation icons for Databases, Query, Permissions, Changes, and Design Documents. The main area displays a table of databases:

Name	Size	# of Docs	Partitioned	Actions
form	430 bytes	3	No	[Icons for query, lock, delete]
noderedkfukd20221110	31.9 KB	4	No	[Icons for query, lock, delete]

At the bottom, it says "Showing 1-2 of 2 databases. Databases per page: 20".

The screenshot shows the document view for the 'form' database. The left sidebar has a 'form' header and a list of actions: All Documents, Query, Permissions, Changes, and Design Documents. The main area displays a table of documents:

id	key	value
07126c92f381a24766411b4c32e938e2	07126c92f381a24766411b4c32e938e2	{ "rev": "1-85222aa40384e1308370db62568..." }
31eb66dc121df2469d2c0b3b89337294	31eb66dc121df2469d2c0b3b89337294	{ "rev": "1-85222aa40384e1308370db62568..." }
fe5618060ea1cf64211f58319bfaf29	fe5618060ea1cf64211f58319bfaf29	{ "rev": "1-85222aa40384e1308370db62568..." }

At the bottom, it says "Showing document 1 - 3. Documents per page: 20".

IBM Watson IoT Platform | Service Details - IBM | Overview | IBM Cloud | Cloudant Dashboard | Node-RED: node-red | Node-RED Dashboard

3e2f065c-7fcd-41f7-a9a5-41f38dc523e9-bluemix.cloudant.com/dashboard.html#database/form/31eb66dc121df2469d2c0b3b89337294

form > 31eb66dc121df2469d2c0b3b89337294

Save Changes Cancel Upload Attachment Clone Document Delete

```
1
2 {
3   "_id": "31eb66dc121df2469d2c0b3b89337294",
4   "_rev": "1-a03199a07bb197acca5c0457121ce786",
5   "payload": {
6     "First name": "Loganath",
7     "Last name": "M",
8     "E-Mail": "logan5@gmail.com",
9     "Password": "0987654321"
10  },
11   "socketid": "u0nx4ve4LegtjlqNAAB1"
12 }
```

Log Out

27°C
Haze

ENG
IN

18:31
18-11-2022