Project Development phase
SPRINT 3

Date	12 November 2022
Team ID	PNT2022TMID15973
Project Name	Project – Smart Waste Management for Metropolitan cities
Maximum Marks	2 Marks

In this Phase, I will explain about the flow of our project.

- As we mentioned in the Data flow graph, we are first using online simulation tool to send the level of the dustbin with the help of ultrasonic sensor using WOKWI platform and we also send the required data such as location, bin name etc...
- This data is being sent to the IBM Watson IOT platform and which the help of IBM Watson IOT node we can get the data in node red.
- We designed few flows to make the data to be in a required format like maps, tables, gauge.
- Here we store the Admin, Co admin, Truck driver details in the database (Cloudant DB)
- We also store the Timings of the BIN which is being filled for future calculations.
- We have also created a python script to generate random BIN values which can also be used instead of WOKWI to send data to the IBM Watson IOT platform.
- I've also added few Screenshots of the things we have done.
- And In Bin database, we also delete the data from the database when the number of elements is above 11 for making use of the space effectively.
- We used world map node for displaying the latitude and longitude in the Map.

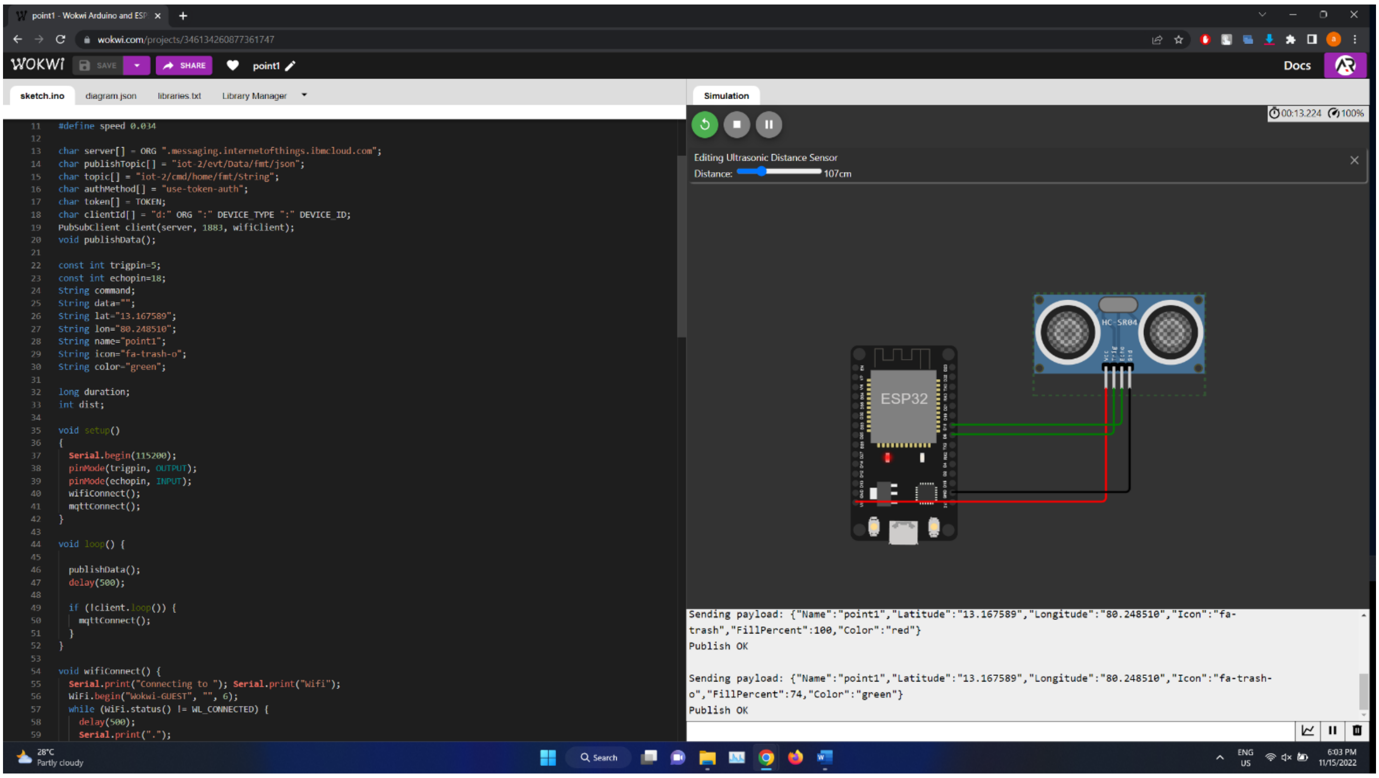
These are the things we have done in our project in Sprint Phase 3

SCREENSHOTS:

1) WOKWI Platform:

As we move the slider of the ultrasonic sensor. The value of the bin get changed

For **BIN 1**:



```
#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=5;
const int echopin=18;
String command;
String data="";
String lat="13.167589";
String lon="80.248510";
String name="point1";
String icon="fa-trash-o";
String color="green";

long duration;
int dist;

void setup()
{
  Serial.begin(115200);
  pinMode(trigpin, OUTPUT);
  pinMode(echopin, 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(".");
```

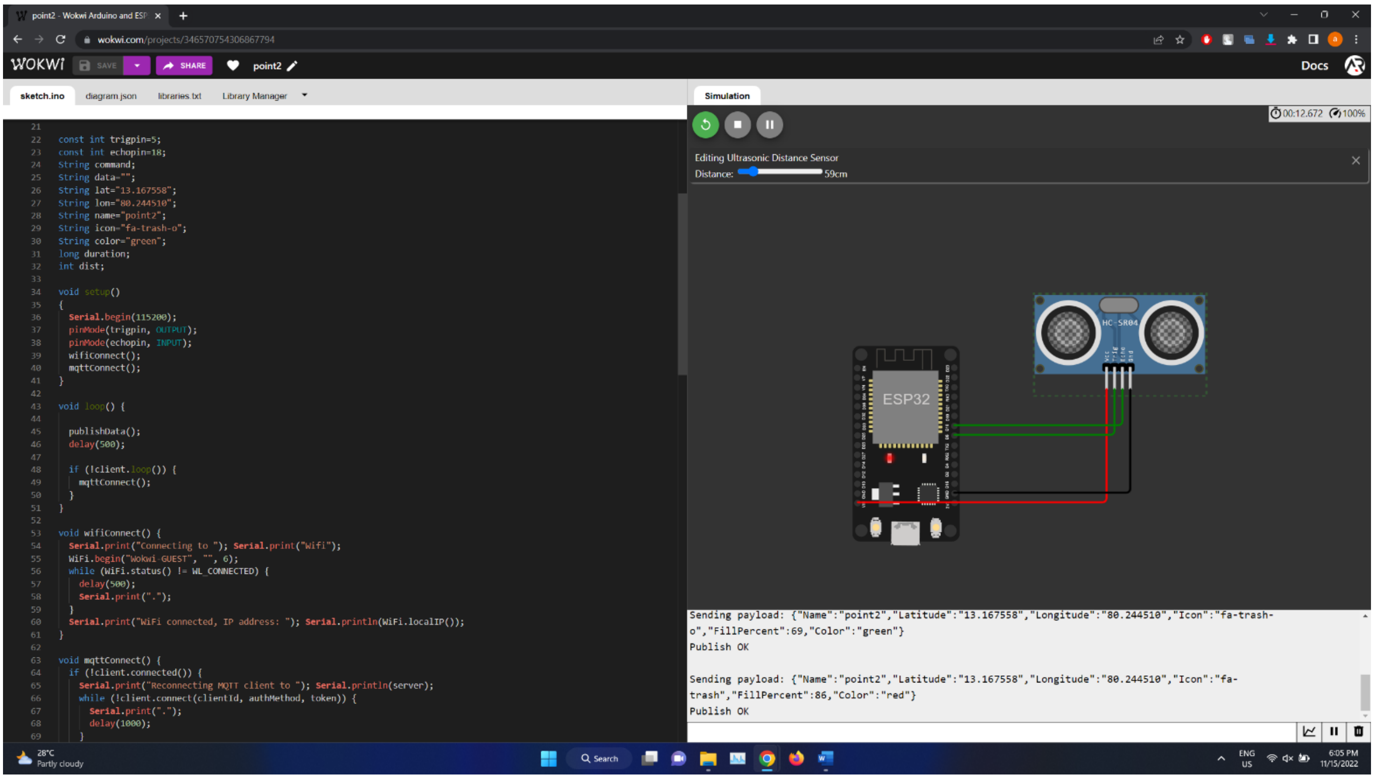
Editing Ultrasonic Distance Sensor

Distance: 107cm

```
Sending payload: {"Name":"point1","Latitude":"13.167589","Longitude":"80.248510","Icon":"fa-
trash","FillPercent":100,"Color":"red"}
Publish OK

Sending payload: {"Name":"point1","Latitude":"13.167589","Longitude":"80.248510","Icon":"fa-trash-
o","FillPercent":74,"Color":"green"}
Publish OK
```

For **BIN 2**:



```

const int trigpin=5;
const int echopin=18;
String command;
String data="";
String lat="13.167558";
String lon="80.244510";
String name="point2";
String icon="fa-trash-o";
String color="green";
long duration;
int dist;

void setup()
{
  Serial.begin(115200);
  pinMode(trigpin, OUTPUT);
  pinMode(echopin, 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(1000);
    }
```

Editing Ultrasonic Distance Sensor

Distance: 59cm

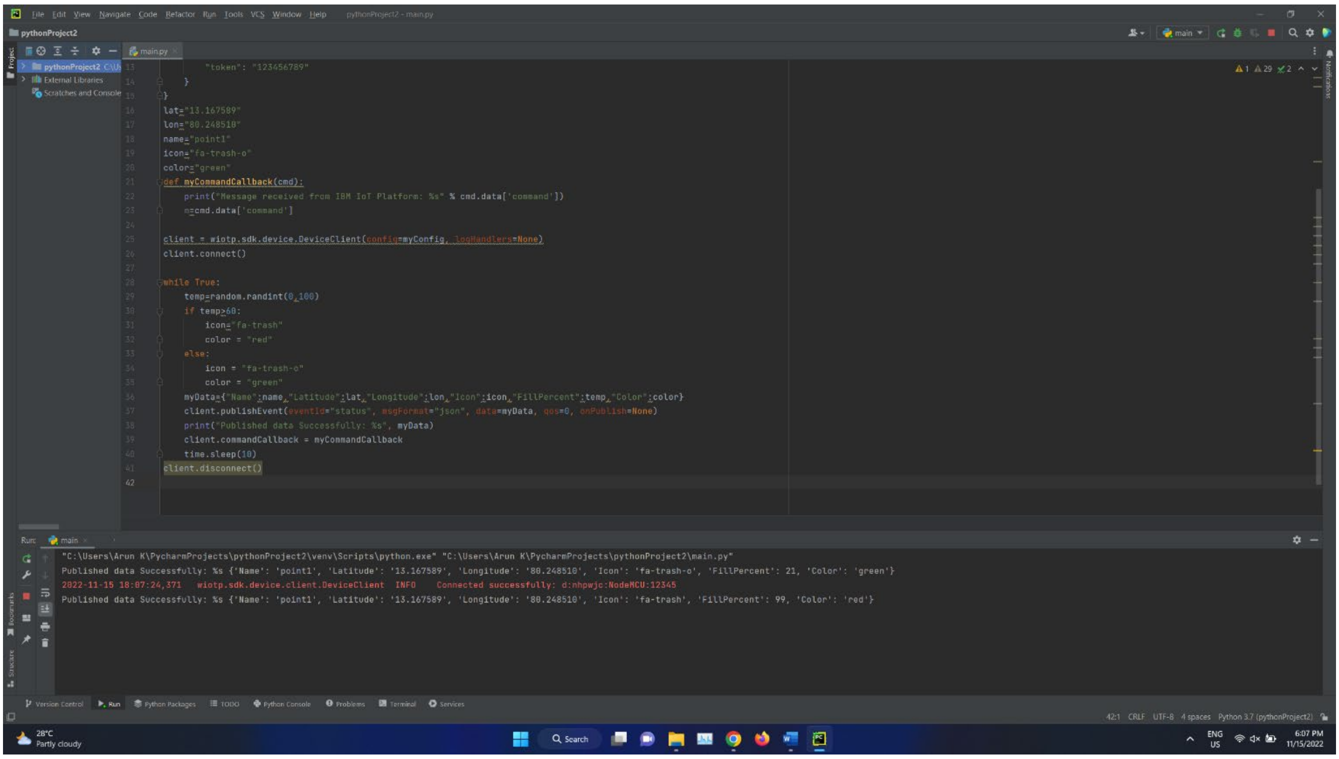
```
Sending payload: {"Name":"point2","Latitude":"13.167558","Longitude":"80.244510","Icon":"fa-trash-
o","FillPercent":69,"Color":"green"}
Publish OK

Sending payload: {"Name":"point2","Latitude":"13.167558","Longitude":"80.244510","Icon":"fa-
trash","FillPercent":86,"Color":"red"}
Publish OK
```

2) Python Code:

Here we can see the Python Code which is used to connect with IBM Watson IOT platform.

For BIN 1:



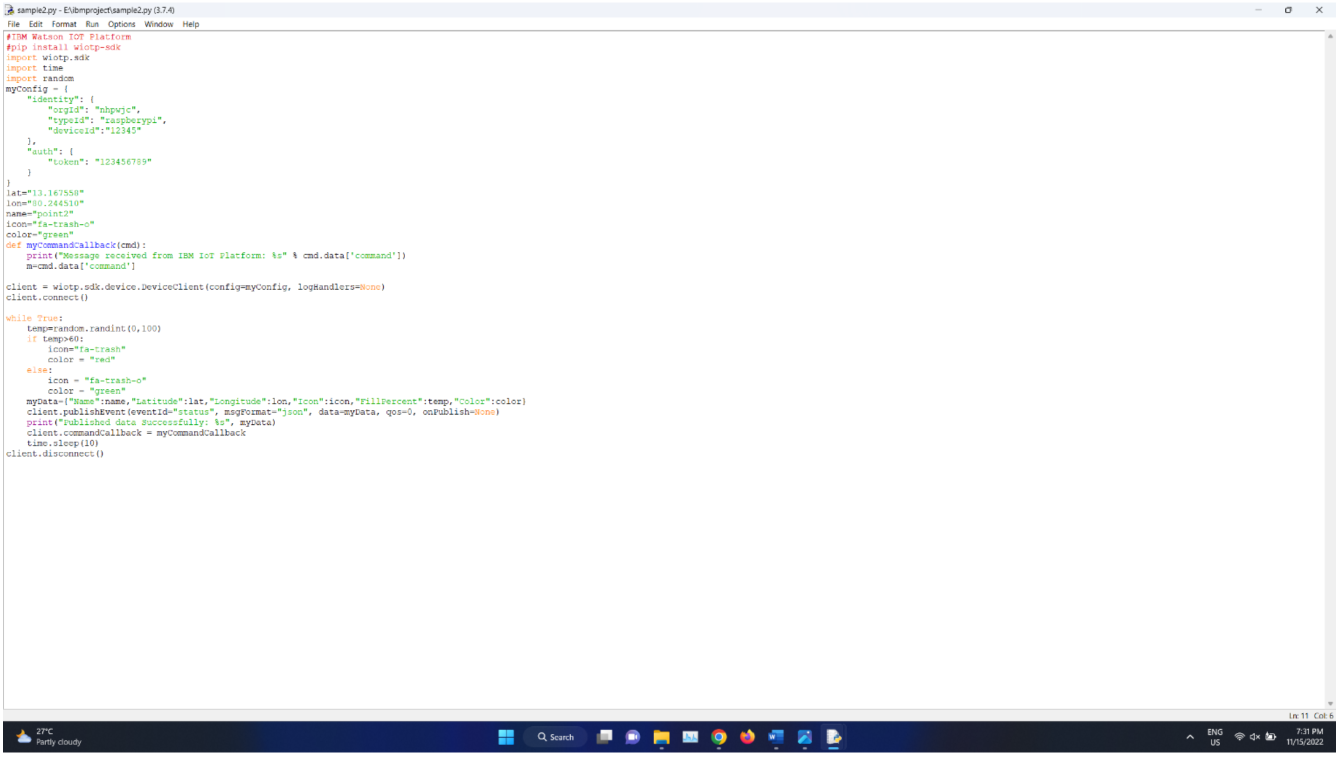
```
            "token": "123456789"
        }
}
lat="13.167589"
lon="80.248510"
name="point1"
icon="fa-trash-o"
color="green"
def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
    m=cmd.data['command']

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

while True:
    temp=random.randint(0,100)
    if temp>60:
        icon="fa-trash"
        color = "red"
    else:
        icon = "fa-trash-o"
        color = "green"
    myData={"Name":name,"Latitude":lat,"Longitude":lon,"Icon":icon,"FillPercent":temp,"Color":color}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
    print("Published data Successfully: %s", myData)
    client.commandCallback = myCommandCallback
    time.sleep(10)
client.disconnect()
```

```
"C:\Users\Arun K\PycharmProjects\pythonProject2\venv\Scripts\python.exe" "C:\Users\Arun K\PycharmProjects\pythonProject2\main.py"
Published data Successfully: %s {'Name': 'point1', 'Latitude': '13.167589', 'Longitude': '80.248510', 'Icon': 'fa-trash-o', 'FillPercent': 21, 'Color': 'green'}
2022-11-15 18:07:24,371   wiotp.sdk.device.client.DeviceClient  INFO    Connected successfully: d:nhpwjc:NodeMCU:12345
Published data Successfully: %s {'Name': 'point1', 'Latitude': '13.167589', 'Longitude': '80.248510', 'Icon': 'fa-trash', 'FillPercent': 99, 'Color': 'red'}
```

For BIN 2:


sample2.py - E:\ibmproject\sample2.py (3.7.4)
File Edit Format Run Options Window Help

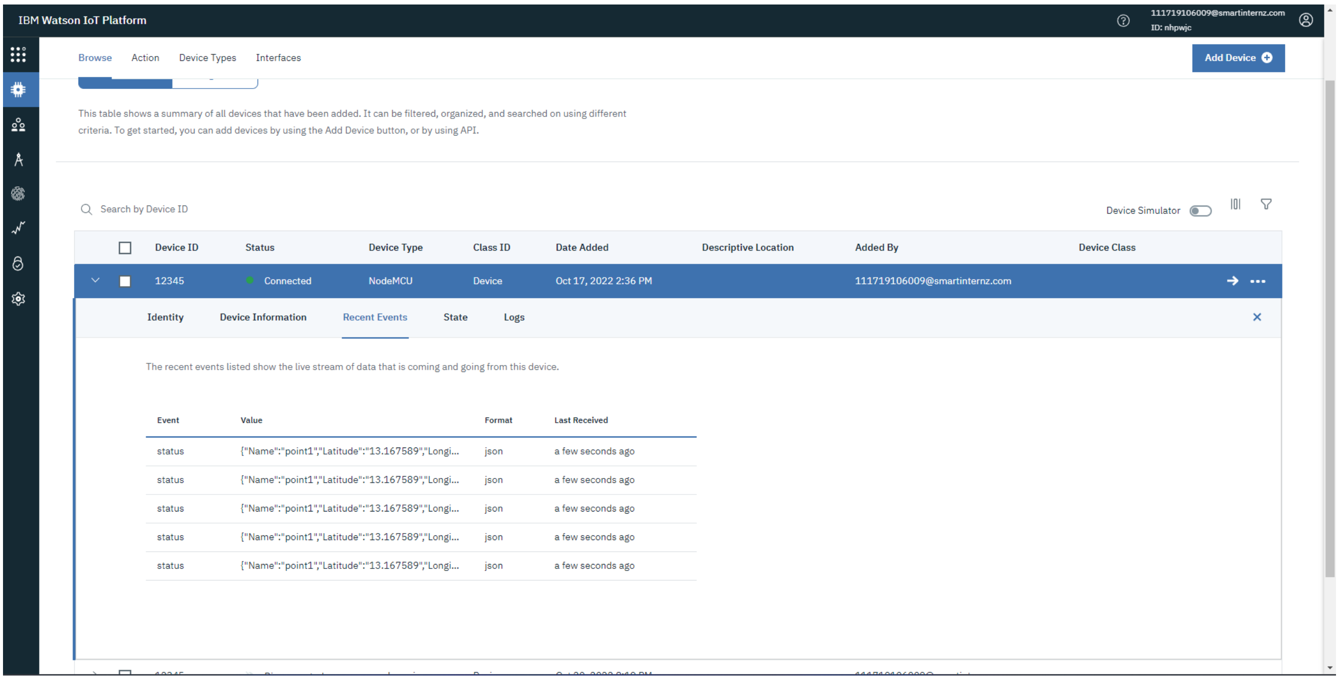
```
#IBM Watson IOT Platform
#pip install wiotp-sdk
import wiotp.sdk
import time
import random
myConfig = {
    "identity": {
        "orgId": "nhpwjc",
        "typeId": "raspberypi",
        "deviceId":"12345"
    },
    "auth": {
        "token": "123456789"
    }
}
lat="13.167558"
lon="80.244510"
name="point2"
icon="fa-trash-o"
color="green"
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()

while True:
    temp=random.randint(0,100)
    if temp>60:
        icon="fa-trash"
        color = "red"
    else:
        icon = "fa-trash-o"
        color = "green"
    myData={"Name":name,"Latitude":lat,"Longitude":lon,"Icon":icon,"FillPercent":temp,"Color":color}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
    print("Published data Successfully: %s", myData)
    client.commandCallback = myCommandCallback
    time.sleep(10)
client.disconnect()
```

3) IBM Watson IOT platform:

Here we can see the output which has been passed from WOKWI Platform or Python Script to IBM Watson IOT platform.



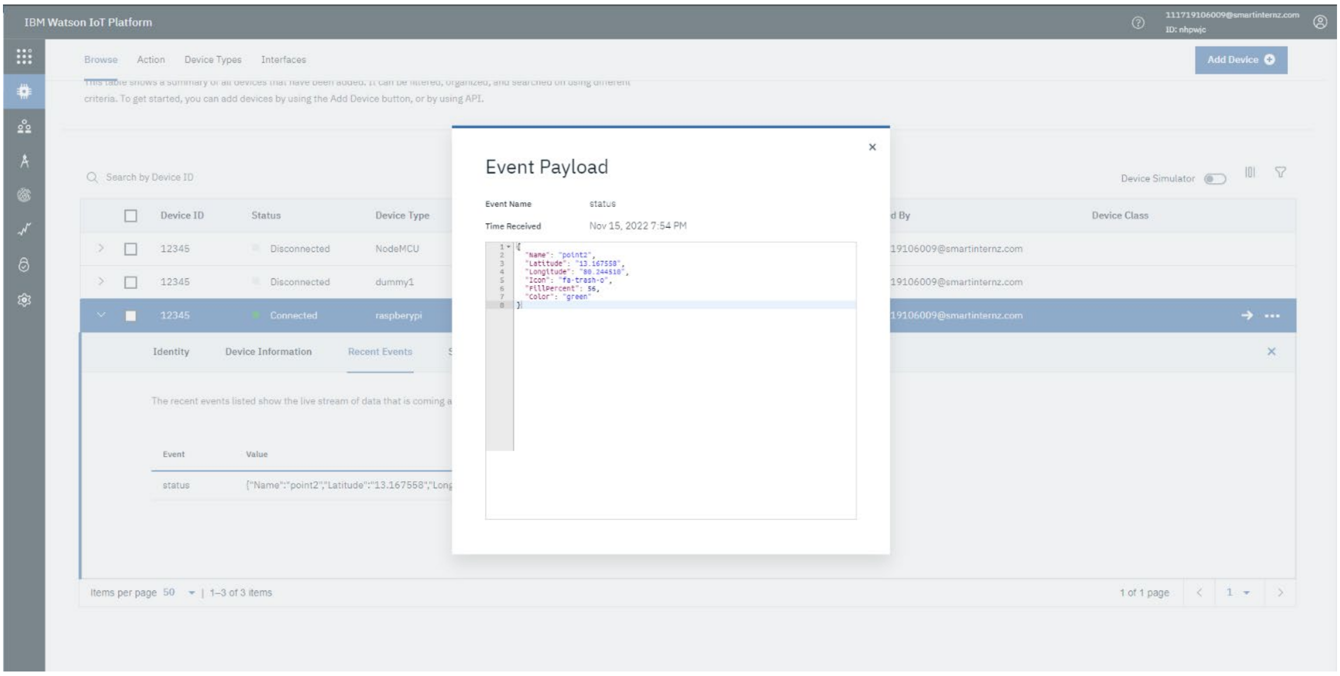
This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location	Added By	Device Class
12345	Connected	NodeMCU	Device	Oct 17, 2022 2:36 PM		111719106009@smartinternz.com	

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

Event	Value	Format	Last Received
status	{"Name":"point1","Latitude":"13.167589","Longi...	json	a few seconds ago
status	{"Name":"point1","Latitude":"13.167589","Longi...	json	a few seconds ago
status	{"Name":"point1","Latitude":"13.167589","Longi...	json	a few seconds ago
status	{"Name":"point1","Latitude":"13.167589","Longi...	json	a few seconds ago
status	{"Name":"point1","Latitude":"13.167589","Longi...	json	a few seconds ago

The Value which is passed is shown here.



criteria. To get started, you can add devices by using the Add Device button, or by using API.

Device ID	Status	Device Type	d By	Device Class
12345	Disconnected	NodeMCU	19106009@smartinternz.com	
12345	Disconnected	dummy1	19106009@smartinternz.com	
12345	Connected	raspberypi	19106009@smartinternz.com	

The recent events listed show the live stream of data that is coming a

Event	Value
status	{"Name":"point2","Latitude":"13.167558","Long

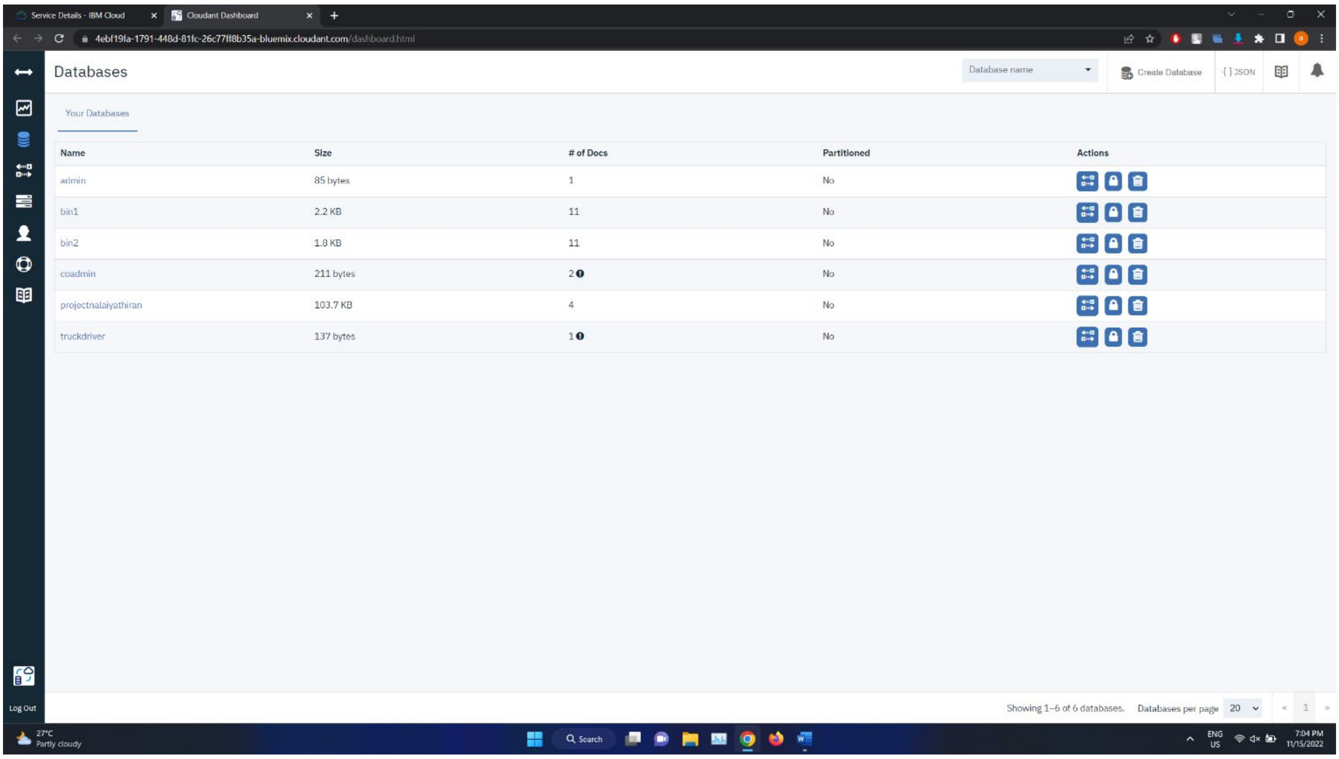
Event Payload

Event Name status

Time Received Nov 15, 2022 7:54 PM

```
{
  "Name": "point2",
  "Latitude": "13.167558",
  "Longitude": "80.244510",
  "Icon": "fa-trash-o",
  "FillPercent": 56,
  "Color": "green"
}
```

4) Cloudant DB:

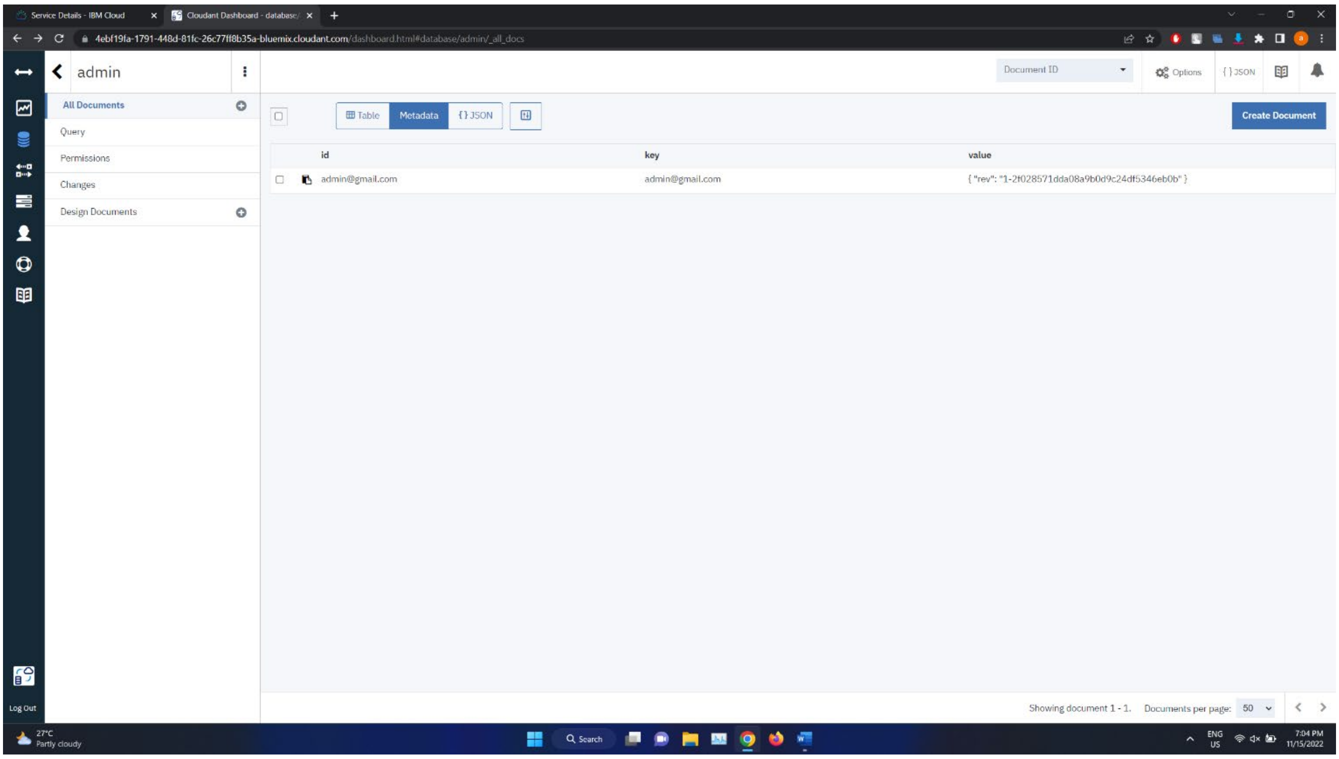


Service Details - IBM Cloud
Cloudant Dashboard
4ebf19fa-1791-448d-81fc-26c77ff8b35a-bluemix.cloudant.com/dashboard.html
Databases
Database name
Create Database
{ } JSON
Your Databases
Log Out

Name	Size	# of Docs	Partitioned	Actions
admin	85 bytes	1	No	
bin1	2.2 KB	11	No	
bin2	1.8 KB	11	No	
coadmin	211 bytes	2	No	
projectnalaiyathiran	103.7 KB	4	No	
truckdriver	137 bytes	1	No	

Showing 1–6 of 6 databases. Databases per page 20

Admin Database:

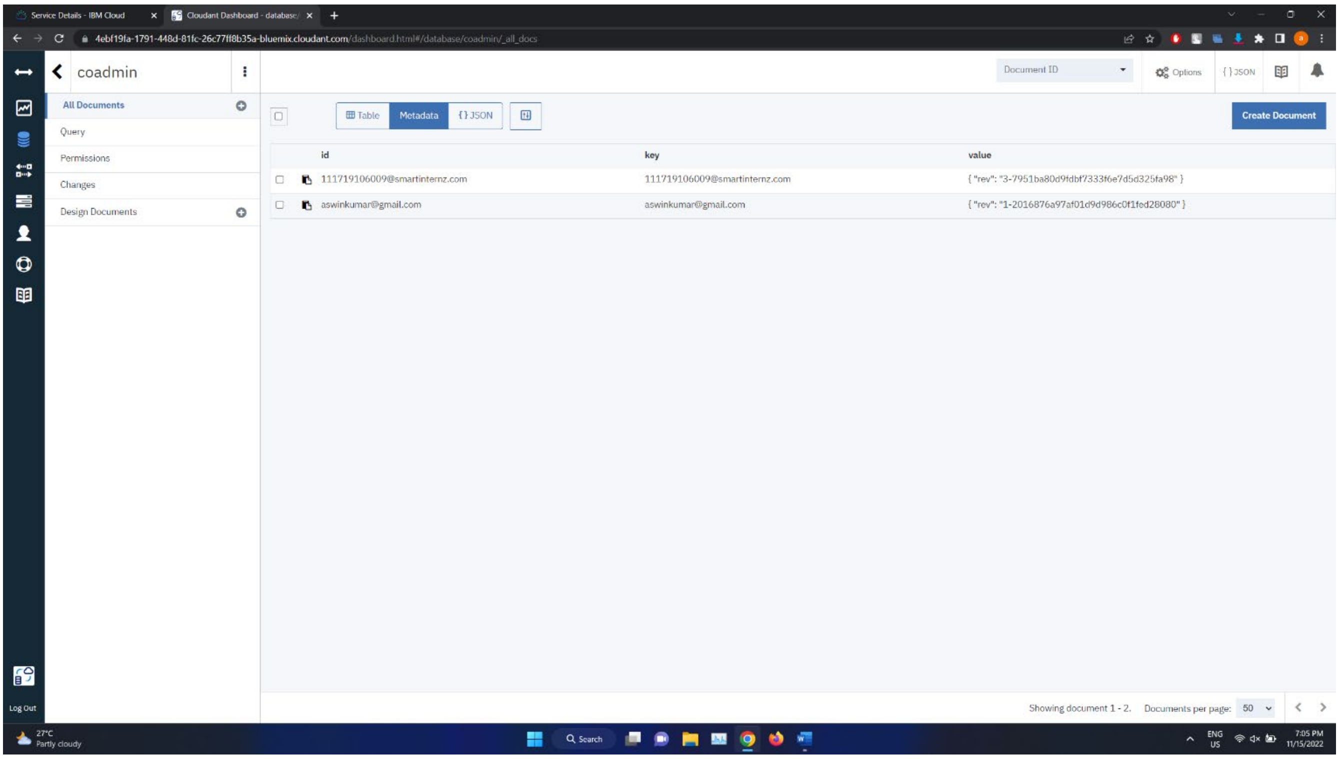


Service Details - IBM Cloud
Cloudant Dashboard - database/
4ebf19fa-1791-448d-81fc-26c77ff8b35a-bluemix.cloudant.com/dashboard.html#database/admin/_all_docs
admin
Document ID
Options
{ } JSON
All Documents
Query
Permissions
Changes
Design Documents
Table
Metadata
{ } JSON
Create Document
Log Out

id	key	value
admin@gmail.com	admin@gmail.com	{ "rev": "1-2f028571dda08a9b0d9c24df5346eb0b" }

Showing document 1 - 1. Documents per page: 50

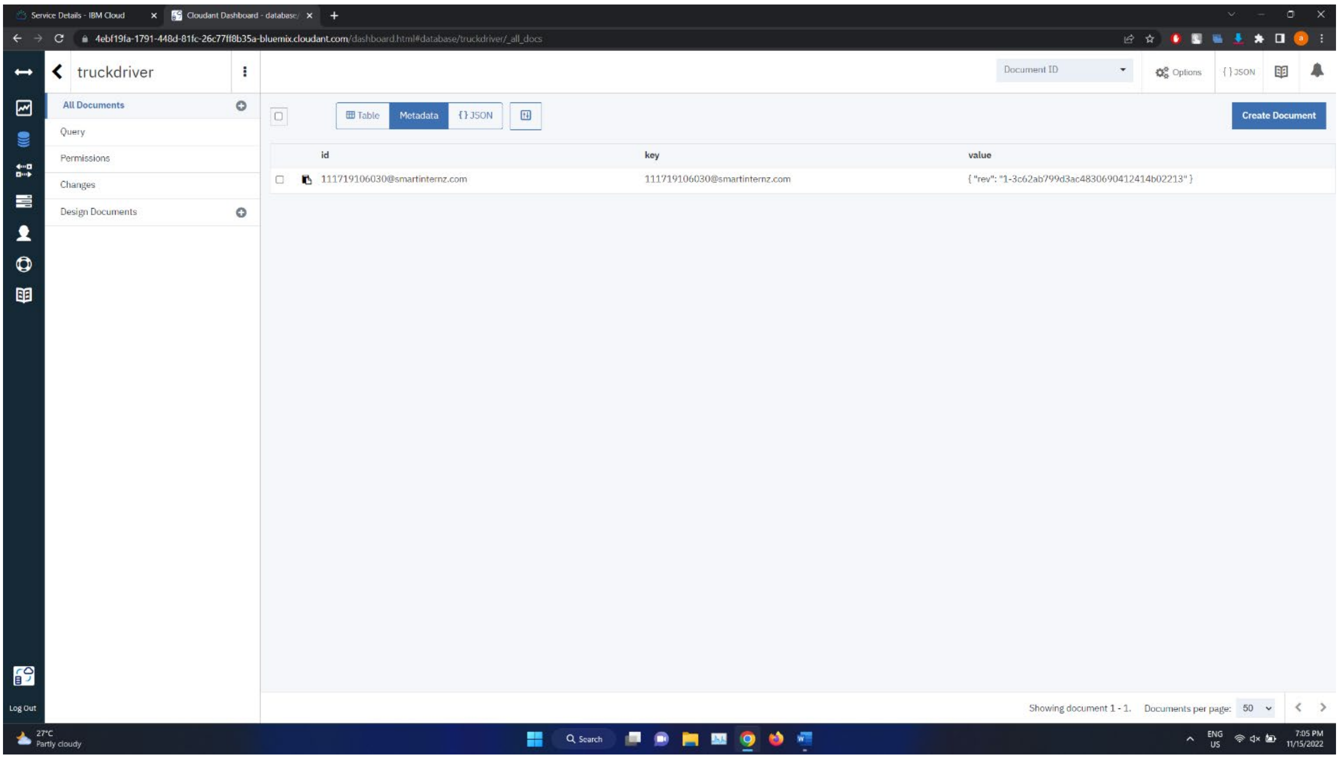
Co-admin Database:



	id	key	value
□	111719106009@smartinternz.com	111719106009@smartinternz.com	{ "rev": "3-7951ba80d9fdbf7333f6e7d5d325fa98" }
□	aswinkumar@gmail.com	aswinkumar@gmail.com	{ "rev": "1-2016876a97af01d9d986c0f1fed28080" }

Showing document 1 - 2. Documents per page: 50

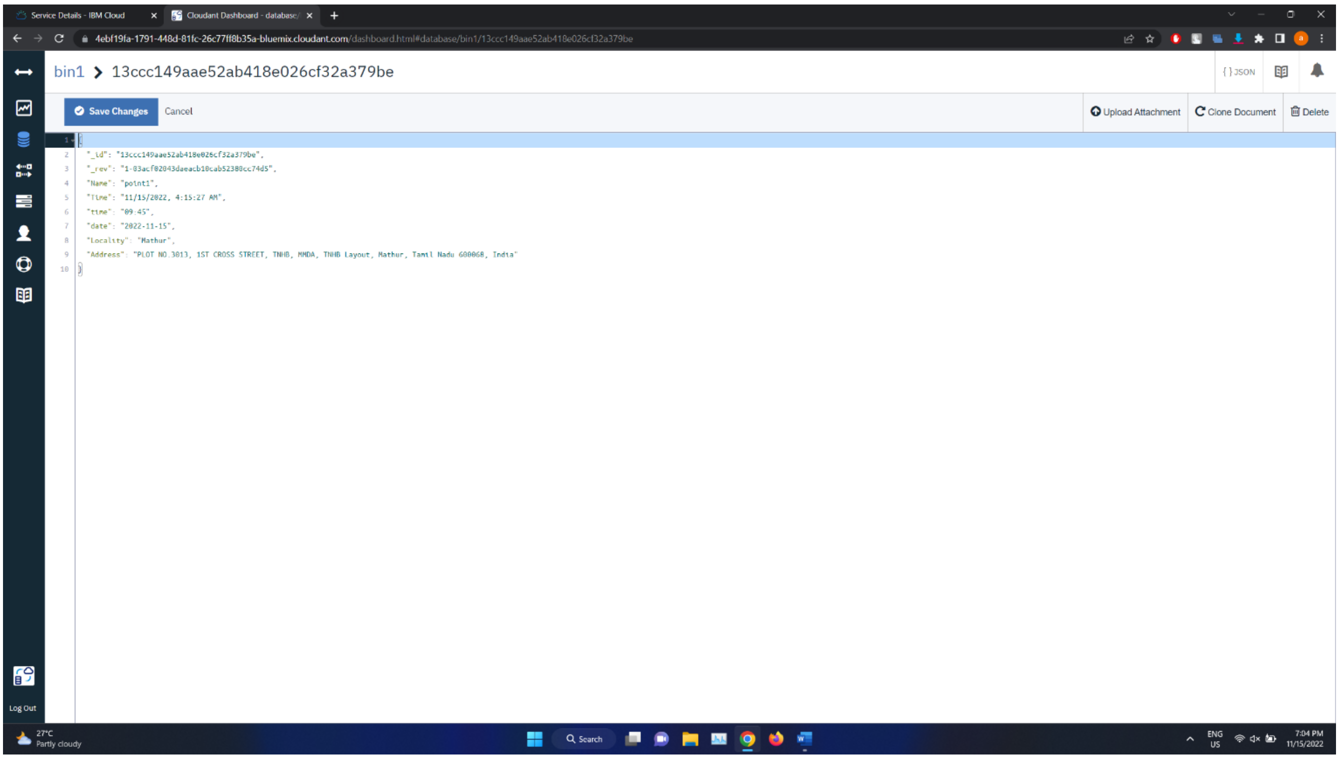
Truck Driver Database:



	id	key	value
□	111719106030@smartinternz.com	111719106030@smartinternz.com	{ "rev": "1-3c62ab799d3ac4830690412414b02213" }

Showing document 1 - 1. Documents per page: 50

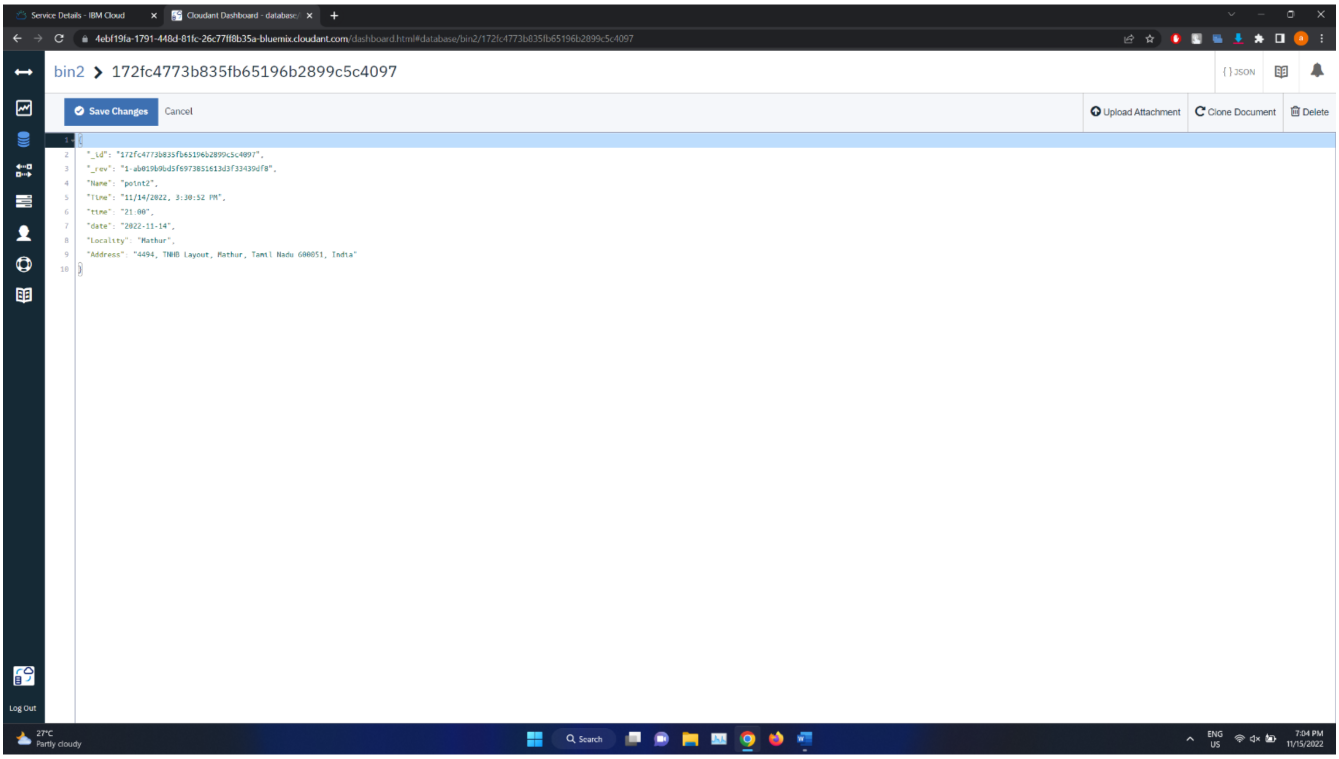
BIN 1 Database



Service Details - IBM Cloud
Cloudant Dashboard - database/
4ebf19fa-1791-448d-81fc-26c77ff8b35a-bluemix.cloudant.com/dashboard.html#database/bin1/13ccc149aae52ab418e026cf32a379be
bin1 > 13ccc149aae52ab418e026cf32a379be
{ } JSON
Save Changes
Cancel
Upload Attachment
Clone Document
Delete
Log Out
27°C
Partly cloudy
Search
ENG
US
7:04 PM
11/15/2022

```
{
  "_id": "13ccc149aae52ab418e026cf32a379be",
  "_rev": "1-03acf02043daeacb10cab52380cc74d5",
  "Name": "point1",
  "Time": "11/15/2022, 4:15:27 AM",
  "time": "09:45",
  "date": "2022-11-15",
  "Locality": "Mathur",
  "Address": "PLOT NO.3013, 1ST CROSS STREET, TNHB, MMDA, TNHB Layout, Mathur, Tamil Nadu 600068, India"
}
```

BIN 2 Database

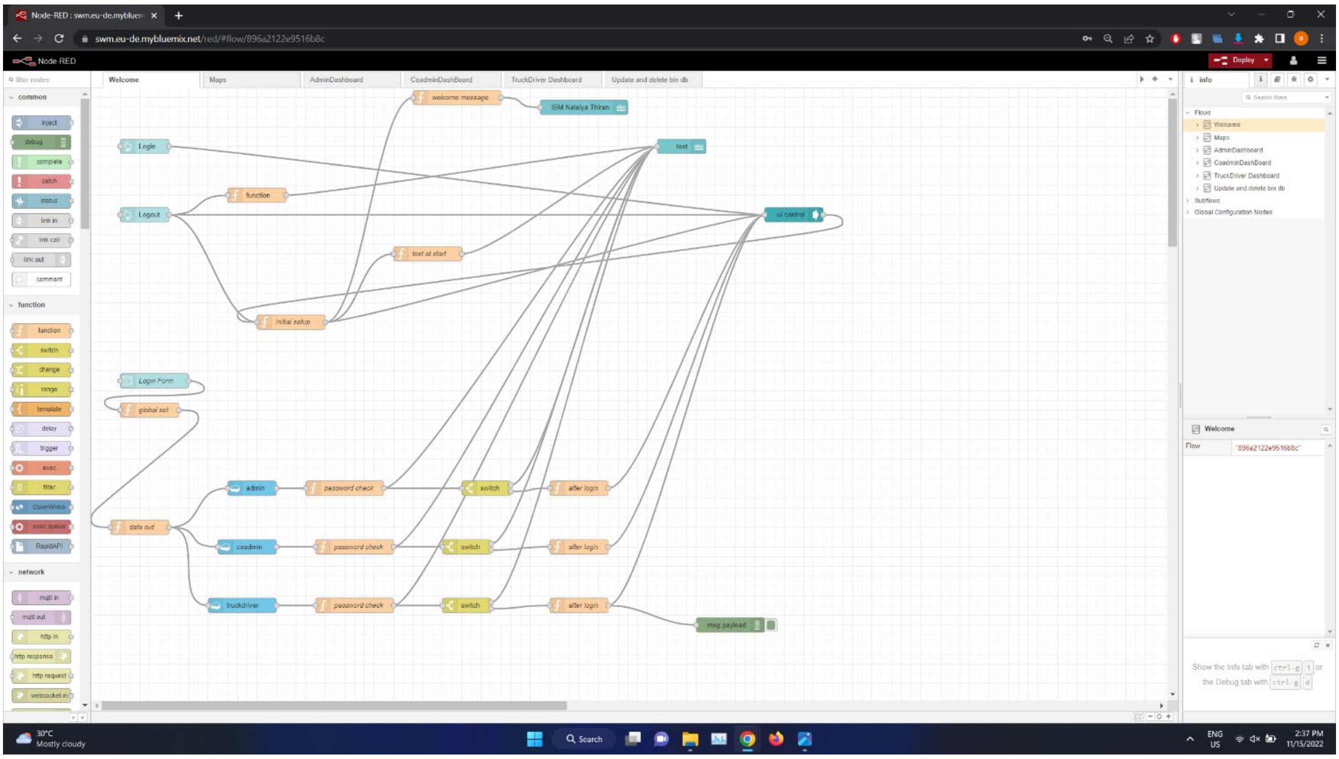


Service Details - IBM Cloud
Cloudant Dashboard - database/
4ebf19fa-1791-448d-81fc-26c77ff8b35a-bluemix.cloudant.com/dashboard.html#database/bin2/172fc4773b835fb65196b2899c5c4097
bin2 > 172fc4773b835fb65196b2899c5c4097
{ } JSON
Save Changes
Cancel
Upload Attachment
Clone Document
Delete
Log Out
27°C
Partly cloudy
Search
ENG
US
7:04 PM
11/15/2022

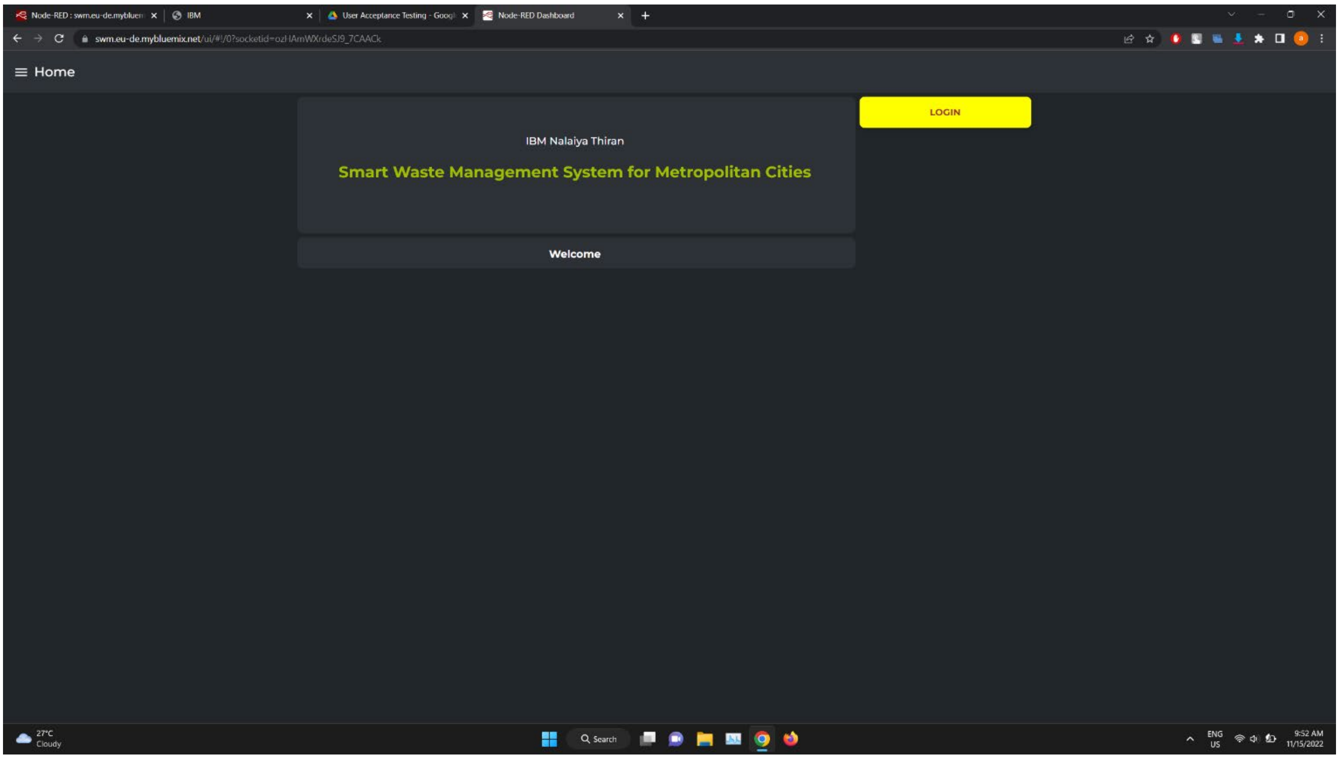
```
{
  "_id": "172fc4773b835fb65196b2899c5c4097",
  "_rev": "1-ab019b9bd5f6973851613d3f33439df8",
  "Name": "point2",
  "Time": "11/14/2022, 3:30:52 PM",
  "time": "21:00",
  "date": "2022-11-14",
  "Locality": "Mathur",
  "Address": "4494, TNHB Layout, Mathur, Tamil Nadu 600051, India"
}
```

5) Node RED flow

Login/Logout (Home Page) Flow:



Home Page:


Node-RED : swm.eu-de.mybluemix.net
IBM
User Acceptance Testing - Googl
Node-RED Dashboard
swm.eu-de.mybluemix.net/ui/#!/0?socketid=ozHAmWXrdeSJ9_7CAACk

Home

LOGIN

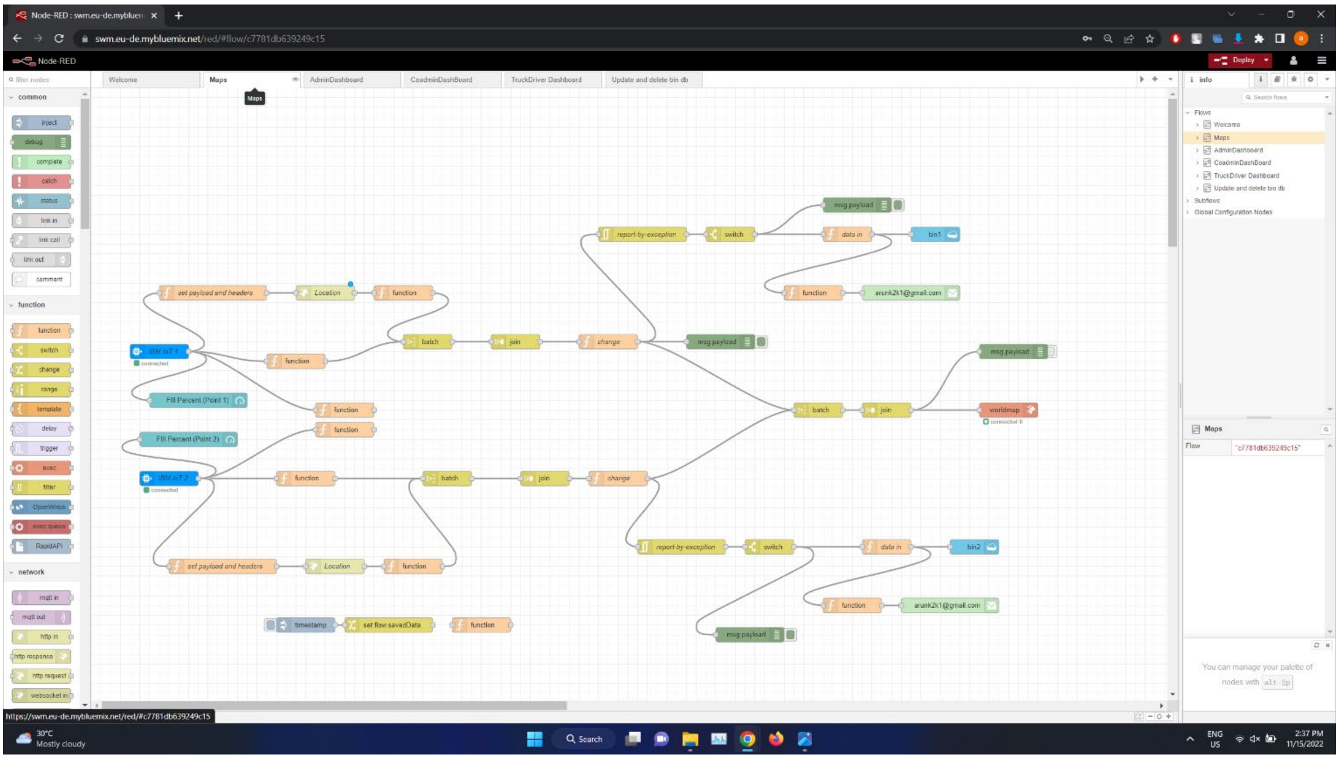
IBM Nalaiya Thiran

Smart Waste Management System for Metropolitan Cities

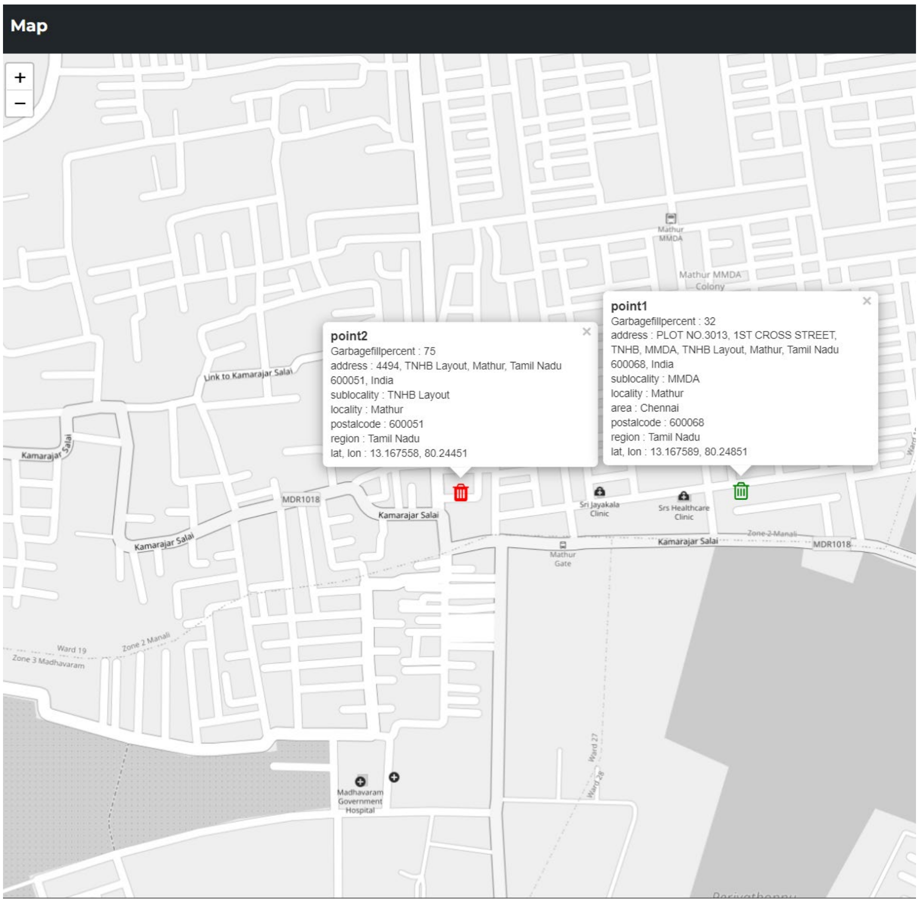
Welcome

Map Flow:

It is used to push bin values got from the IBM Watson IOT platform and push them into Database



Map Output page:



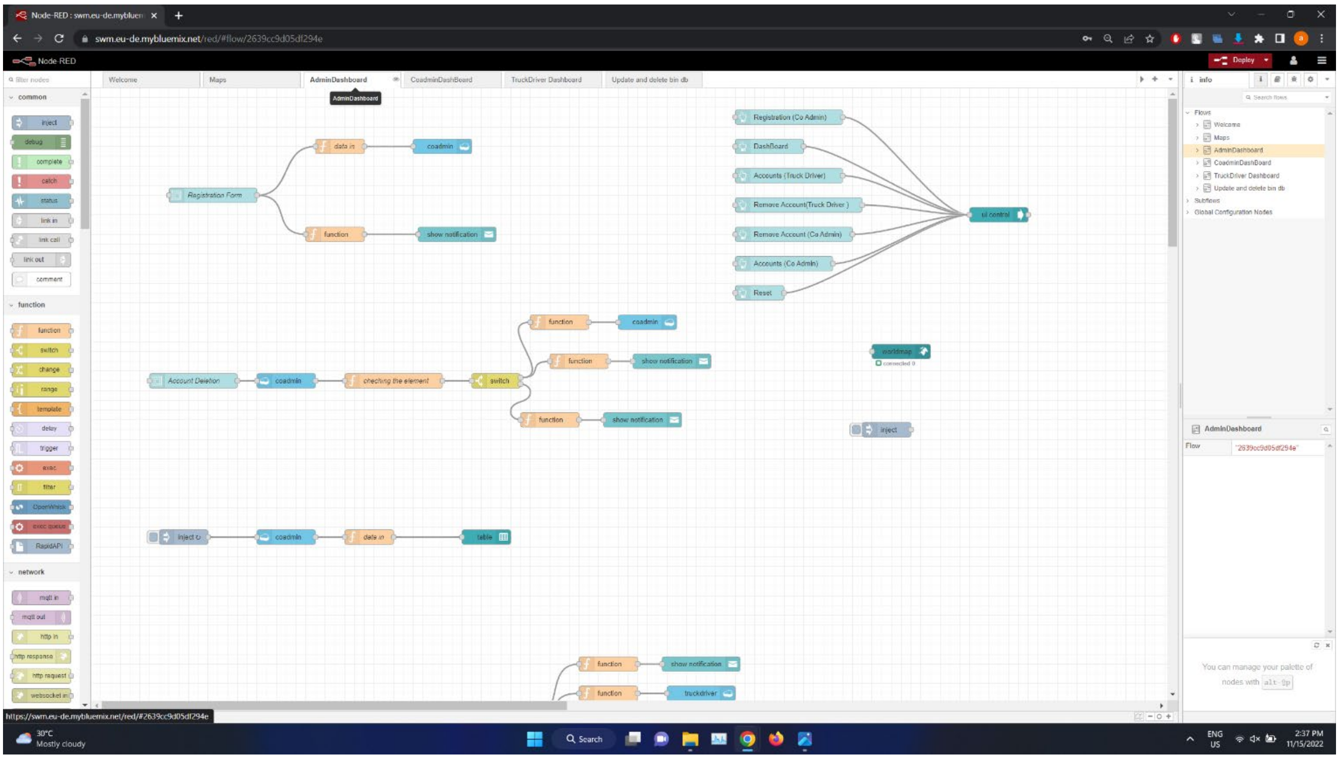
Map

point2
Garbagefillpercent : 75
address : 4494, TNHB Layout, Mathur, Tamil Nadu 600051, India
sublocality : TNHB Layout
locality : Mathur
postalcode : 600051
region : Tamil Nadu
lat, lon : 13.167558, 80.24451

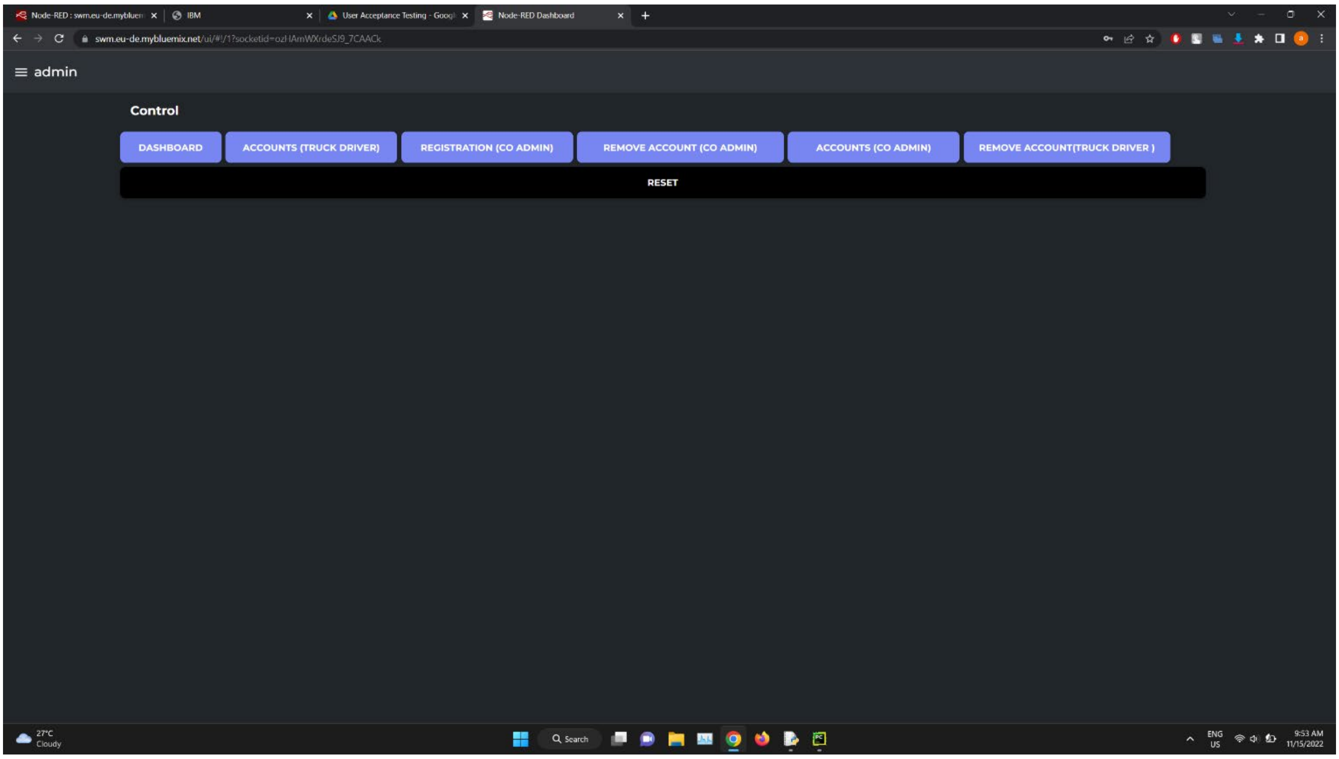
point1
Garbagefillpercent : 32
address : PLOT NO.3013, 1ST CROSS STREET, TNHB, MMDA, TNHB Layout, Mathur, Tamil Nadu 600068, India
sublocality : MMDA
locality : Mathur
area : Chennai
postalcode : 600068
region : Tamil Nadu
lat, lon : 13.167589, 80.24851

Admin Flow:

This flow helps to create all the buttons and UI for Admin page

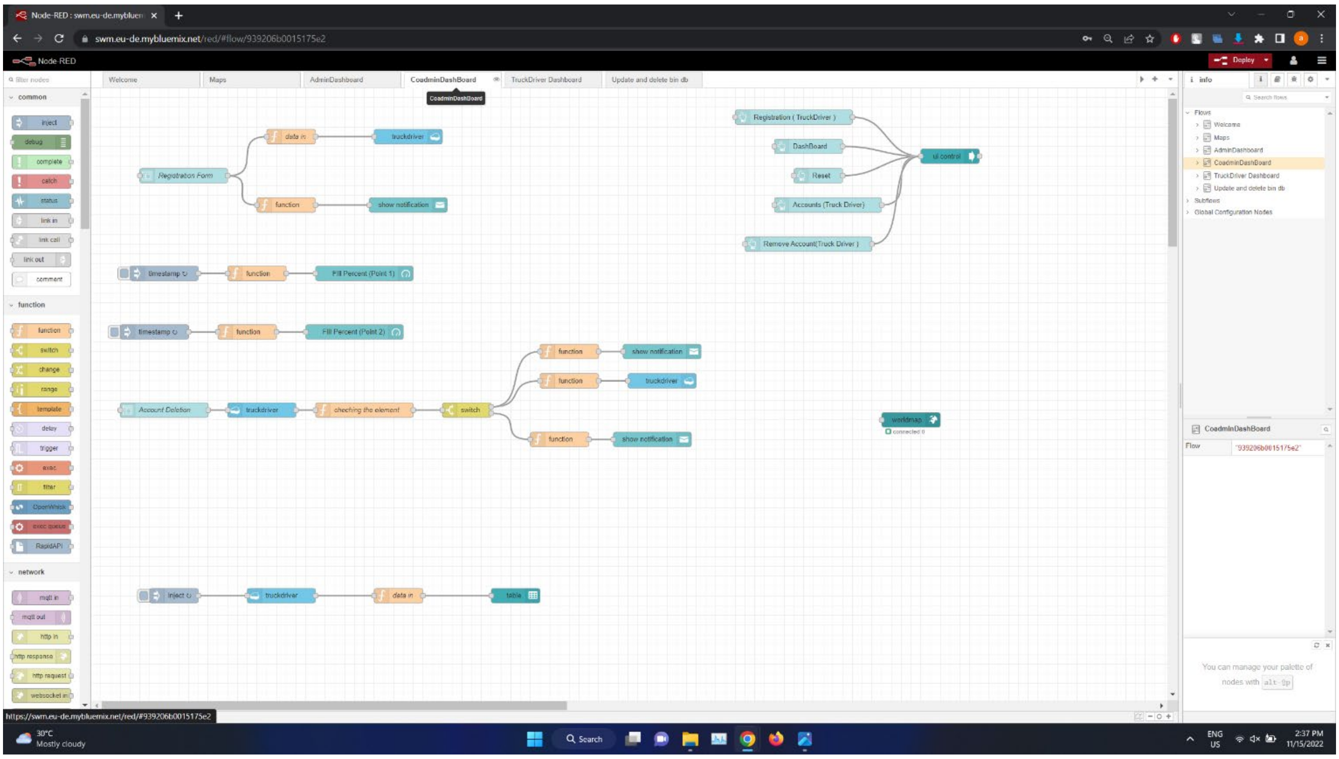


Admin Page:

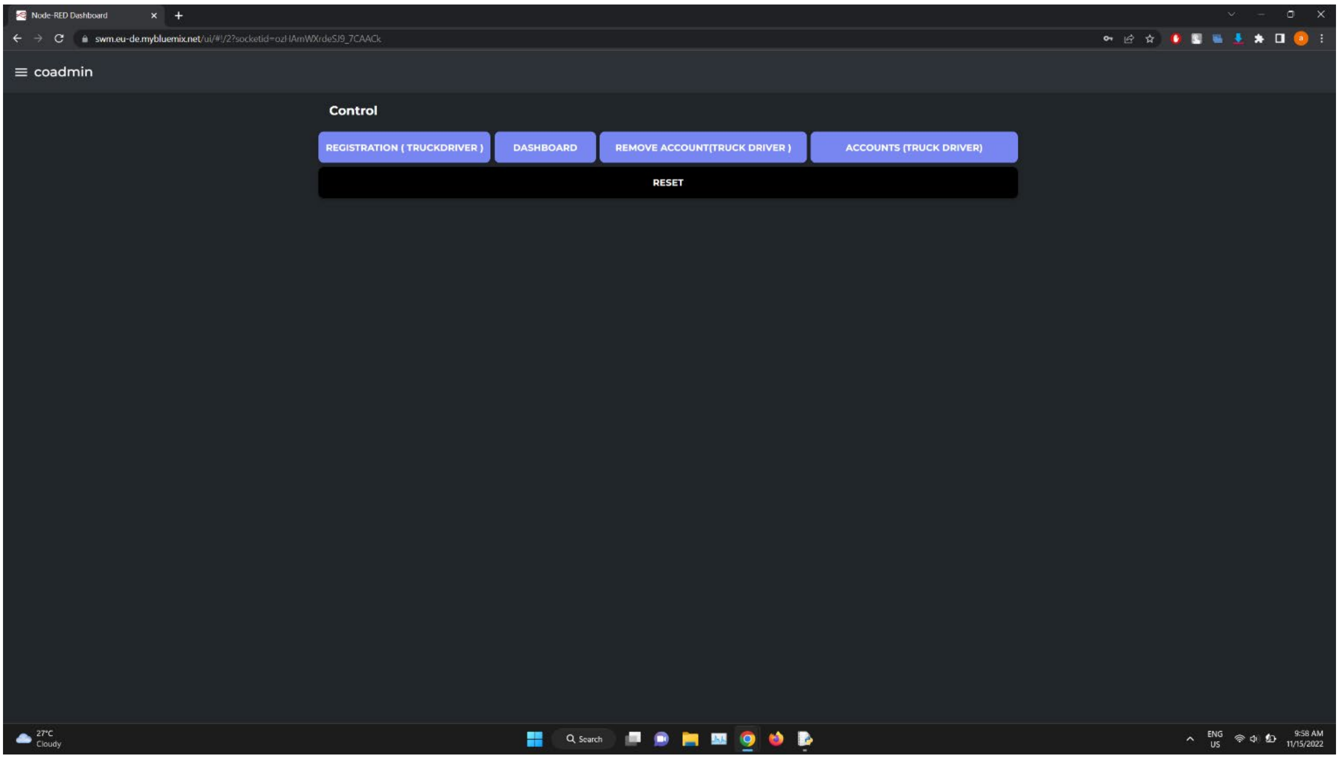

Node-RED : swm.eu-de.mybluem
IBM
User Acceptance Testing - Googl
Node-RED Dashboard
swm.eu-de.mybluemix.net/ui/#!/1?socketid=ozHAmWXrdeSJ9_7CAACk
admin
Control
DASHBOARD
ACCOUNTS (TRUCK DRIVER)
REGISTRATION (CO ADMIN)
REMOVE ACCOUNT (CO ADMIN)
ACCOUNTS (CO ADMIN)
REMOVE ACCOUNT(TRUCK DRIVER)
RESET
27°C
Cloudy
Search
ENG
US
9:53 AM
11/15/2022

Co-Admin Flow:

Here we can see all the UI based functions used for Co-admin page.

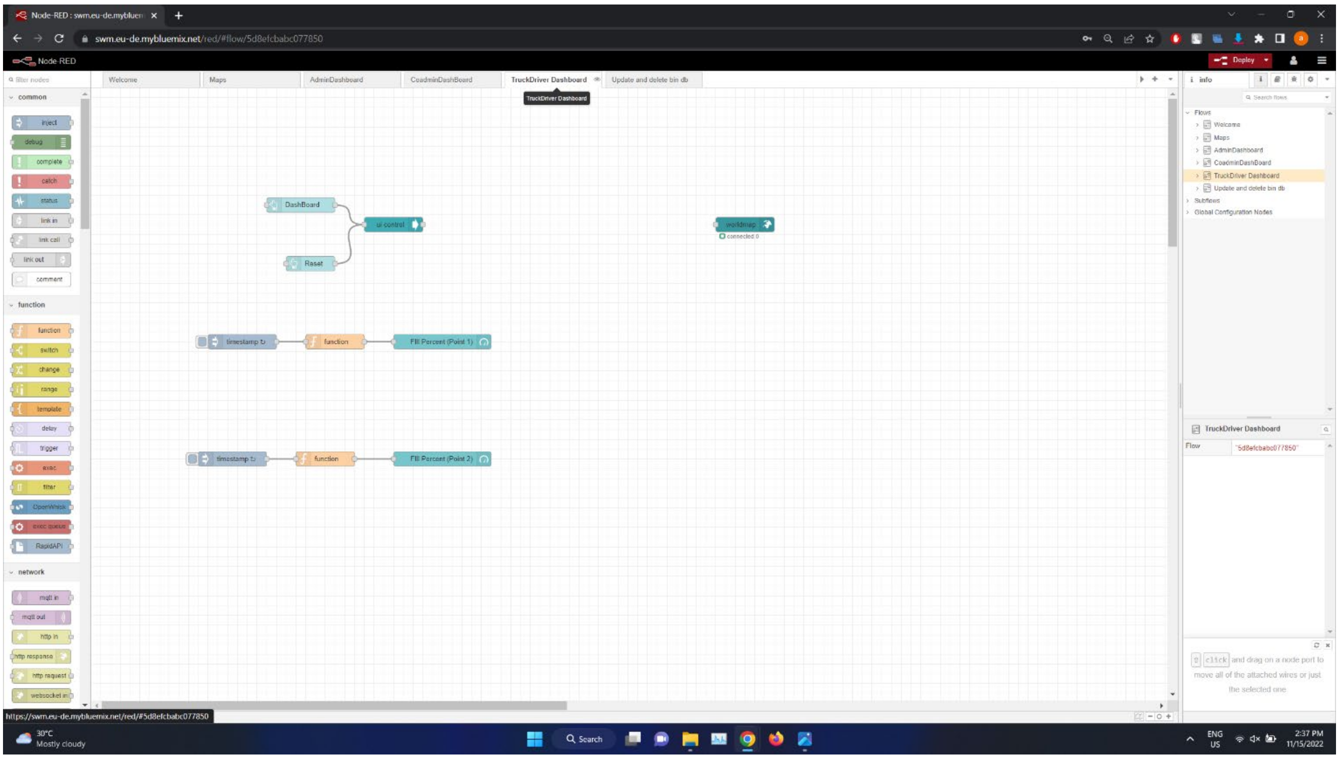


Co-Admin Output page:

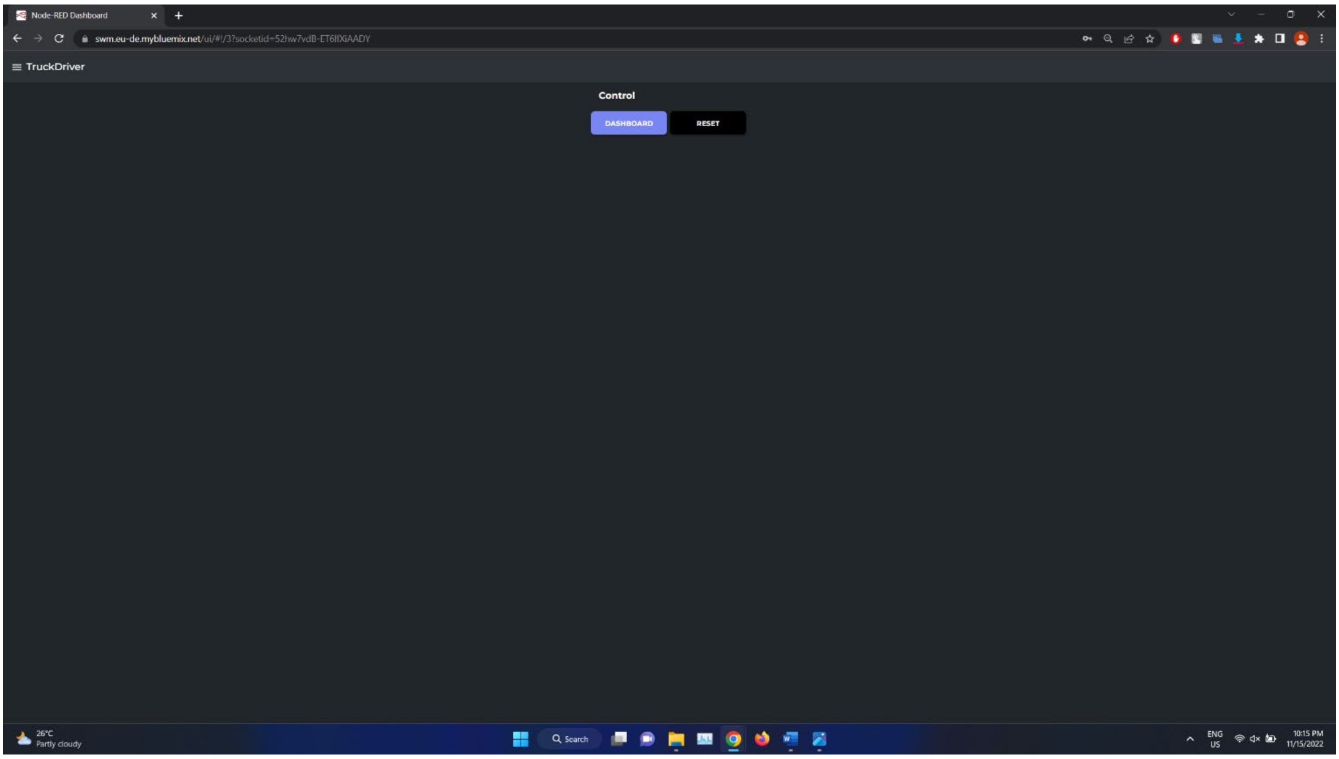

Node-RED Dashboard
swm.eu-de.mybluemix.net/ui/#!/2?socketid=ozHAmWXrdeSJ9_7CAACk
coadmin
Control
REGISTRATION (TRUCKDRIVER)
DASHBOARD
REMOVE ACCOUNT(TRUCK DRIVER)
ACCOUNTS (TRUCK DRIVER)
RESET
27°C
Cloudy
Search
ENG
US
9:58 AM
11/15/2022

Truck Driver Flow:

Here we can see all the UI based functions used for Truck Driver page.

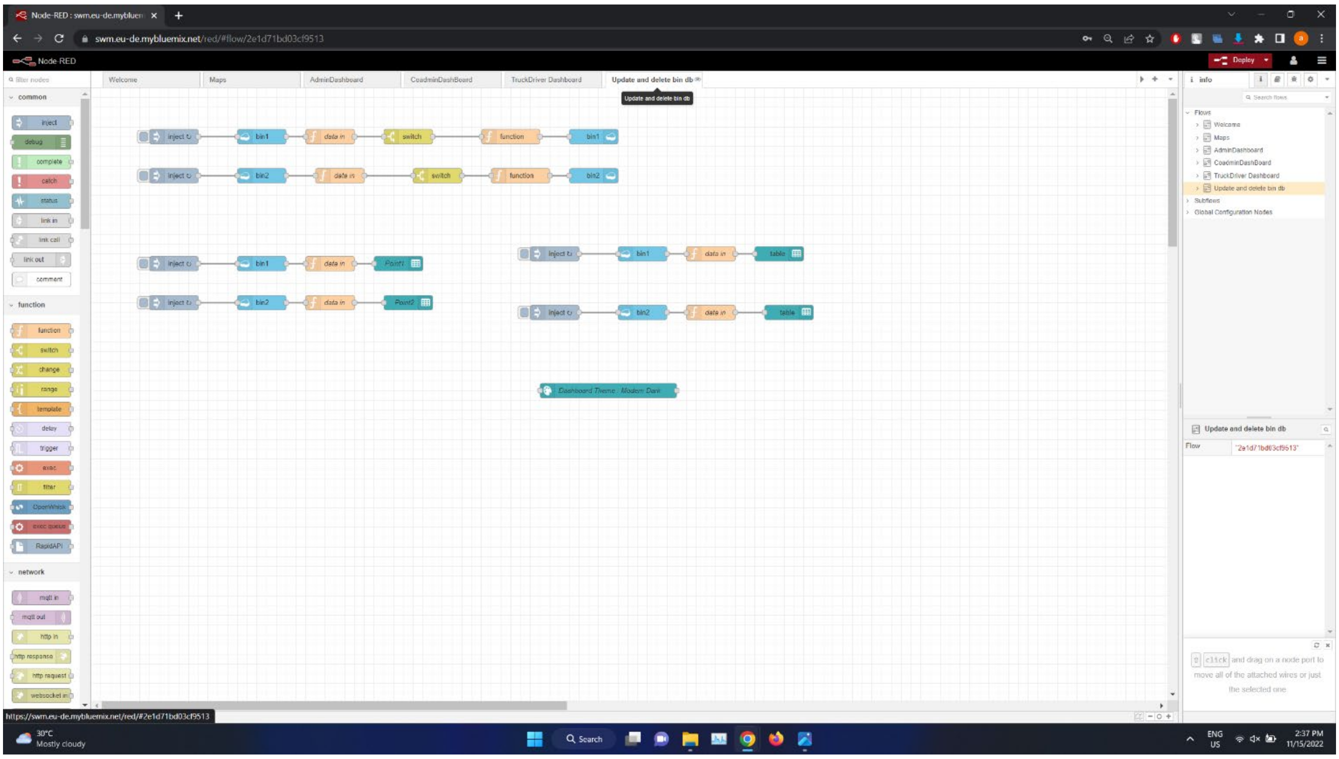


Truck Driver Output page:

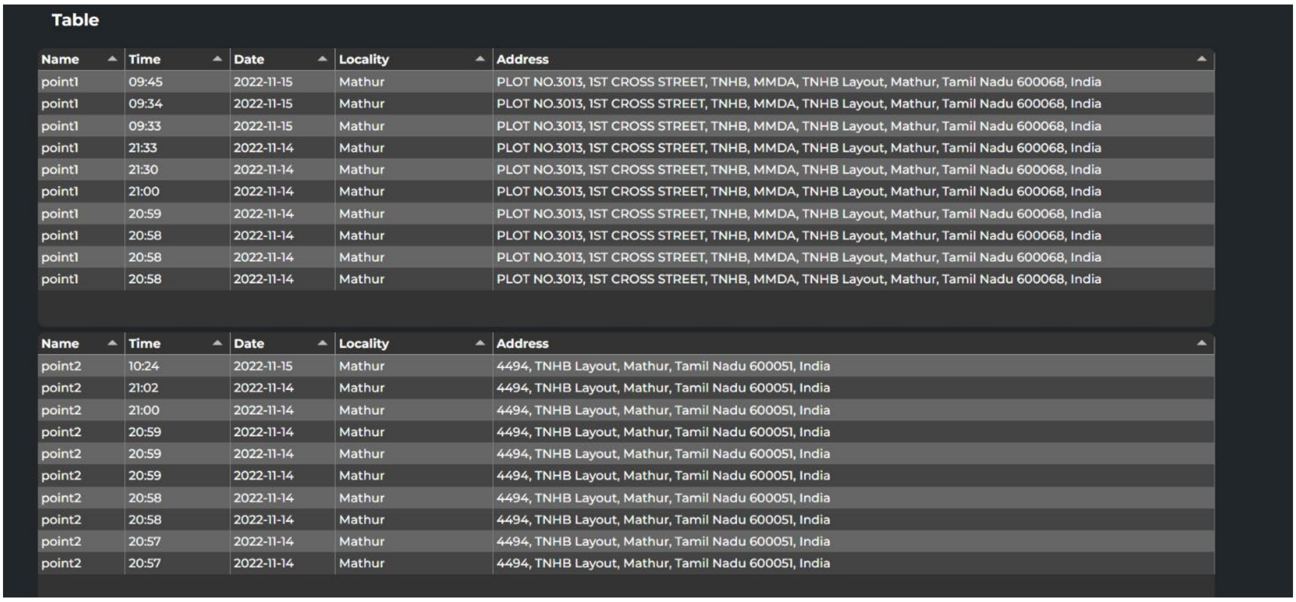


Bin table Flow:

This flow used retrieve the data from the Database and push them into the Table.



Output:



Table

Name	Time	Date	Locality	Address
point1	09:45	2022-11-15	Mathur	PLOT NO.3013, 1ST CROSS STREET, TNHB, MMDA, TNHB Layout, Mathur, Tamil Nadu 600068, India
point1	09:34	2022-11-15	Mathur	PLOT NO.3013, 1ST CROSS STREET, TNHB, MMDA, TNHB Layout, Mathur, Tamil Nadu 600068, India
point1	09:33	2022-11-15	Mathur	PLOT NO.3013, 1ST CROSS STREET, TNHB, MMDA, TNHB Layout, Mathur, Tamil Nadu 600068, India
point1	21:33	2022-11-14	Mathur	PLOT NO.3013, 1ST CROSS STREET, TNHB, MMDA, TNHB Layout, Mathur, Tamil Nadu 600068, India
point1	21:30	2022-11-14	Mathur	PLOT NO.3013, 1ST CROSS STREET, TNHB, MMDA, TNHB Layout, Mathur, Tamil Nadu 600068, India
point1	21:00	2022-11-14	Mathur	PLOT NO.3013, 1ST CROSS STREET, TNHB, MMDA, TNHB Layout, Mathur, Tamil Nadu 600068, India
point1	20:59	2022-11-14	Mathur	PLOT NO.3013, 1ST CROSS STREET, TNHB, MMDA, TNHB Layout, Mathur, Tamil Nadu 600068, India
point1	20:58	2022-11-14	Mathur	PLOT NO.3013, 1ST CROSS STREET, TNHB, MMDA, TNHB Layout, Mathur, Tamil Nadu 600068, India
point1	20:58	2022-11-14	Mathur	PLOT NO.3013, 1ST CROSS STREET, TNHB, MMDA, TNHB Layout, Mathur, Tamil Nadu 600068, India
point1	20:58	2022-11-14	Mathur	PLOT NO.3013, 1ST CROSS STREET, TNHB, MMDA, TNHB Layout, Mathur, Tamil Nadu 600068, India

Name	Time	Date	Locality	Address
point2	10:24	2022-11-15	Mathur	4494, TNHB Layout, Mathur, Tamil Nadu 600051, India
point2	21:02	2022-11-14	Mathur	4494, TNHB Layout, Mathur, Tamil Nadu 600051, India
point2	21:00	2022-11-14	Mathur	4494, TNHB Layout, Mathur, Tamil Nadu 600051, India
point2	20:59	2022-11-14	Mathur	4494, TNHB Layout, Mathur, Tamil Nadu 600051, India
point2	20:59	2022-11-14	Mathur	4494, TNHB Layout, Mathur, Tamil Nadu 600051, India
point2	20:59	2022-11-14	Mathur	4494, TNHB Layout, Mathur, Tamil Nadu 600051, India
point2	20:58	2022-11-14	Mathur	4494, TNHB Layout, Mathur, Tamil Nadu 600051, India
point2	20:58	2022-11-14	Mathur	4494, TNHB Layout, Mathur, Tamil Nadu 600051, India
point2	20:57	2022-11-14	Mathur	4494, TNHB Layout, Mathur, Tamil Nadu 600051, India
point2	20:57	2022-11-14	Mathur	4494, TNHB Layout, Mathur, Tamil Nadu 600051, India