

**Project Development Phase**  
**Sprint-3: MIT App Design and Testing**

Date	7 November 2022
Team id	PNT2022TMID39519
Project Name	Project – Signs with Smart Connectivity for Better Road Safety
Maximum Marks	8 Marks

## Wokwi Simulation

Wokwi Simulation interface showing a project named "final\_iot.ino copy". The interface includes a code editor on the left, a simulation window on the right, and a terminal at the bottom.

**Code Editor (sketch.ino):**

```
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 dht connect
8
9 void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);
10
11 //-----credentials of IBM Accounts-----
12
13 #define ORG "psh4py"//IBM ORGANITION ID
14 #define DEVICE_TYPE "alert-device"//Device type mentioned in ibm watson IOT Platform
15 #define DEVICE_ID "4571"//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 event perform a
24 char subscribetopic[] = "iot-2/cmd/command/fmt/String";// cmd REPRESENT command type AND
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
31 WiFiClient wificlient; // creating the instance for wificlient
32 PubSubClient client(server, 1883, callback, wificlient); //calling the predefined client
33
34
35 void setup()// configuring the ESP32
```

**Simulation Window:** Shows a circuit diagram with an ESP32 microcontroller connected to a DHT22 temperature and humidity sensor. The sensor is connected to a breadboard with an ON switch. The simulation is running, indicated by a green play button and a timer showing 00:49.005 and 93% battery.

**Terminal Output:**

```
{ "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 }
```

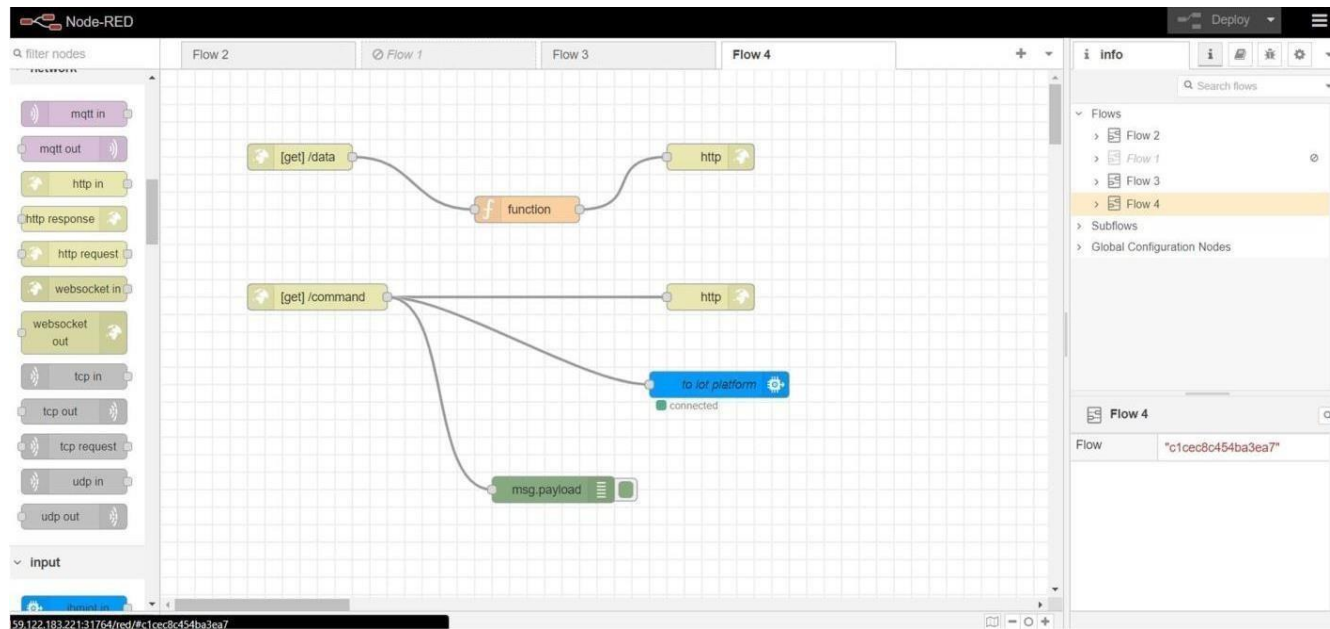
## IoT Device – IoT Platform

The screenshot displays the 'Recent Events' tab for a device named '0001'. The interface includes a top navigation bar with 'Browse', 'Action', 'Device Types', and 'Interfaces'. A sidebar on the left contains various icons. The main content area shows a table of recent events, with a message indicating that the events are a live stream of data. The table has four columns: 'Event', 'Value', 'Format', and 'Last Received'. The events are all of type 'rnd\_number' and are received in 'json' format. A status box at the bottom right indicates '1 Simulation running'.

Event	Value	Format	Last Received
rnd_number	{"Lane_1":5,"Lane_2":83,"Lane_3":30,"Lane_4":...	json	a few seconds ago
rnd_number	{"Lane_1":59,"Lane_2":59,"Lane_3":94,"Lane_4":...	json	a few seconds ago
rnd_number	{"Lane_1":93,"Lane_2":88,"Lane_3":49,"Lane_4":...	json	a few seconds ago
rnd_number	{"Lane_1":2,"Lane_2":61,"Lane_3":21,"Lane_4":...	json	a few seconds ago
rnd_number	{"Lane_1":70,"Lane_2":11,"Lane_3":69,"Lane_4":...	json	a few seconds ago

1 Simulation running

## Node Red – Connect with MIT AppInventor



## Edit function node

Delete

Cancel

☐ Properties



'g• Name

Name



☐ Setup

On Start

On Message

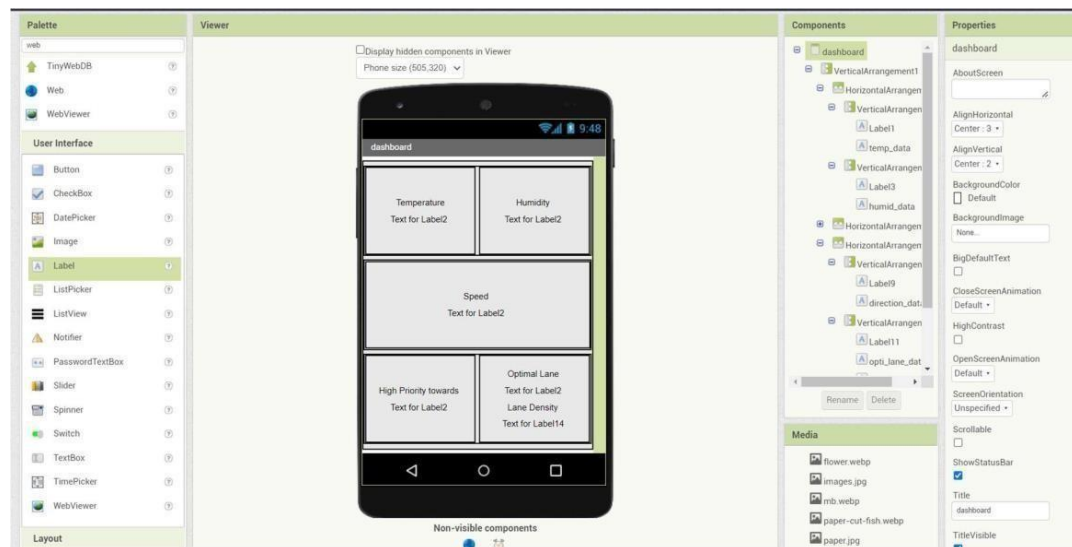
On Stop

```
• msg.payload = {
2   "temp": global.get("temp"),
3   "humid": global.get("humid"),
p  "speed": global.get("speed"),
s   "n": global.get("n"),
6   "s": global.get("s"),
7   "e": global.get("e"),
8   "w": global.get("w"),
g   "res": global.get("res"),
16  "11": global.get("11"),
tt  "12": global.get("12"),
12  "13": global.get("13"),
13  "14": global.get("14"),
t4  "optimal lane": global.get("optimal lane")
15
16' };
17
ig return msg;
```

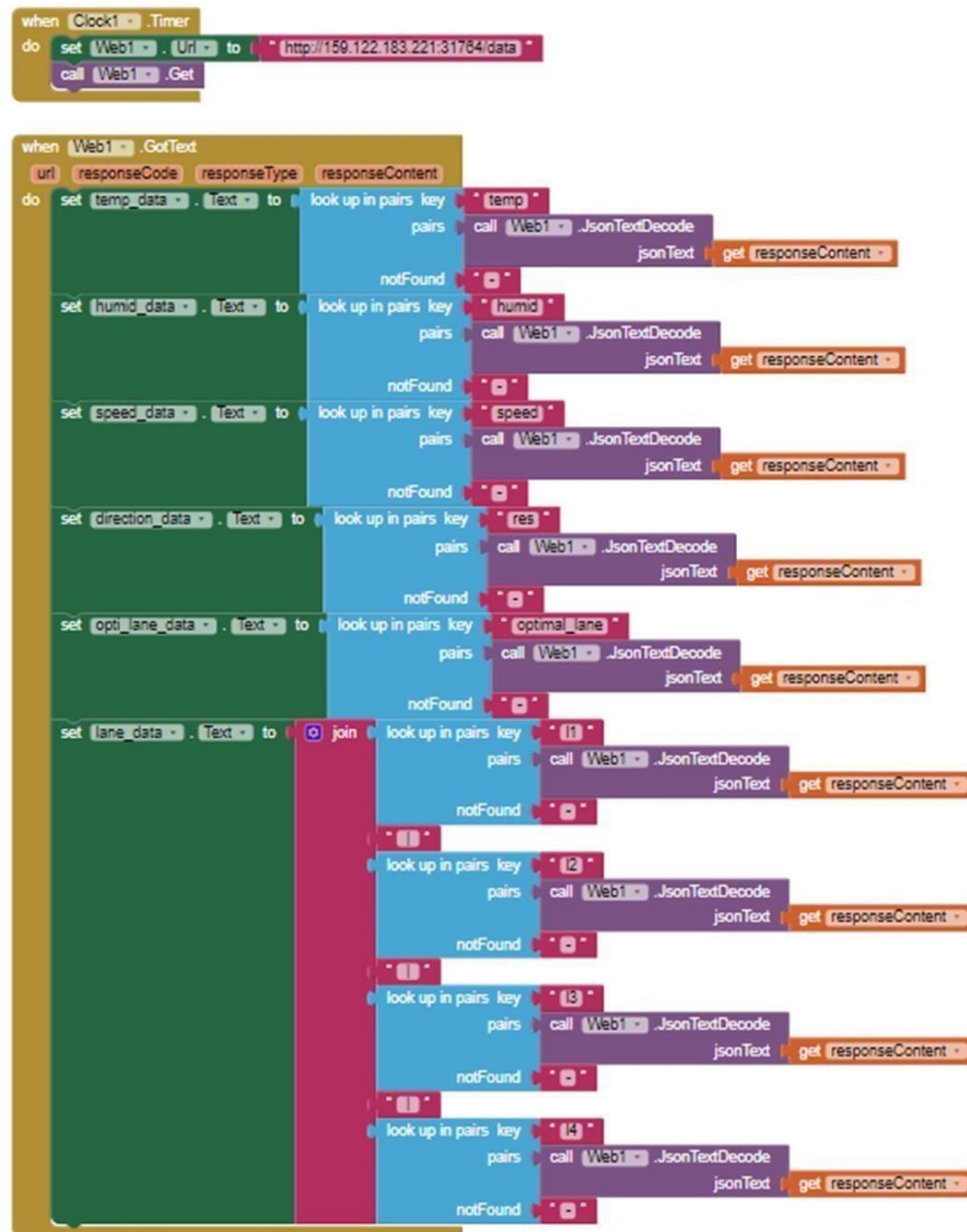
## Output from Node red:

```
← → ↻ Not secure | 159.122.183.221:31764/data
Google YouTube MATLAB Document... LaTeX Base | Online... ECE Notes Seniors' Download - Know... see eSim Sanskrit Word List...
{"temp":14.9,"humid":86,"speed":80,"n":0,"s":0,"e":0,"w":1,"res":"West","11":69,"12":99,"13":19,"14":40,"optimal_lane":"Lane 3"}
```

## MIT App Inventor UI design:



## MIT App Inventor Backend design:



**Sprint 3 delivery:**

**(OUTPUT) Display from MIT App:**

