

## PROJECT DEVELOPMENT PHASE

### SPRINT 1 DELIVERY

TEAM ID	PNT2022TMID37162
Project Name	Smart Waste Management System for Metropolitan cities
Sprint Number	Sprint 1

#### Python Coding:-

```
PythonCoding.py - C:\Users\ECS FINANCIAL\Desktop\PythonCoding.py (3.7.0)
File Edit Format Run Options Window Help

import time
import sys
import ibmiotf.application
import ibmiotf.device
import random

organization = "mfy97c"
deviceType = "device3"
deviceId = "1357"
authMethod = "token"
authToken = "12345678"

def myCommandCallback(cmd):
    print("Command recieved : %s"% cmd.data['command'])
    status = cmd.data['command']
    if status == "lighton":
        print("LED is on")
    else:
        print("LED is Off")

try:
    deviceOptions = {"org": organization, "type": deviceType,"id":deviceId,"auth-method":authMethod,"auth-token":authToken}
    deviceCli = ibmiotf.device.Client(deviceOptions)

except Exception as e:
    print("Caught exception connecting device: %s"%str(e))
    sys.exit()

deviceCli.connect()
while True:
    for i in range(10):
        Bin1Level= random.randint(0,100)
        Bin2Level = random.randint(0,100)
        Bin3Level = random.randint(0,100)

        data1={'Bin1Level':Bin1Level , 'Bin2Level':Bin2Level, 'Bin3Level':Bin3Level}

        if Bin1Level > 90:
            warn1 = 'alert bin full'
        elif Bin1Level > 60:
            warn1 = 'Time to clean'
        else:
            warn1 = 'May be cleaned later'

        if Bin2Level > 90:
            warn2 = 'alert bin full'
        elif Bin2Level > 60:
            warn2 = 'Time to clean'
        else:
            warn2 = 'May be cleaned later'

        if Bin3Level > 90:
            warn3 = 'alert bin full'
        elif Bin3Level > 60:
            warn3 = 'Time to clean'
        else:
            warn3 = 'May be cleaned later'

        def myOnPublishCallback():
            print("\nPublished Bin1Level = %s "%Bin1Level,
                  "Bin2Level = %s "%Bin2Level ,
                  "Bin3Level = %s "%Bin3Level,"to IBM Watson")
            print("Bin1-%s"%warn1)
            print("Bin2-%s"%warn2)
            print("Bin3-%s"%warn3)
        success = deviceCli.publishEvent("IoTSensor", "json", data1, qos=0, on_publish=myOnPublishCallback)
        if not success:
            print("Not conncted to IOTF")
        time.sleep(10)
        deviceCli.commandCallback=myCommandCallback
    deviceCli.disconnect()
```

```
PythonCoding.py - C:\Users\ECS FINANCIAL\Desktop\PythonCoding.py (3.7.0)
File Edit Format Run Options Window Help

Bin2Level = random.randint(0,100)
Bin3Level = random.randint(0,100)

data1={'Bin1Level':Bin1Level , 'Bin2Level':Bin2Level, 'Bin3Level':Bin3Level}

if Bin1Level > 90:
    warn1 = 'alert bin full'
elif Bin1Level > 60:
    warn1 = 'Time to clean'
else:
    warn1 = 'May be cleaned later'

if Bin2Level > 90:
    warn2 = 'alert bin full'
elif Bin2Level > 60:
    warn2 = 'Time to clean'
else:
    warn2 = 'May be cleaned later'

if Bin3Level > 90:
    warn3 = 'alert bin full'
elif Bin3Level > 60:
    warn3 = 'Time to clean'
else:
    warn3 = 'May be cleaned later'

def myOnPublishCallback():
    print("\nPublished Bin1Level = %s "%Bin1Level,
          "Bin2Level = %s "%Bin2Level ,
          "Bin3Level = %s "%Bin3Level,"to IBM Watson")
    print("Bin1-%s"%warn1)
    print("Bin2-%s"%warn2)
    print("Bin3-%s"%warn3)
success = deviceCli.publishEvent("IoTSensor", "json", data1, qos=0, on_publish=myOnPublishCallback)
if not success:
    print("Not conncted to IOTF")
time.sleep(10)
deviceCli.commandCallback=myCommandCallback
deviceCli.disconnect()
```

## PYHTON OUTPUT;-

```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\ECS FINANCIAL\Desktop\PythonCoding.py =====
2022-11-18 14:17:19,654 ibmiotf.device.Client INFO Connected successfully: dmfy87c:device3:1357

Published BinLevel = 9 Bin2Level = 1 Bin3Level = 59 to IBM Watson
Bin1-May be cleaned later
Bin2-May be cleaned later
Bin3-May be cleaned later

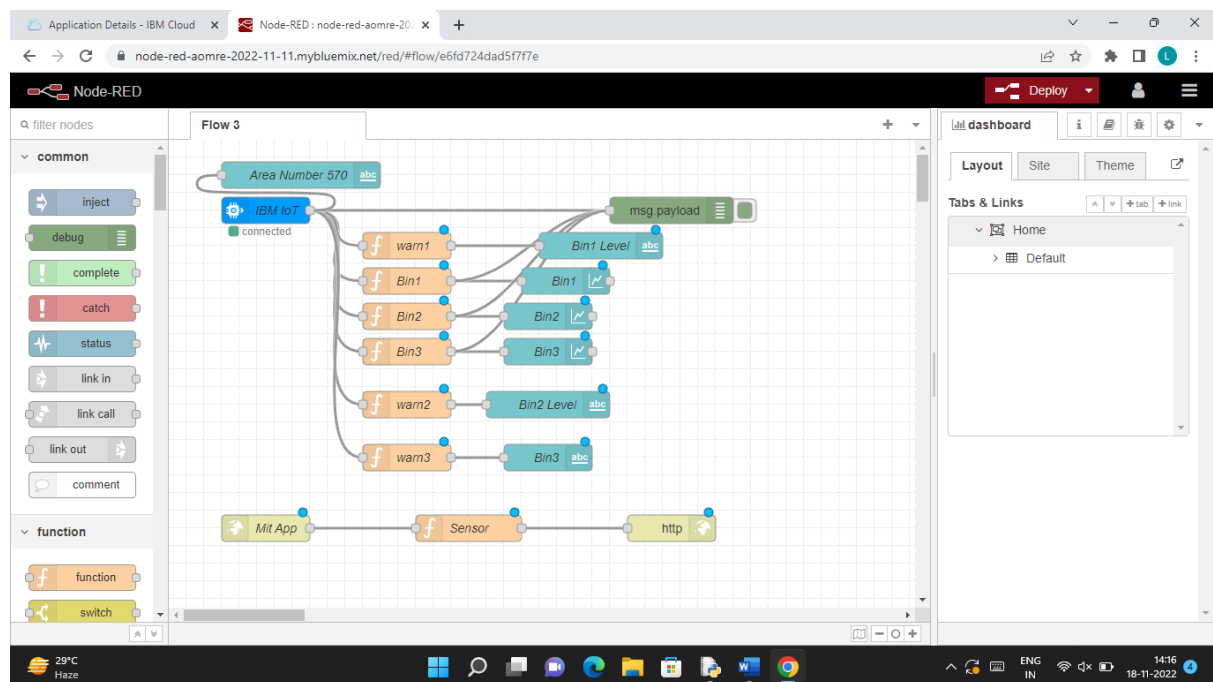
Published BinLevel = 42 Bin2Level = 95 Bin3Level = 58 to IBM Watson
Bin1-May be cleaned later
Bin2-alert bin full
Bin3-May be cleaned later

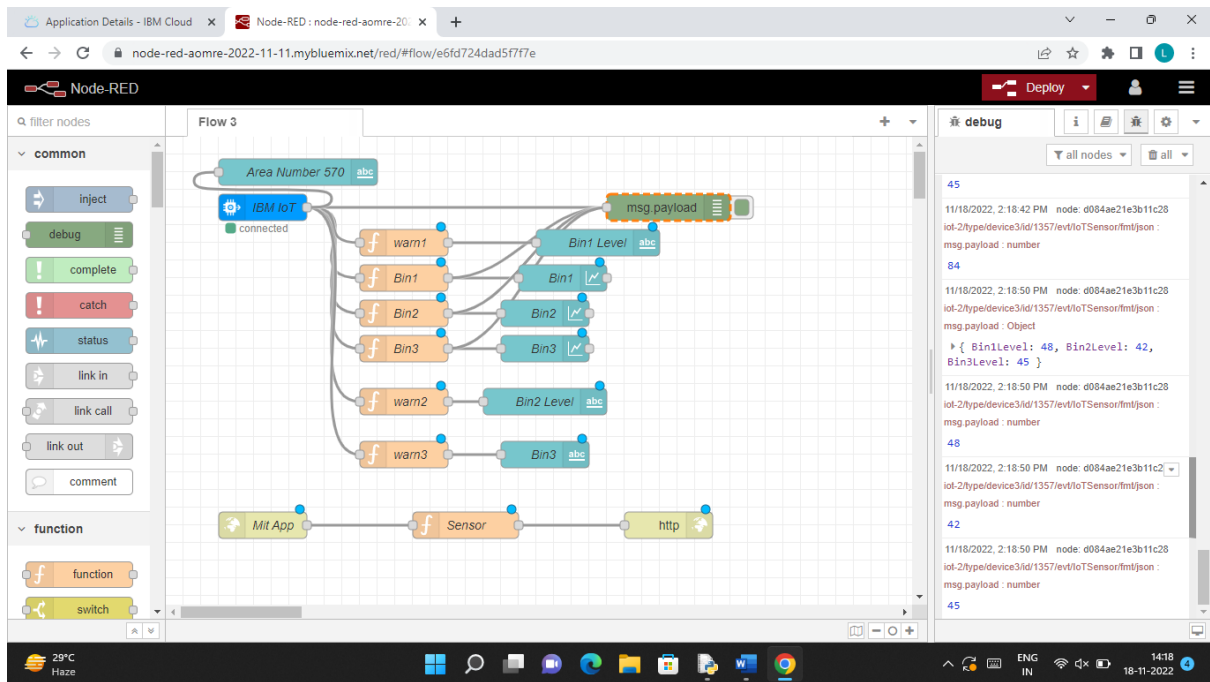
Published BinLevel = 9 Bin2Level = 38 Bin3Level = 44 to IBM Watson
Bin1-May be cleaned later
Bin2-May be cleaned later
Bin3-May be cleaned later

Published BinLevel = 19 Bin2Level = 87 Bin3Level = 7 to IBM Watson
Bin1-May be cleaned later
Bin2-Time to clean
Bin3-May be cleaned later

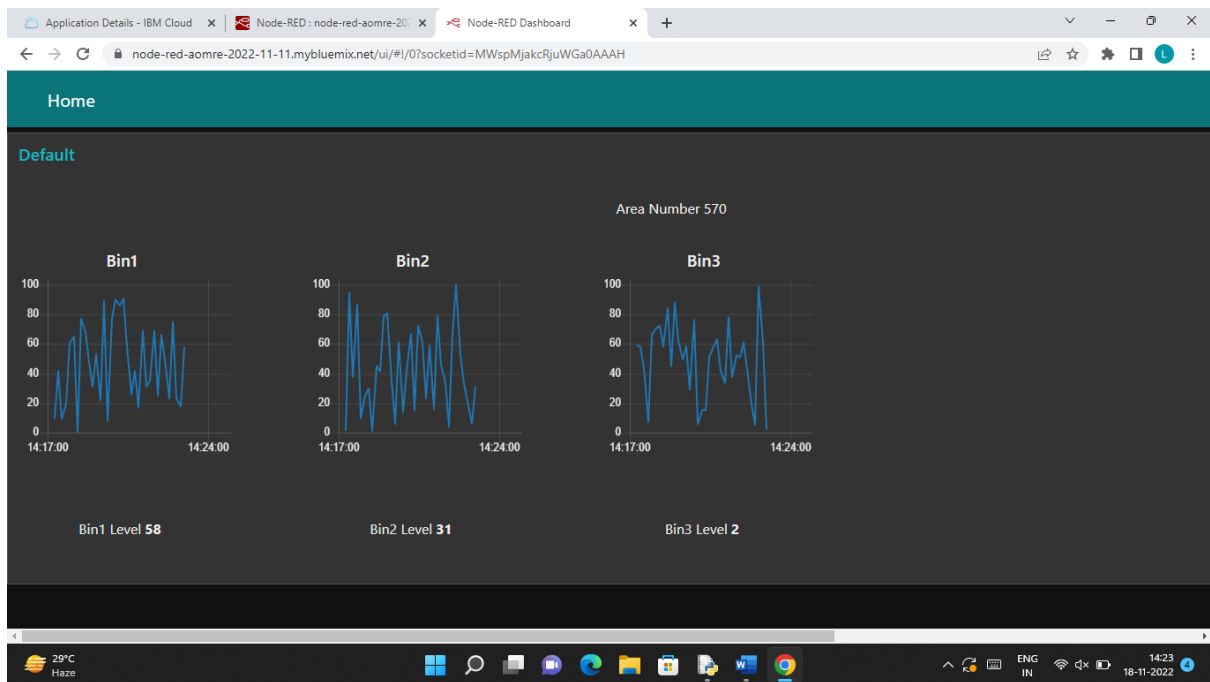
Published BinLevel = 61 Bin2Level = 10 Bin3Level = 66 to IBM Watson
Bin1-Time to clean
Bin2-May be cleaned later
Bin3-Time to clean
|
```

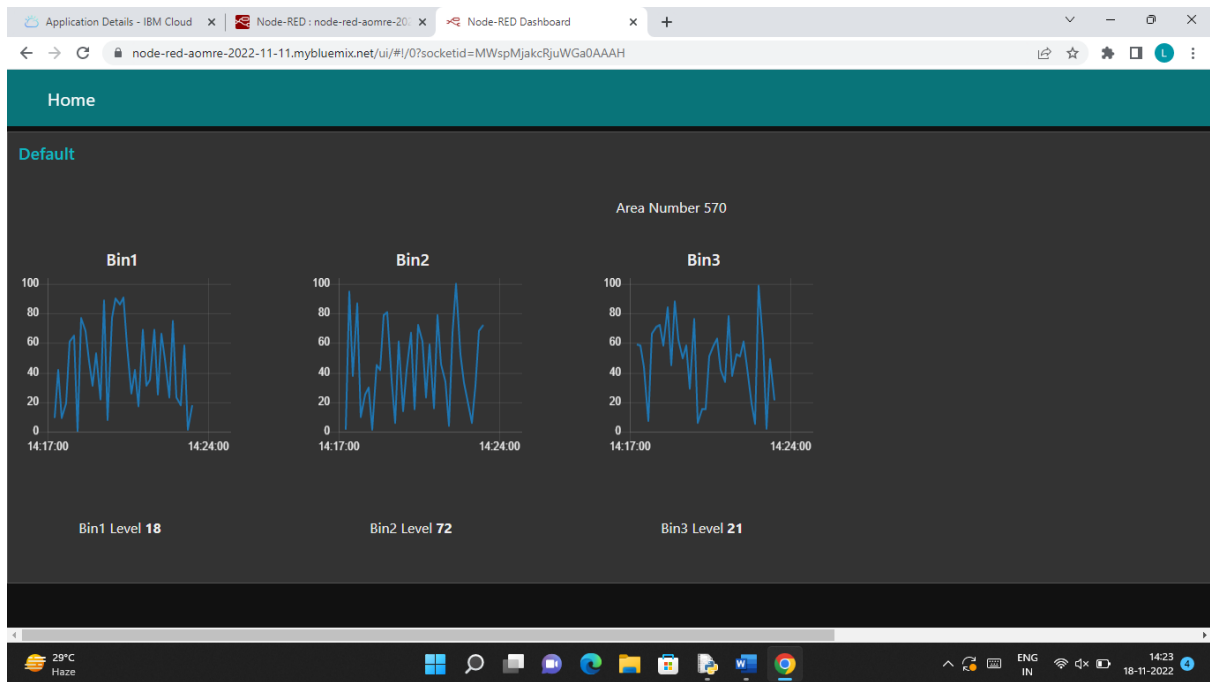
## NODE-RED CONNECTIONS;-





## NODE RED DASHBOARD:-





#### ADMIN LOGIN PAGE:-

The screenshot shows a web browser window displaying an admin login page. The browser's address bar shows the URL: `localhost/Project/`. The page has a green background. In the center, there is a dark brown rectangular box containing the following text: "TEAM ID : PNT2022TMID37162", "SMART WASTE MANAGEMENT SYSTEM FOR METROPOLITAN CITIES", and "ADMIN LOGIN". Below this text, there are two input fields: "User Name" and "Password". A blue "Login" button is located at the bottom right of the dark brown box. The Windows taskbar at the bottom shows the time as 14:40 on 18-11-2022, with a temperature of 29°C and a "Haze" weather condition.

#### LOGIN CODE:-

This screenshot shows the Visual Studio Code editor with the `index.php` file open. The Explorer sidebar on the left shows the project structure with files like `launch.json`, `db_conn.php`, `home.php`, `index.php`, `login.php`, `logout.php`, and `style.css`. The main editor area displays the HTML code for `index.php`, which includes a login form and a header section. The status bar at the bottom indicates the file is at line 7, column 13, with 4 spaces, UTF-8 encoding, and CRLF line endings. The bottom status bar also shows the system temperature as 29°C and the time as 14:40 on 18-11-2022.

```
1 <!DOCTYPE html>
2
3 <html>
4
5 <head>
6
7   <title> | LOGIN</title>
8
9   <link rel="stylesheet" type="text/css" href="style.css">
10
11 </head>
12
13 <body>
14
15   <form action="login.php" method="post">
16     <h2> TEAM ID : PNT2022TMD37162 </h2>
17     <h2> SMART WASTE MANAGEMENT SYSTEM FOR METROPOLITAN CITIES </h2>
18     <h2>ADMIN LOGIN</h2>
19
20
21   <?php if (isset($_POST['uname']) & & isset($_POST['password'])) { >
```

This screenshot shows the Visual Studio Code editor with the `login.php` file open. The Explorer sidebar on the left shows the project structure with files like `launch.json`, `db_conn.php`, `home.php`, `index.php`, `login.php`, `logout.php`, and `style.css`. The main editor area displays the PHP code for `login.php`, which includes session management, database connection, and a validation function. The status bar at the bottom indicates the file is at line 70, column 1, with 4 spaces, UTF-8 encoding, and CRLF line endings. The bottom status bar also shows the system temperature as 29°C and the time as 14:41 on 18-11-2022.

```
1 <?php
2
3 session_start();
4
5 include "db_conn.php";
6
7 if (isset($_POST['uname']) & & isset($_POST['password'])) {
8
9   function validate($data){
10
11     $data = trim($data);
12
13     $data = stripslashes($data);
14
15     $data = htmlspecialchars($data);
16
17     return $data;
18
19   }
20
21   $uname = validate($_POST['uname']);
```