

Project Development Phase

Sprint – III

Date	14 November 2022
Team id	PNT2022TMID51225
Project name	Signs with smart connectivity for better road safety

Sprint targets:

Sprint	Functional requirements	USN	User story/Task	Story points	Priority	Team members
Sprint-3	Login	USN-5	As an admin, I should have an account	7	Low	Rithiga Sneka Monika
Sprint-3	Dashboard	USN-6	As an admin, I should be able to see and observe the sign nodes	13	Medium	Angelin Sankareswari

Wowki Simulation:

The screenshot displays the Wowki simulation environment. On the left, a code editor shows the following sketch:

```
1 #include <WiFi.h> //library for wifi
2 #include <PubSubClient.h> //library for MQTT
3 #include "DHT.h" // Library for dht11
4 #define DHTPIN 5 // what pin we're connected to
5 #define DHTTYPE DHT22 // define type of sensor DHT 11
6
7 DHT dht (DHTPIN, DHTTYPE); // creating the instance by passing pin and type of
8
9 void callback(char* topic, byte* payload, unsigned int payloadLength)
10
11 //-----credentials of IBM Accounts-----
12
13 #define ORG "twidrq" //IBM ORGANIZATION ID
14 #define DEVICE_TYPE "Sample_one" //Device type mentioned in ibm watson IOT Plat
15 #define DEVICE_ID "4054" //Device ID mentioned in ibm watson IOT Platform
16 #define TOKEN "12345678" //Token
17 String data3;
18 float h, t;
19
20 //----- Customise the above values -----
21
22 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
23 char publishTopic[] = "iot-2/evt/Data/fmt/json"; // topic name and type of even
24 char subscribTopic[] = "iot-2/cmd/command/fmt/String"; // cmd REPRESENT comma
25 char authMethod[] = "use-token-auth"; // authentication method
26 char token[] = TOKEN;
27 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //client id
28
29 //-----
30
```

On the right, the simulation interface shows a visual representation of the ESP32 microcontroller and the DHT22 sensor module. The sensor is connected to the microcontroller via a breadboard. The simulation output shows the following data:

```
temp:37.40, humidity:86.00, North:0, South:0, East:0, West:0
Publish ok
temp:37.40
humidity:86.00
Sending payload:
{"temp":37.40,"humidity":86.00,"North":0,"South":0,"East":0,"West":0}
Publish ok
temp:37.40
```

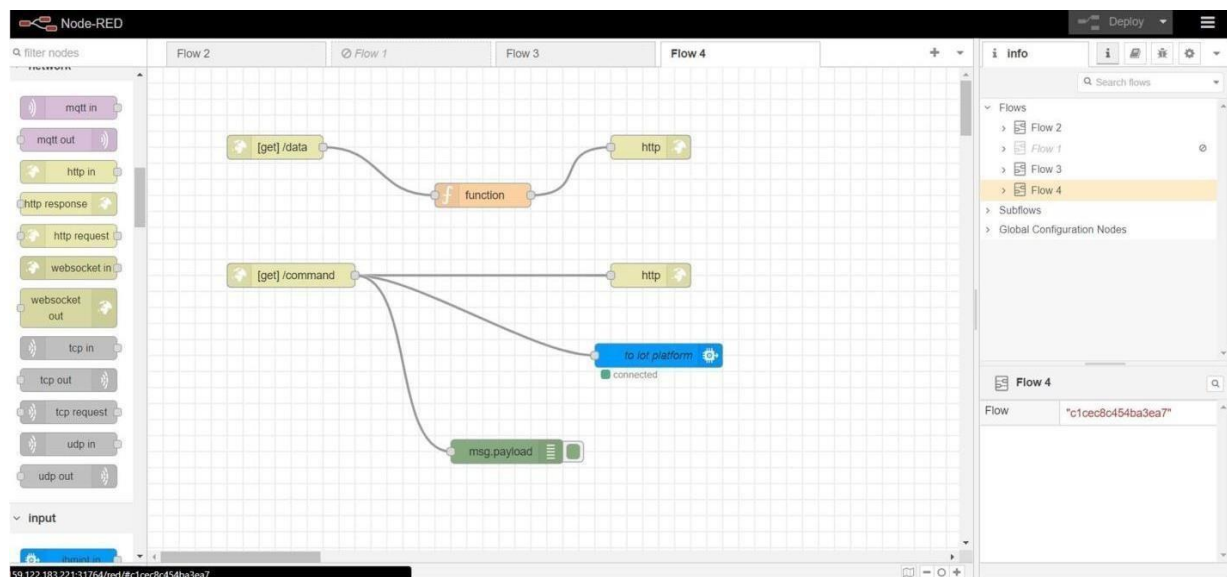
IoT Device in IoT Platform:

The screenshot shows a web interface for an IoT platform. At the top, there's a navigation bar with 'Browse', 'Action', 'Device Types', and 'Interfaces'. A 'Add Device' button is on the right. Below this is a table listing devices. The selected device is '4054', which is 'Disconnected', of type 'Sample_one', with class ID 'Device', and was added on 'Nov 7, 2022 10:15 PM'. Below the table, there's a tabbed interface with 'Identity', 'Device Information', 'Recent Events', 'State', and 'Logs'. The 'Recent Events' tab is active, showing a message: 'The recent events listed show the live stream of data that is coming and going from this device.' Below this is a table of recent events:

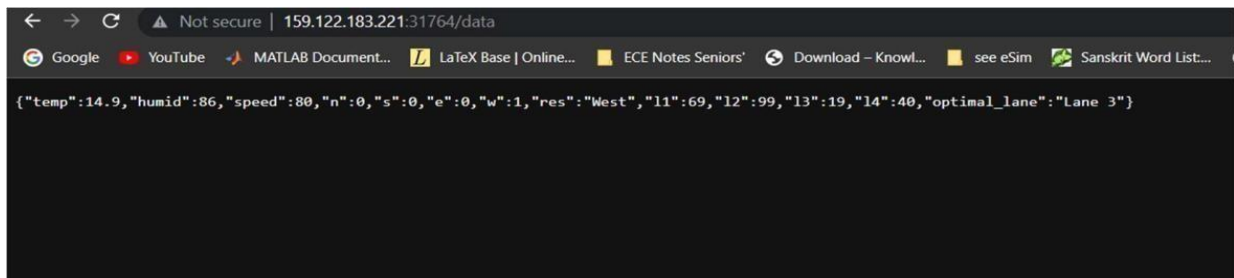
Event	Value	Format	Last Received
event_1	{"temperature":40,"humidity":38}	json	a few seconds ago
event_1	{"temperature":21,"humidity":72}	json	a few seconds ago
event_1	{"temperature":28,"humidity":74}	json	a few seconds ago
event_1	{"temperature":15,"humidity":32}	json	a few seconds ago
event_1	{"temperature":47,"humidity":26}	json	a few seconds ago

At the bottom right, there's a status indicator: '1 Simulation running'.

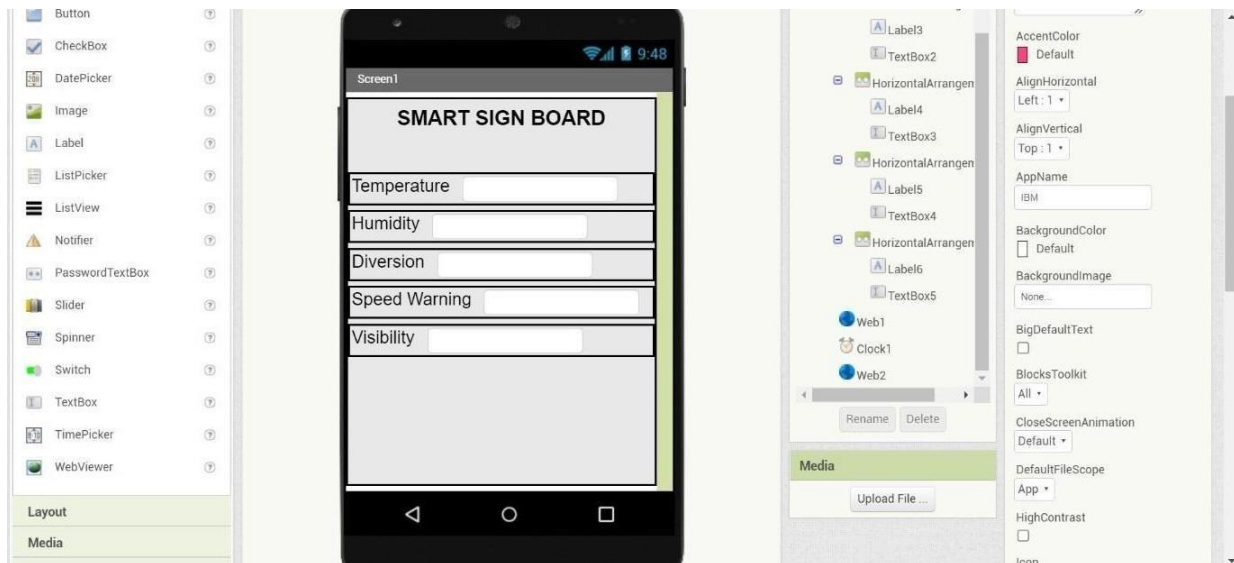
Node red:



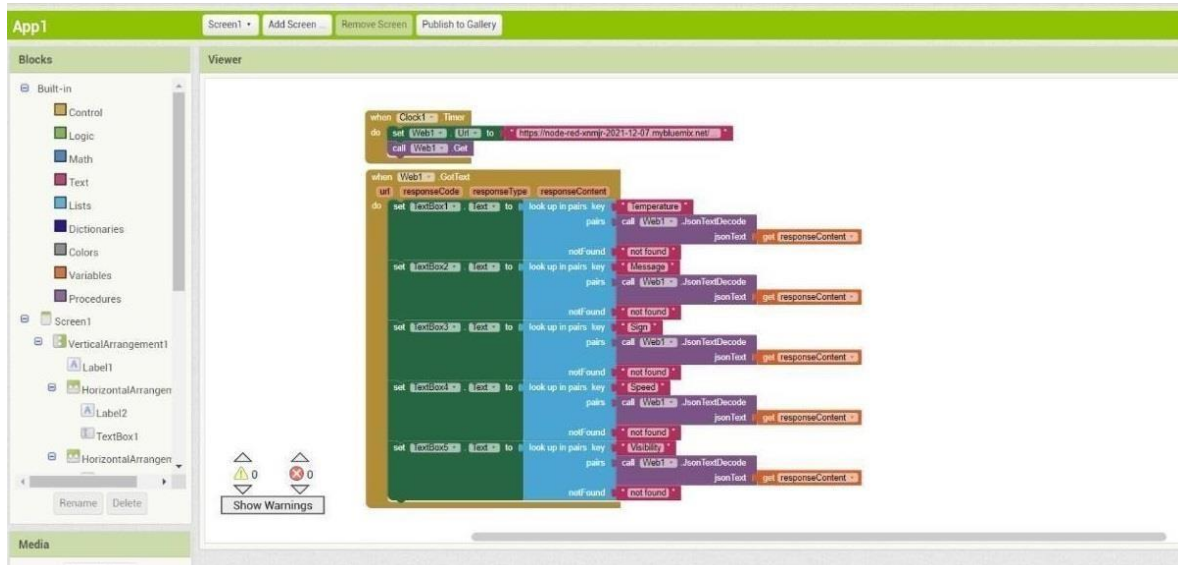
Creating link for connecting with the MIT app Inventor and getting the output:



MIT App Inventor UI design: (Frontend design)



Connecting created Node-RED link with the MIT App inventor :(Backend design)



Display from MIT App: (Output)

