

PROJECT DEVELOPMENT PHASE


SPRINT-1

Date	12 November 2022
Team ID	PNT2022TMID00923
Project Name	Signs with Smart Connectivity for Better Road Safety

SPRINT-1:

- In sprint-1, we collect temperature and humidity data from OpenWeatherMap website for a particular city as input.
- We have developed the code for the same and to publish it to IBM IoT Watson, node-red and finally display them in MIT APP INVENTOR.
- We have also implemented a condition such that, if humidity is less than 100, then it displays a warning like "PLEASE SLOW DOWN".

PYTHON CODE:

 *PROJECTFINALDND.py - D:/1ibm/PROJECTFINALDND.py (3.7.0)*

File Edit Format Run Options Window Help

```
#connect and send a datapoint "temp" with value integer value into the cloud as a type of event for every 10 seconds
deviceCli.connect()

while True:

    #get sensor data from DHT11

    a = "https://api.openweathermap.org/data/2.5/weather?q=Chennai,%20IN&appid=e2bea247ed9ad643a04d9a8e55499d5f"
    r=requests.get(url=a)
    data=r.json()

    Temp= data['main']['temp']
    Humd= data['main']['humidity']
    data= {'temp':Temp,'humid':Humd}
    dist=random.randint(0,50)
    dis={'dista':dist}

    if (Humd<100):
        warn={'alert':'PLEASE SLOW DOWN!!!!!!'}
```

IBM IoT WATSON PLATFORM:

The screenshot shows the IBM Watson IoT Platform interface. The 'Recent Events' tab is active, displaying a table of events. The event highlighted with a red box is as follows:

Event	Value	Format	Last Received
IoTSensor	{"dista":1}	json	a few seconds ago
IoTSensor	{"alert":"PLEASE SLOW DOWN!!!!!!!"}	json	a few seconds ago
IoTSensor	{"temp":300.14,"humid":94}	json	a few seconds ago
IoTSensor	{"inst":"stop"}	json	a few seconds ago
IoTSensor	{"dista":0}	json	a few seconds ago

At the bottom of the interface, it states "1 Simulation running".

NODE-RED:

The screenshot displays the Node-RED web interface in a browser. The main workspace shows a flow named 'Flow 1' with the following components and connections:

- Input:** An 'IBM IoT' sensor node (blue) with a 'connected' status.
- Processing:** The flow splits into three parallel paths:
 - Temperature Path:** A 'temperature' function node (orange) connected to a 'temp' output node (teal).
 - Humidity Path:** A 'humidity' function node (orange) connected to a 'humid' output node (teal).
 - Alert Path:** An 'alert' function node (orange) connected to a 'function' node (orange).
- Output:** The three paths merge into a single stream that passes through a 'msg.payload' node (green), then a 'msg' node (orange), and finally a '[get] /inst' node (yellow). This is followed by a 'function' node (orange) and an 'http' node (yellow).

The right sidebar shows the 'debug' console with logs for the flow. A red box highlights the following log entry:

```
{
  "temp": 300.14,
  "humid": 94,
  "alert": "PLEASE SLOW DOWN!!!!!!"
}
```

MIT APP INVENTOR:

The screenshot shows the MIT App Inventor web interface. The browser address bar displays `ai2.appinventor.mit.edu/#5724286015635456`. The interface includes a top navigation bar with links like 'Projects', 'Connect', 'Build', 'Settings', and 'Help'. Below this is a green header bar for the project 'ibmapplication' with buttons for 'Screen2', 'Add Screen...', 'Remove Screen', and 'Publish to Gallery'. The left sidebar contains a 'Blocks' palette with categories like 'Built-in', 'Control', 'Logic', 'Math', 'Text', 'Lists', 'Dictionaries', 'Colors', 'Variables', and 'Procedures'. The main 'Viewer' area shows a visual programming script:

- when Web1 GotText** event block.
- do** loop containing:
 - set TextBox1 . Text** to **look up in pairs** key **temp**.
 - pairs** block: **call Web1 . JsonTextDecodeWithDictionaries** with **uri** **responseCode** and **responseContent**.
 - jsonText** block: **get responseContent**.
 - notFound** block: **not found**.
 - set TextBox2 . Text** to **look up in pairs** key **humid**.
 - pairs** block: **call Web1 . JsonTextDecodeWithDictionaries** with **uri** **responseCode** and **responseContent**.
 - jsonText** block: **get responseContent**.
 - notFound** block: **not found**.
 - set TextBox5 . Text** to **look up in pairs** key **alert**.
 - pairs** block: **call Web1 . JsonTextDecodeWithDictionaries** with **uri** **responseCode** and **responseContent**.
 - jsonText** block: **get responseContent**.
 - notFound** block: **not found**.
- Show Warnings** block.

The bottom of the screen shows a Windows taskbar with the date and time 12:57 on 13-11-2022.

The screenshot shows a Node-RED interface with a 'Home' page. The browser address bar displays `node-red-ilupv-2022-11-08.eu-gb.mybluemix.net/ui/#/I/0?socketid=eNA_alHYV6A5fnD8AAAF`. The page features two gauge charts under the heading 'Default':

- humid** gauge: A semi-circular gauge with a red needle pointing to the value **94** units. The scale ranges from 0 to 100.
- temp** gauge: A semi-circular gauge with a red needle pointing to the value **300.14** units. The scale ranges from 0 to 300.

The bottom of the screen shows a Windows taskbar with the date and time 17:00 on 13-11-2022.

USER APP SCREENSHOT:

