Project Development – Delivery of Sprint-4

Team ID	PNT2022TMID47715
Project Name	Project - Smart Waste Management System For Metropolitan Cities

Web User Interface

Make the user Interact with the Following Software as,

- Wokwi For Simulation
- IBM Watson IoT platform
- Node-RED and Dashboard
- MIT App Inventor

Wokwi for Simulation

The User Interact with Wokwi Software as where the IoT of Smart Waste Management System Contain ESP32 Microcontroller and Ultrasonic Sensor are Connected. These Sensor Were used to Calculate the Level of Garbage and can be Detected with Code.

The level of Garbage where user need to operate the Senor because of Simulation. Then This Wokwi were connected with IBM IoT Watson platform using code of device Type, ID, Authentication Token.

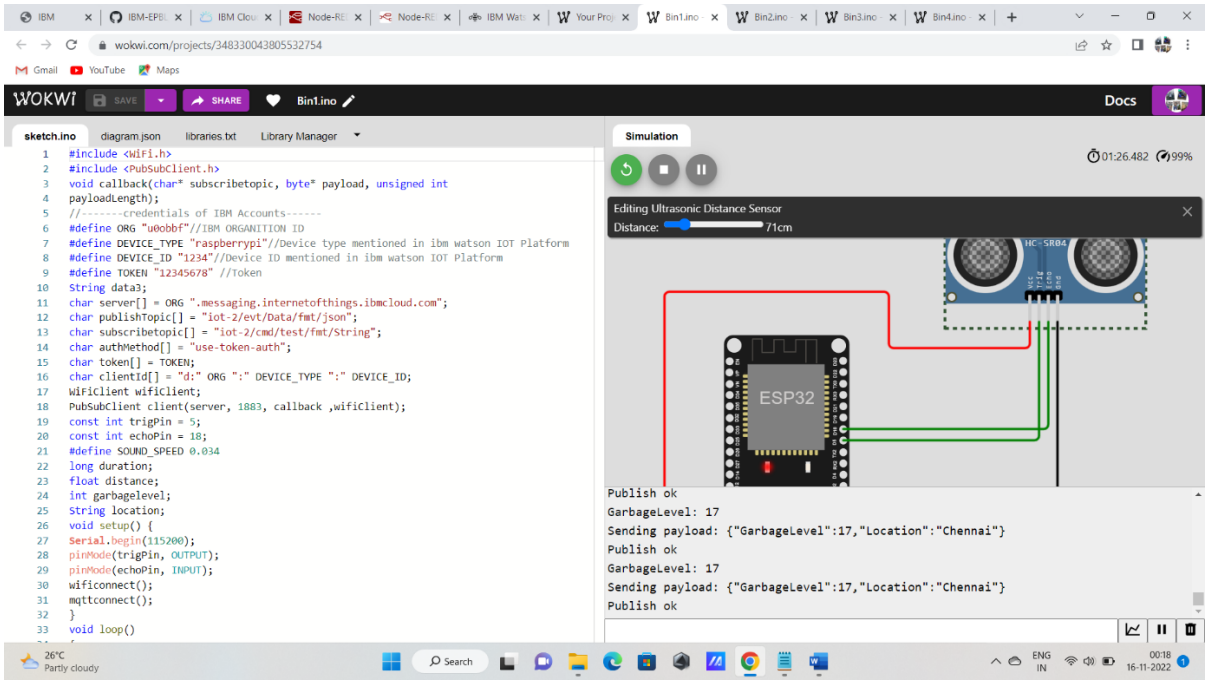
The Following Images Shows the User Interaction with Ultrasonic Sensor in Wokwi and the Link are given below.

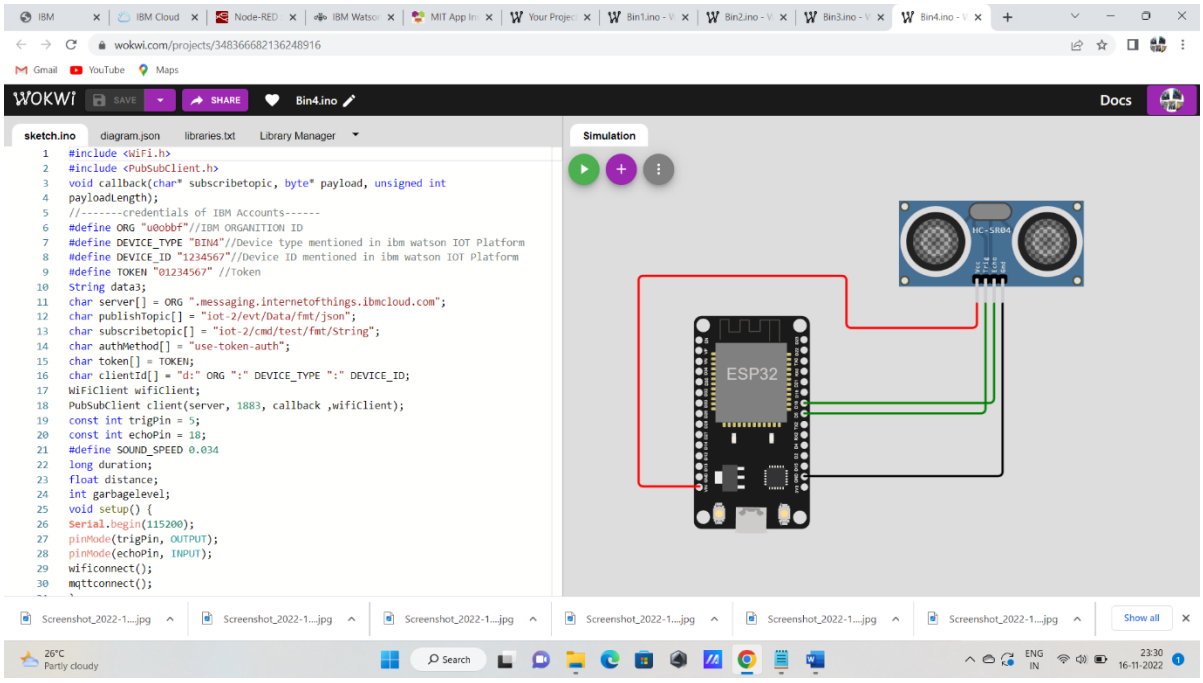
SmartBin 1: https://wokwi.com/projects/348330043805532754

SmartBin 2: https://wokwi.com/projects/348365535900074578

SmartBin 3: https://wokwi.com/projects/348366103082173011

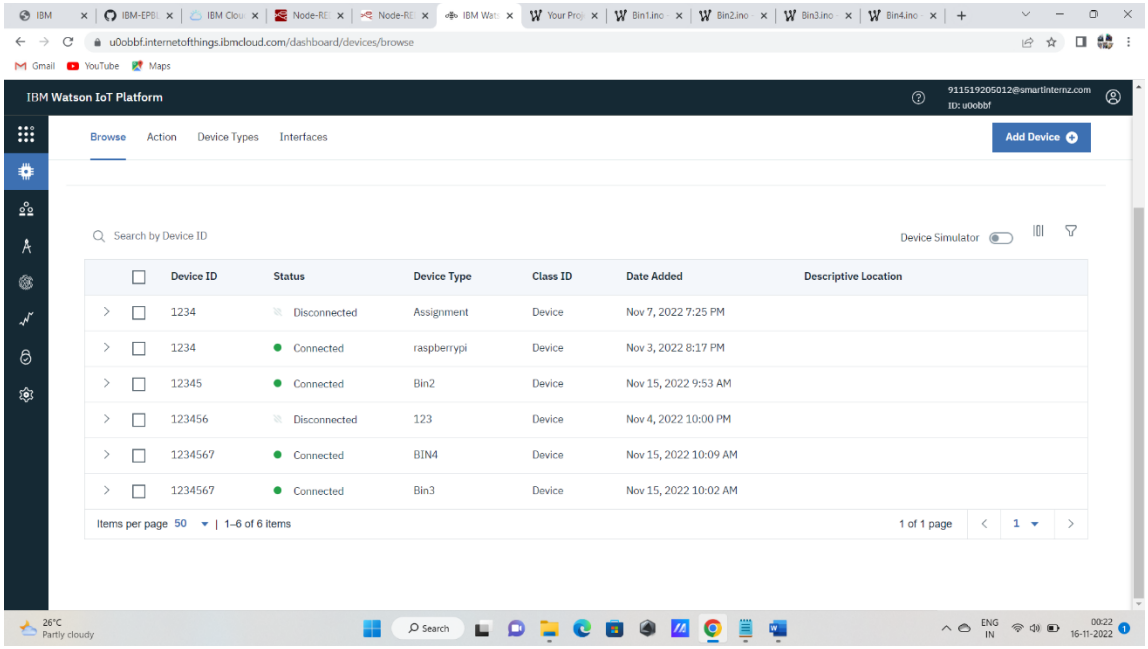
SmartBin 4: https://wokwi.com/projects/348366682136248916

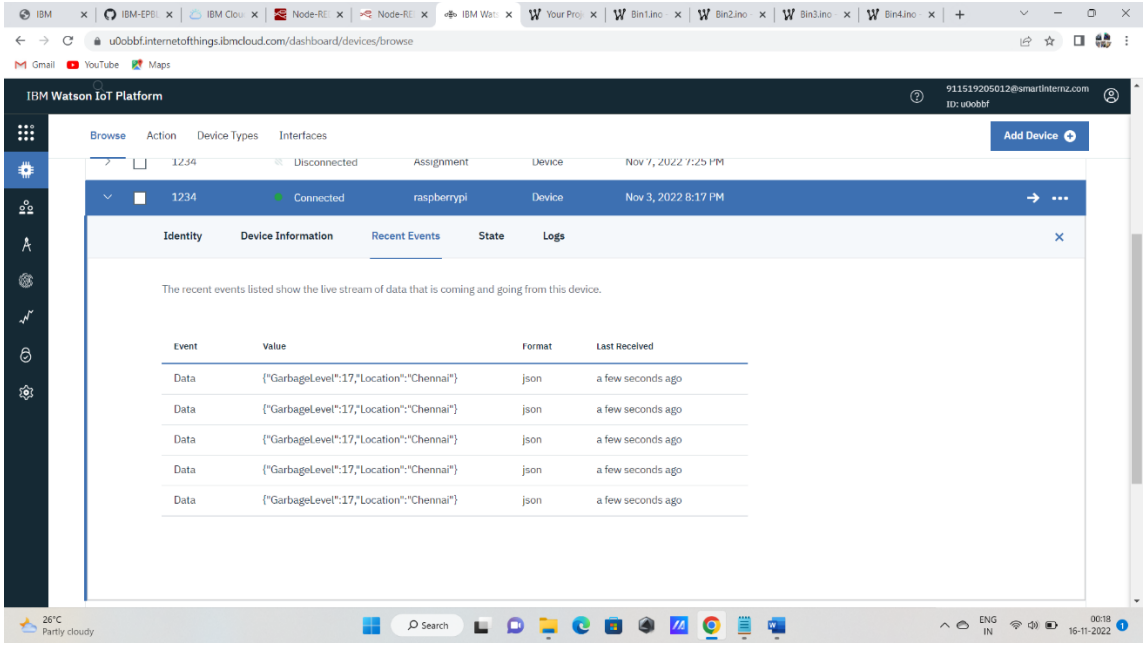

wokwi.com/projects/348330043805532754
WOKWI
SAVE
SHARE
Bin1.ino
Docs
sketch.ino
diagram.json
libraries.txt
Library Manager
#include <WiFi.h>
#include <PubSubClient.h>
void callback(char* subscribetopic, byte* payload, unsigned int
payloadLength);
//-------credentials of IBM Accounts------
#define ORG "u0obbf"//IBM ORGANITION ID
#define DEVICE_TYPE "raspberrypi"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "1234"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "12345678" //Token
String data3;
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Data/fmt/json";
char subscribetopic[] = "iot-2/cmd/test/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
WiFiClient wifiClient;
PubSubClient client(server, 1883, callback ,wifiClient);
const int trigPin = 5;
const int echoPin = 18;
#define SOUND_SPEED 0.034
long duration;
float distance;
int garbagelevel;
String location;
void setup() {
Serial.begin(115200);
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
wificonnect();
mqttconnect();
}
void loop()
Simulation
01:26.482
99%
Editing Ultrasonic Distance Sensor
Distance:
71cm
HC-SR04
ESP32
Publish ok
GarbageLevel: 17
Sending payload: {"GarbageLevel":17,"Location":"Chennai"}
Publish ok
GarbageLevel: 17
Sending payload: {"GarbageLevel":17,"Location":"Chennai"}
Publish ok
26°C
Partly cloudy
Search
ENG
IN
00:18
16-11-2022

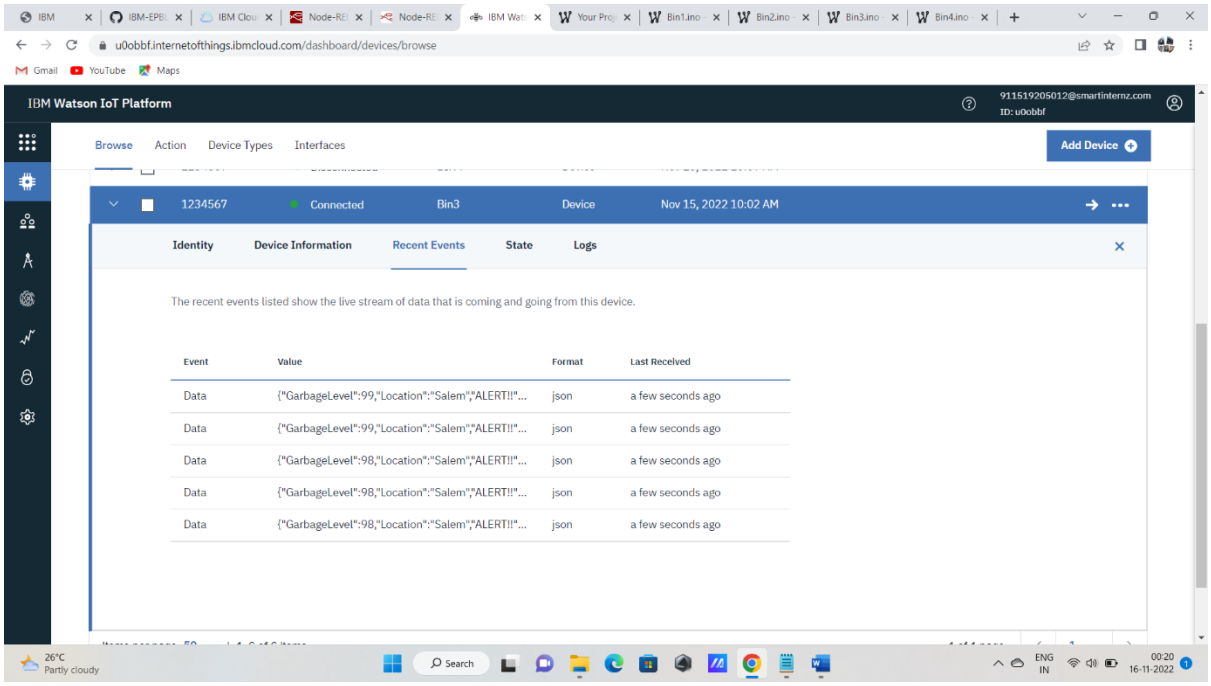

wokwi.com/projects/348366682136248916
WOKWI
SAVE
SHARE
Bin4.ino
Docs
sketch.ino
diagram.json
libraries.txt
Library Manager
#include <WiFi.h>
#include <PubSubClient.h>
void callback(char* subscribetopic, byte* payload, unsigned int
payloadLength);
//-------credentials of IBM Accounts------
#define ORG "u0obbf"//IBM ORGANITION ID
#define DEVICE_TYPE "BIN4"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "1234567"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "01234567" //Token
String data3;
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Data/fmt/json";
char subscribetopic[] = "iot-2/cmd/test/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
WiFiClient wifiClient;
PubSubClient client(server, 1883, callback ,wifiClient);
const int trigPin = 5;
const int echoPin = 18;
#define SOUND_SPEED 0.034
long duration;
float distance;
int garbagelevel;
void setup() {
Serial.begin(115200);
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
wificonnect();
mqttconnect();
Simulation
HC-SR04
ESP32
Screenshot_2022-1....jpg
Show all
26°C
Partly cloudy
Search
ENG
IN
23:30
16-11-2022

IBM Watson IoT Platform:

Create a device in IBM Watson IoT platform and Connect to the Wokwi. The Data's from Wokwi were send to IoT Watson Platform and The User can be seen and detected.

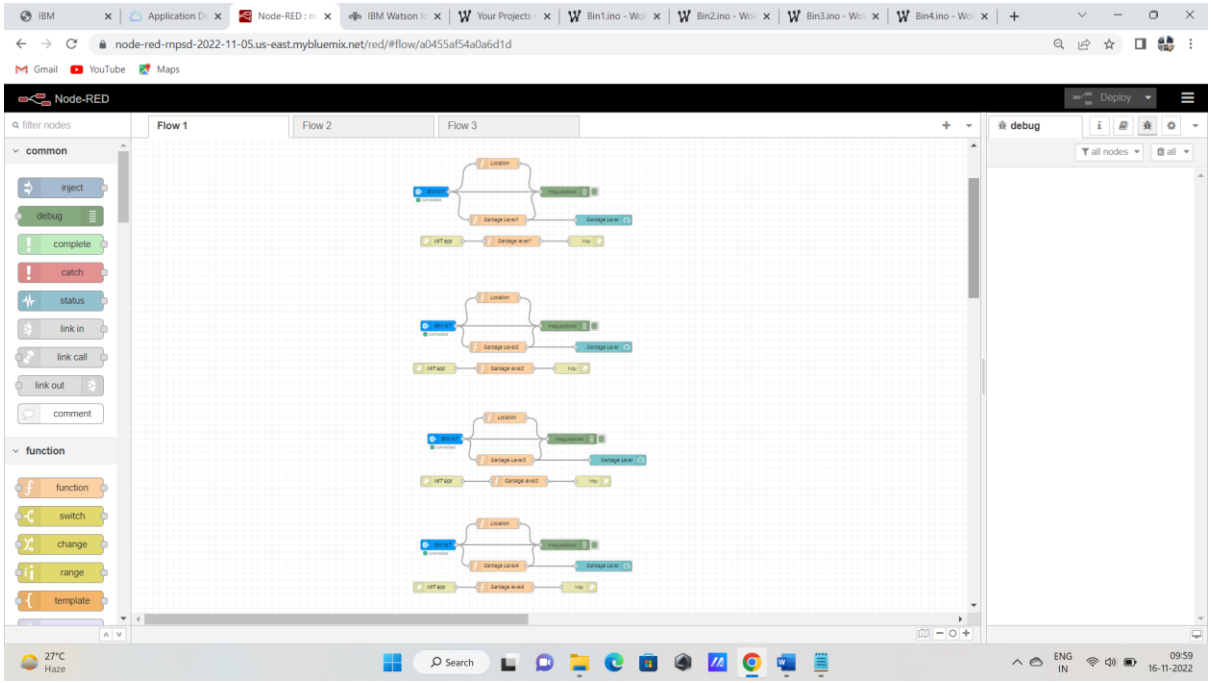


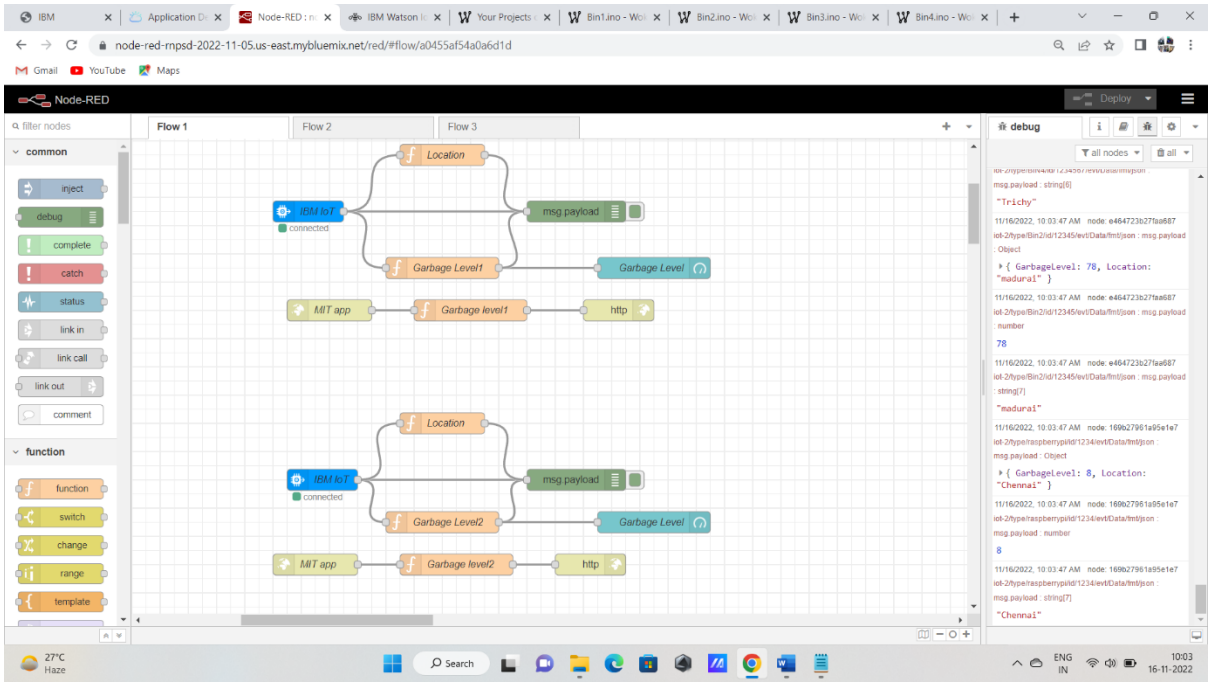


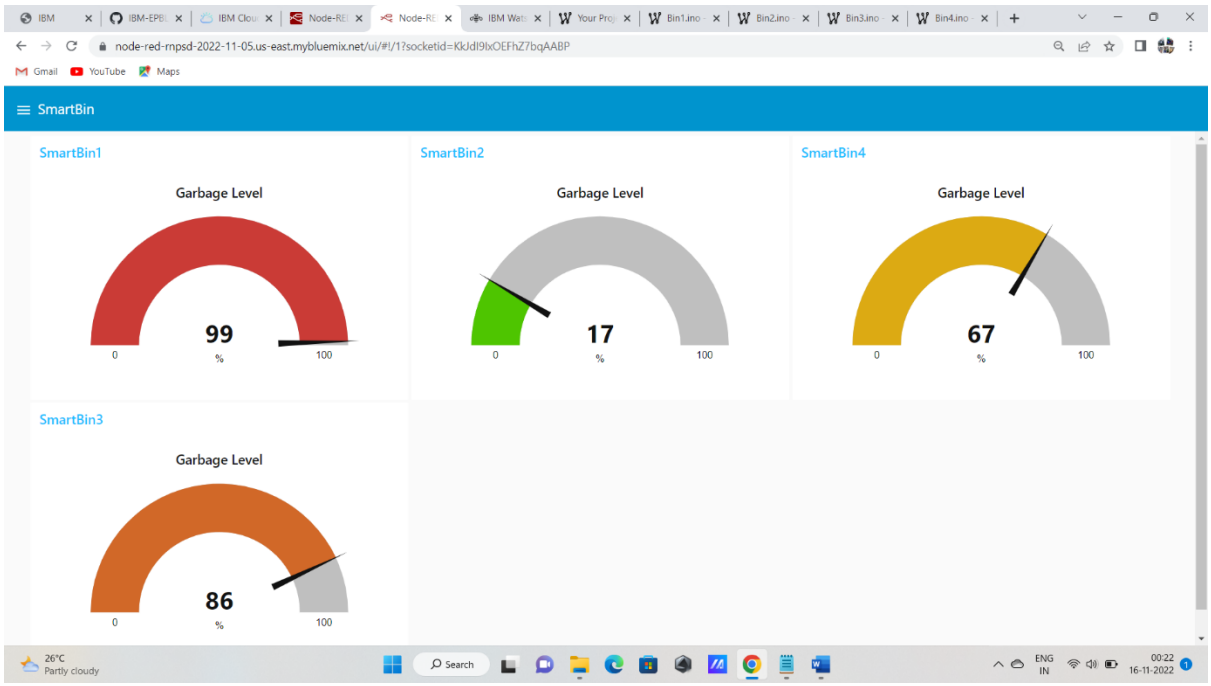


Node-RED and Dashboard

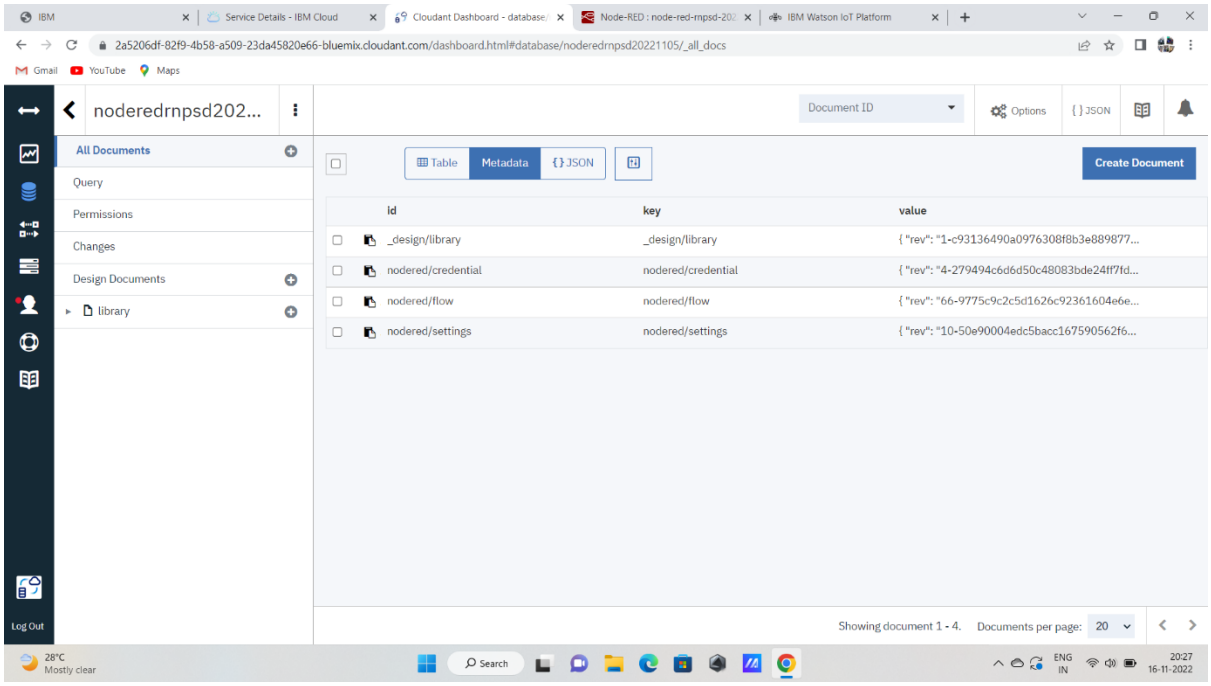
The Node-RED which is used for workflow as where the IBM Watson IoT platform and MIT app Inventor are connected with Nodes. This IBM IoT was connected with Wokwi. The Node-RED contain Dashboard as Chart for Level of the Garbage.





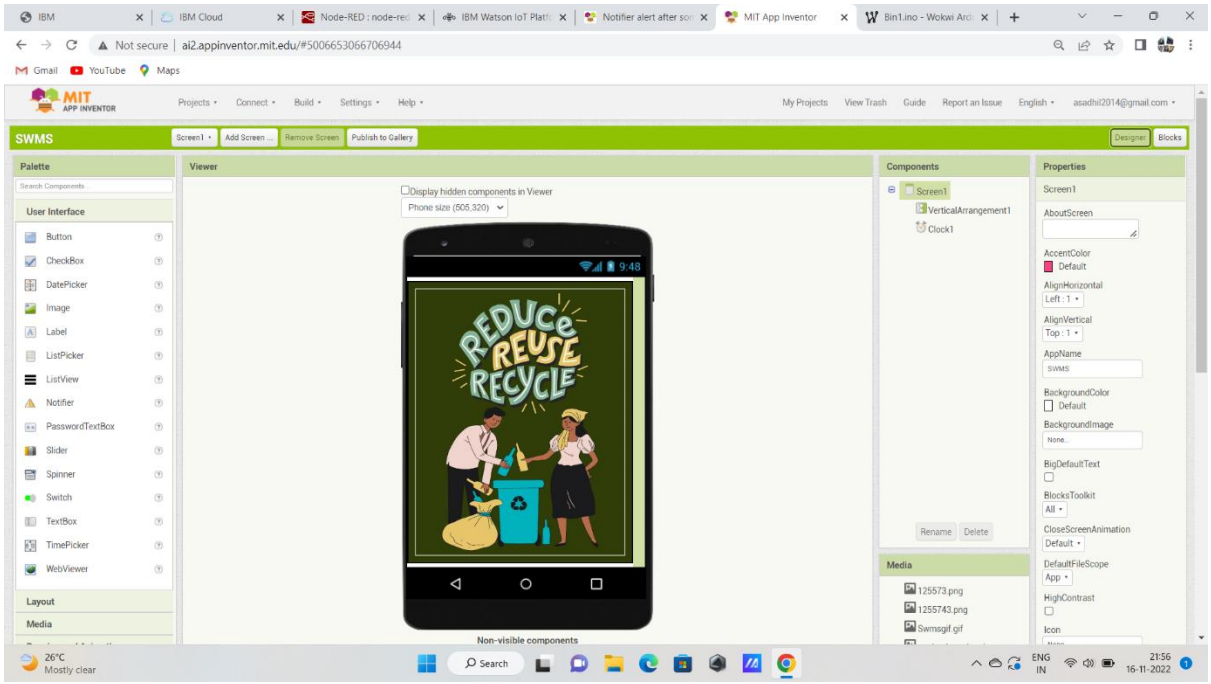


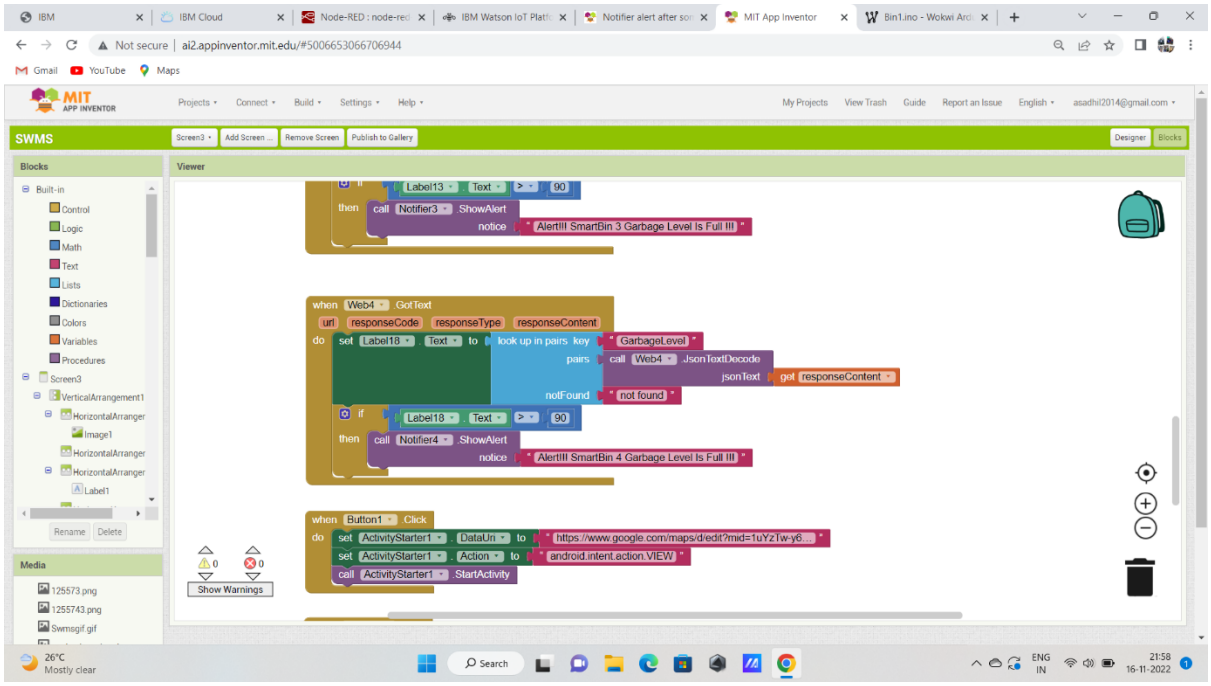
The IBM Cloud Contain Cloudant DB as where the Node-RED Connection and workflow are Noted with Documents and Links. The Following Image Shown as,



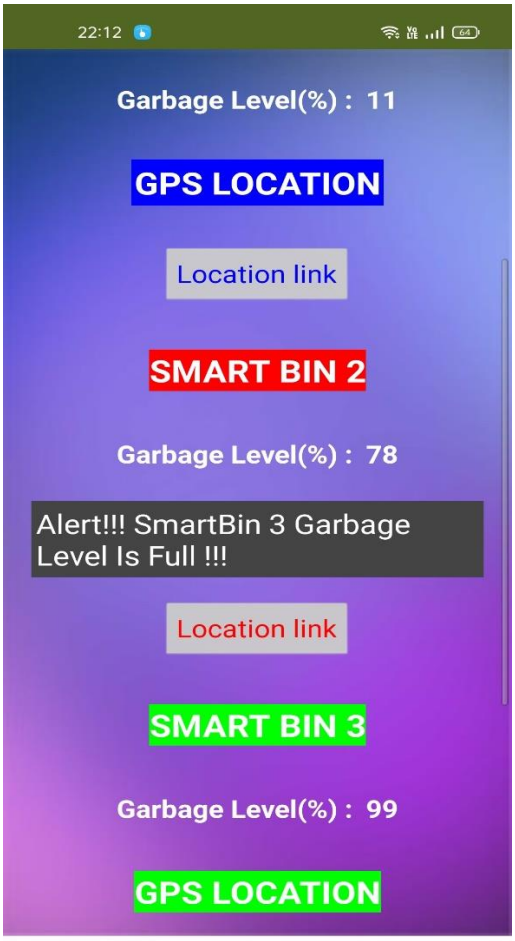
MIT App Inventor

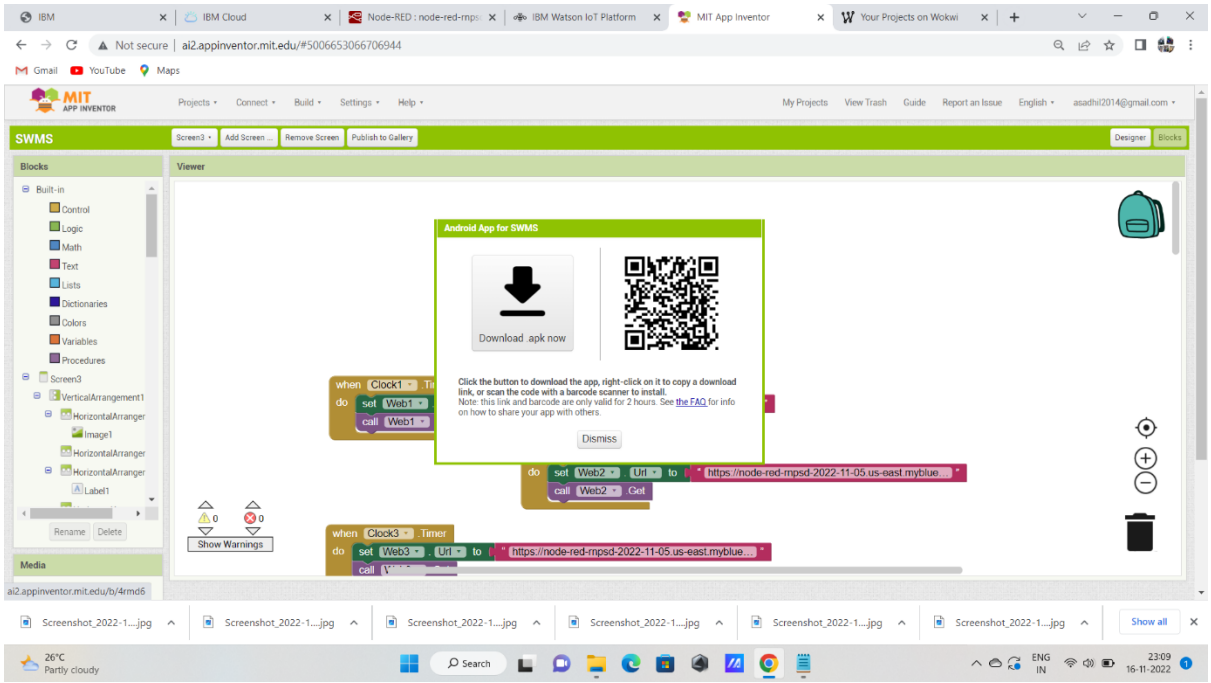
With Help of MIT App Inventor, The User can Download the app in mobile and can be Executed to See the Level of Garbage with Alert Message and Location(Random).




MIT App Inventor
SWMS
Blocks
Viewer
when Web4 .GotText
call Notifier3 .ShowAlert
notice "Alert!!! SmartBin 3 Garbage Level Is Full !!!"
call Notifier4 .ShowAlert
notice "Alert!!! SmartBin 4 Garbage Level Is Full !!!"
when Button1 .Click
call ActivityStarter1 .StartActivity
Show Warnings


22:11
SMART WASTE MANAGEMENT SYSTEM FOR METROPOLTIAN CITIES
USERNAME :
PASSWORD :
SUBMIT


22:12
Garbage Level(%) : 11
GPS LOCATION
Location link
SMART BIN 2
Garbage Level(%) : 78
Alert!!! SmartBin 3 Garbage Level Is Full !!!
Location link
SMART BIN 3
Garbage Level(%) : 99
GPS LOCATION


IBM
IBM Cloud
Node-RED : node-red-mps
IBM Watson IoT Platform
MIT App Inventor
Your Projects on Wokwi
Not secure | ai2.appinventor.mit.edu/#5006653066706944
Gmail
YouTube
Maps
MIT APP INVENTOR
Projects
Connect
Build
Settings
Help
My Projects
View Trash
Guide
Report an Issue
English
asadhil2014@gmail.com
SWMS
Screen3
Add Screen ...
Remove Screen
Publish to Gallery
Designer
Blocks
Blocks
Built-in
Control
Logic
Math
Text
Lists
Dictionaries
Colors
Variables
Procedures
Screen3
VerticalArrangement1
HorizontalArranger
Image1
HorizontalArranger
HorizontalArranger
Label1
Rename
Delete
Media
Viewer
Android App for SWMS
Download .apk now
Click the button to download the app, right-click on it to copy a download link, or scan the code with a barcode scanner to install.
Note: this link and barcode are only valid for 2 hours. See the FAQ for info on how to share your app with others.
Dismiss
when Clock1 .Ti
do set Web1
call Web1
do set Web2 Url to " https://node-red-mpsd-2022-11-05.us-east.myblue... "
call Web2 .Get
when Clock3 .Timer
do set Web3 Url to " https://node-red-mpsd-2022-11-05.us-east.myblue... "
Show Warnings
ai2.appinventor.mit.edu/b/4rmd6
Screenshot_2022-1....jpg
Screenshot_2022-1....jpg
Screenshot_2022-1....jpg
Screenshot_2022-1....jpg
Screenshot_2022-1....jpg
Show all
26°C
Partly cloudy
Search
ENG
IN
23:09
16-11-2022