

PROJECT DEVELOPMENT PHASE

SPRINT 2

TEAM ID	PNT2022TMID44448
PROJECT NAME	Signs With smart connectivity for better road safety

SPRINT TARGET :

Sprint	Functional requirments (Epic)	User Story Number	User Story/Task	Story points	priority
Sprint-2	Safe ride	USN-4	As a traveler, I should have a hustie free journey	20	Medium

Wokwi Simulation: <https://wokwi.com/projects/348562583249224276>

The screenshot displays the Wokwi web-based simulation environment. On the left, the 'sketch.ino' file is open, showing the following code:

```

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* subscribetopic, byte* payload, unsigned int payloadLength)
10
11 //-----credentials of IBM Accounts-----
12
13 #define ORG "w5704q" //IBM ORGANITION ID
14 #define DEVICE_TYPE "PNTIBM" //Device type mentioned in ibm watson IOT Platform
15 #define DEVICE_ID "PNTIBM" //Device ID mentioned in ibm watson IOT Platform
16 #define TOKEN "Wzi6IvG7x2rEYl7pc8" //Token
17 String data3;
18 float h, t;
19
20
21 //----- Customise the above values -----
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 subscribetopic[] = "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

```

On the right, the 'Simulation' window shows a visual representation of the ESP32 microcontroller connected to a DHT22 sensor. The sensor's output is displayed as follows:

```

{"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

```

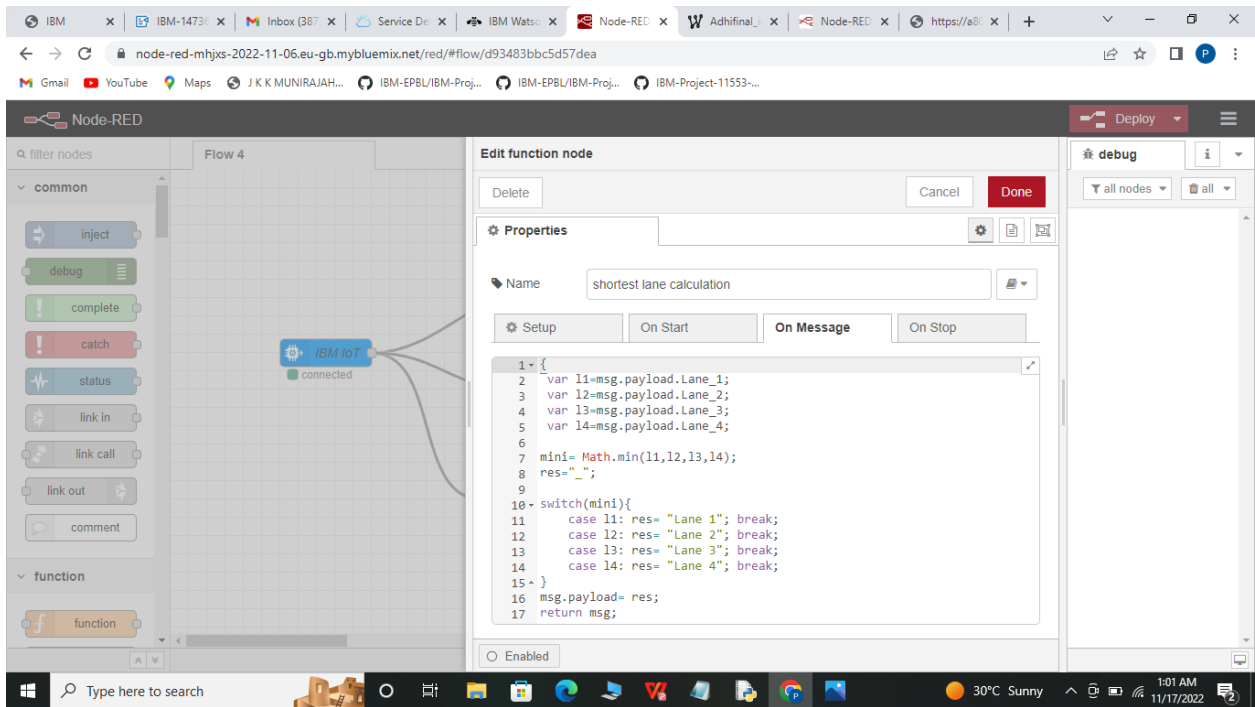
IoT Device – IoT Platform:

The screenshot shows the IBM Watson IoT Platform dashboard. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A table lists devices, with 'PNTIBM' selected. Below the table, the 'Recent Events' tab is active, displaying a list of events. The events table has columns for Event, Value, Format, and Last Received.

Event	Value	Format	Last Received
Data	{"temp":37.4,"humidity":86,"North":0,"South":0,...	json	a few seconds ago
Data	{"temp":37.4,"humidity":86,"North":0,"South":0,...	json	a few seconds ago
Data	{"temp":37.4,"humidity":86,"North":0,"South":0,...	json	a few seconds ago
Data	{"temp":37.4,"humidity":86,"North":0,"South":0,...	json	a few seconds ago
Data	{"temp":37.4,"humidity":86,"North":0,"South":0,...	json	a few seconds ago

Node Red :

The screenshot shows the Node-RED interface. The central workspace displays a flow named 'Flow 7'. The flow starts with a '[get] /data' node, followed by a 'function' node, and then an 'http' node. Another path starts from a '[get] /command' node, which branches into two 'http' nodes and an 'IoT platform' node. A 'msg.payload' node is also present. The left sidebar shows the 'input' and 'output' node categories. The right sidebar shows the 'info' panel with a search bar and a list of flows.



Node Red Web UI :

