Project Delivery Sprint - 3

Date	17 Nov 2022
Team ID	PNT2022TMID04704
Project Name	Smart Farmer - IoT Enabled Smart Farming Application

Sprint	Functional Requirement(Epic)	User Story Number	User Story /Task
Sprint-3	Interface for connecting to IBM IoT cloud.	USN-4	Temperature and soil moisture sensor sends the data to the cloud via IBM Watson service.

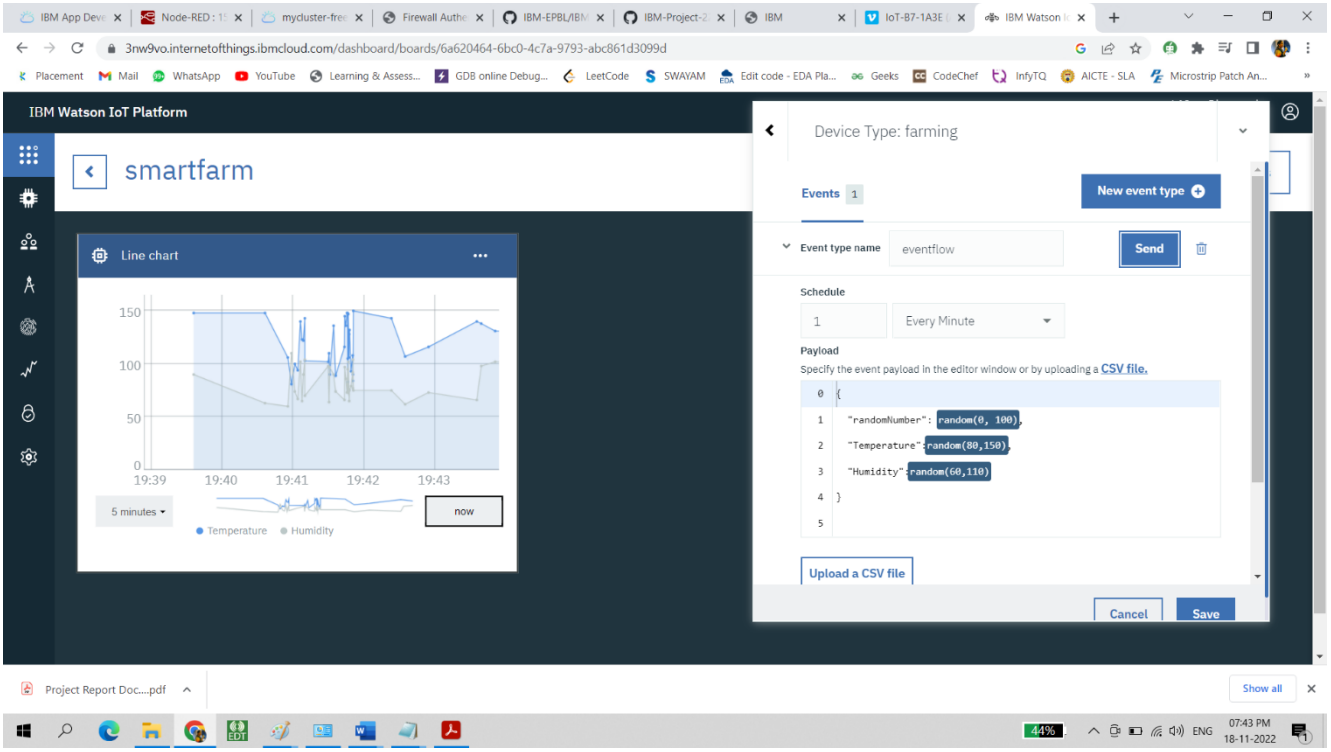
Connecting IOT Simulator to IBM Watson IOT Platform

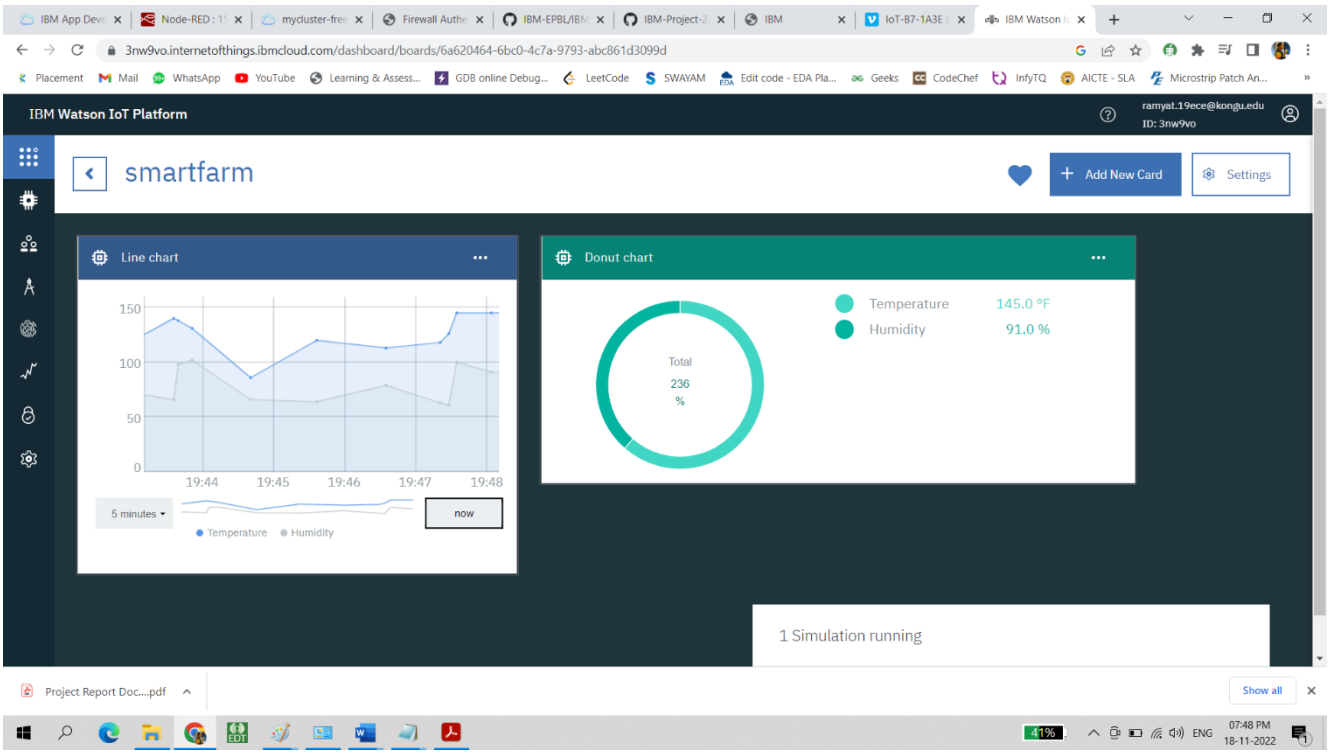
Give the credentials of your device in IBM

Watson Mycredentials given to simulator are:

```
organization = "3nw9vo"
deviceType = "farming"
deviceId = "application"
authMethod = "token"
authToken = "87654321"
```

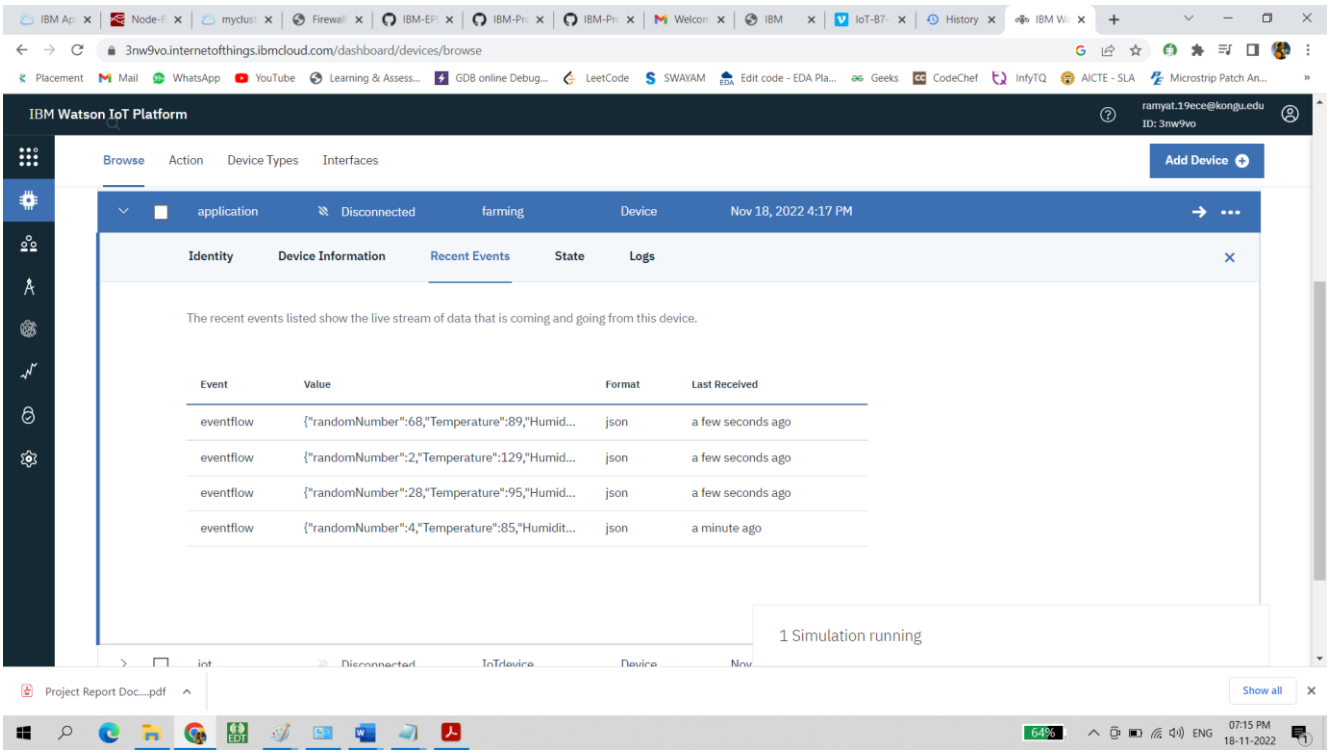
- You can see the received data in graphs by creating cards in Boards tab
- You will receive the simulator data in cloud


IBM Watson IoT Platform
smartfarm
Line chart
150
100
50
0
19:39
19:40
19:41
19:42
19:43
5 minutes
now
Temperature
Humidity
Device Type: farming
Events 1
New event type
Event type name
eventflow
Send
Schedule
1
Every Minute
Payload
Specify the event payload in the editor window or by uploading a CSV file.
{
"randomNumber": random(0, 100),
"Temperature": random(80,150),
"Humidity": random(60,110)
}
Upload a CSV file
Cancel
Save
Project Report Doc....pdf
Show all
44%
ENG
07:43 PM
18-11-2022


IBM Watson IoT Platform
ramyat.19ece@kongu.edu
ID: 3nw9vo
smartfarm
Add New Card
Settings
Line chart
150
100
50
0
19:44
19:45
19:46
19:47
19:48
5 minutes
now
Temperature
Humidity
Donut chart
Total
236
%
Temperature
145.0 °F
Humidity
91.0 %
1 Simulation running
Project Report Doc....pdf
Show all
41%
ENG
07:48 PM
18-11-2022

- You can see the received data in Recent Events under your device
- Data received in this format (json)

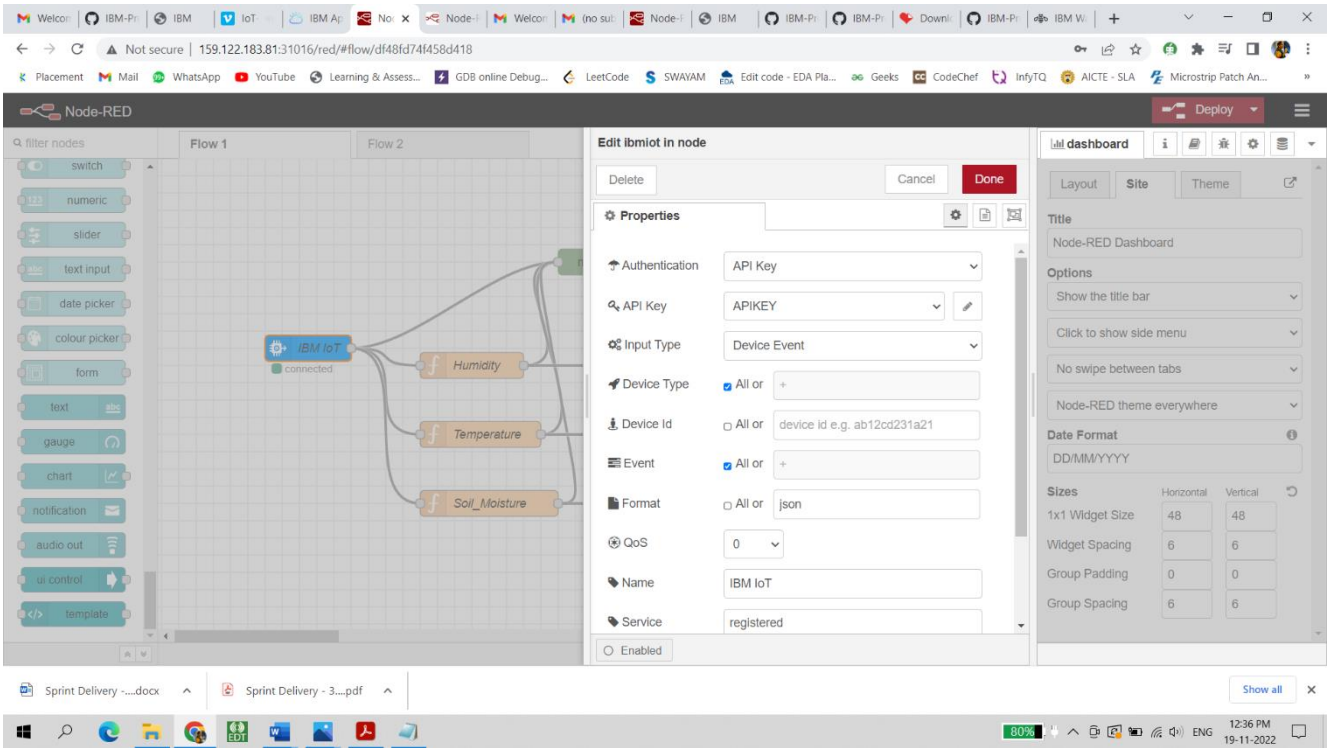

```
{
"Moisture":89,
"temp":96.0,
"Humid":89
}
```

Sprint	Functional Requirement(Epic)	User Story Number	User Story /Task
Sprint-3	Create Node Red Simulator	USN - 5	Create Node-Red Service and create a web application

Configuration of Node-Red to collect IBM cloud data

The node IBM IOT App In is added to Node-Red workflow. Then the appropriate device credentials obtained earlier are entered into the node to connect and fetch device telemetry to Node-Red.

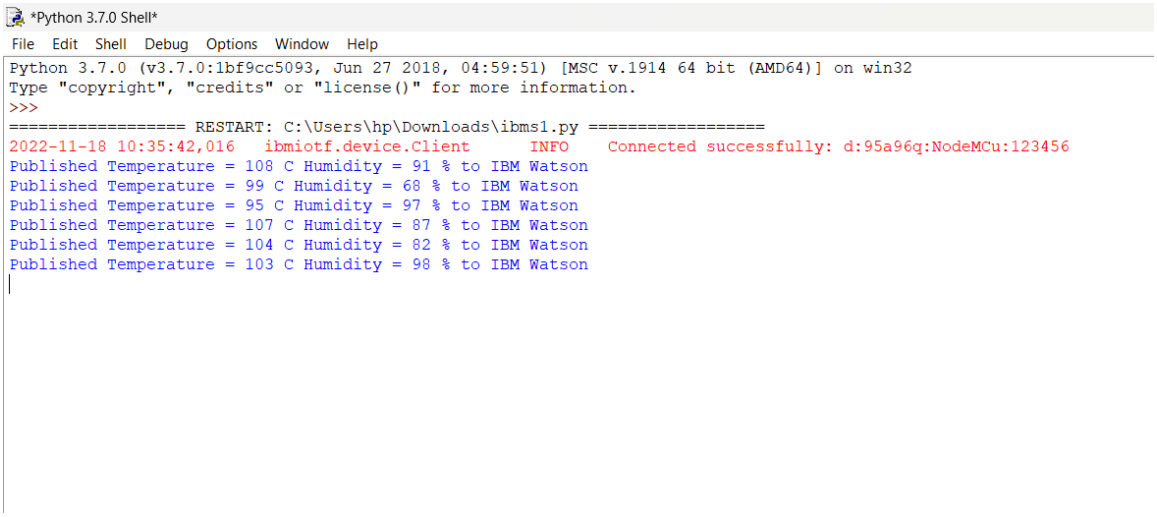


- Once it is connected Node-Red receives data from the device.
- Display the data using debug node for verification.
- Connect function node and write the Java script code to get each reading separately.
- The Java script code for the function node is:
- **msg.payload = msg.payload.Temperature**

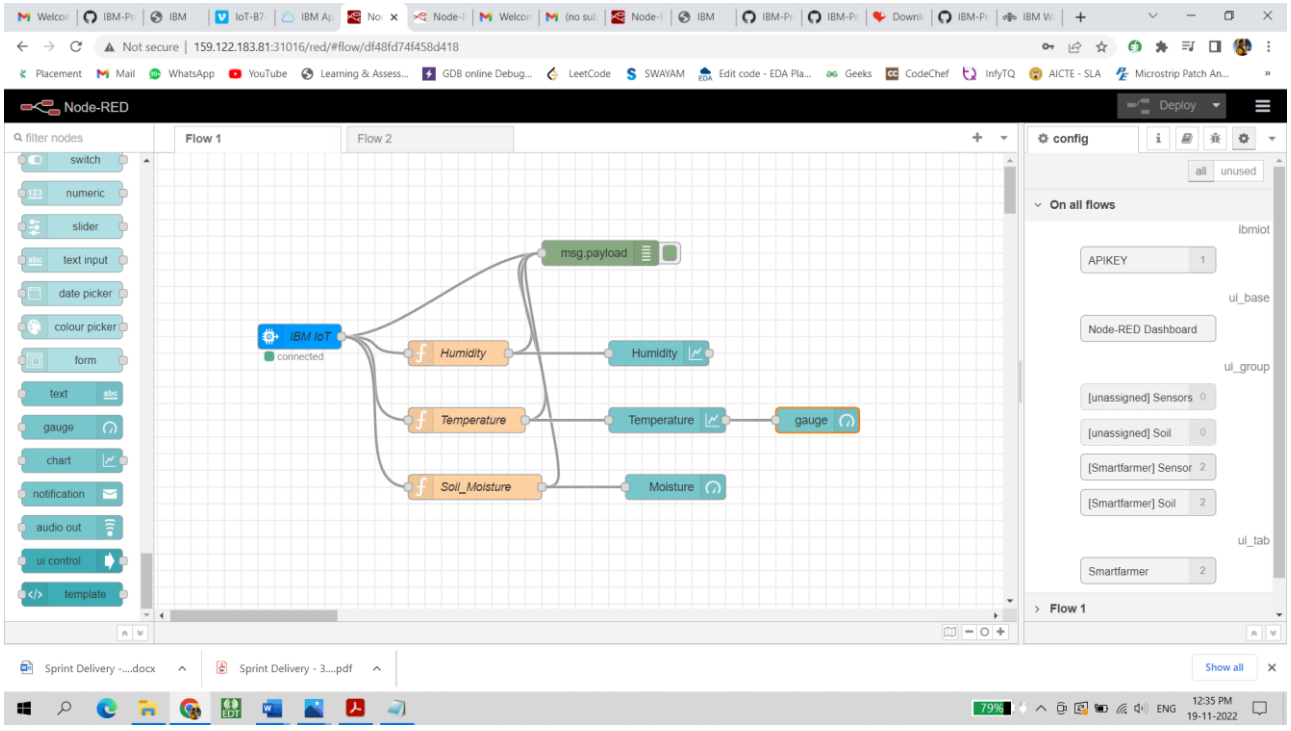
return msg;

- Finally connect Gauge nodes from dashboard to see the data in UI.

- Data send by the python code



- Data received from the cloud in Node-Red console



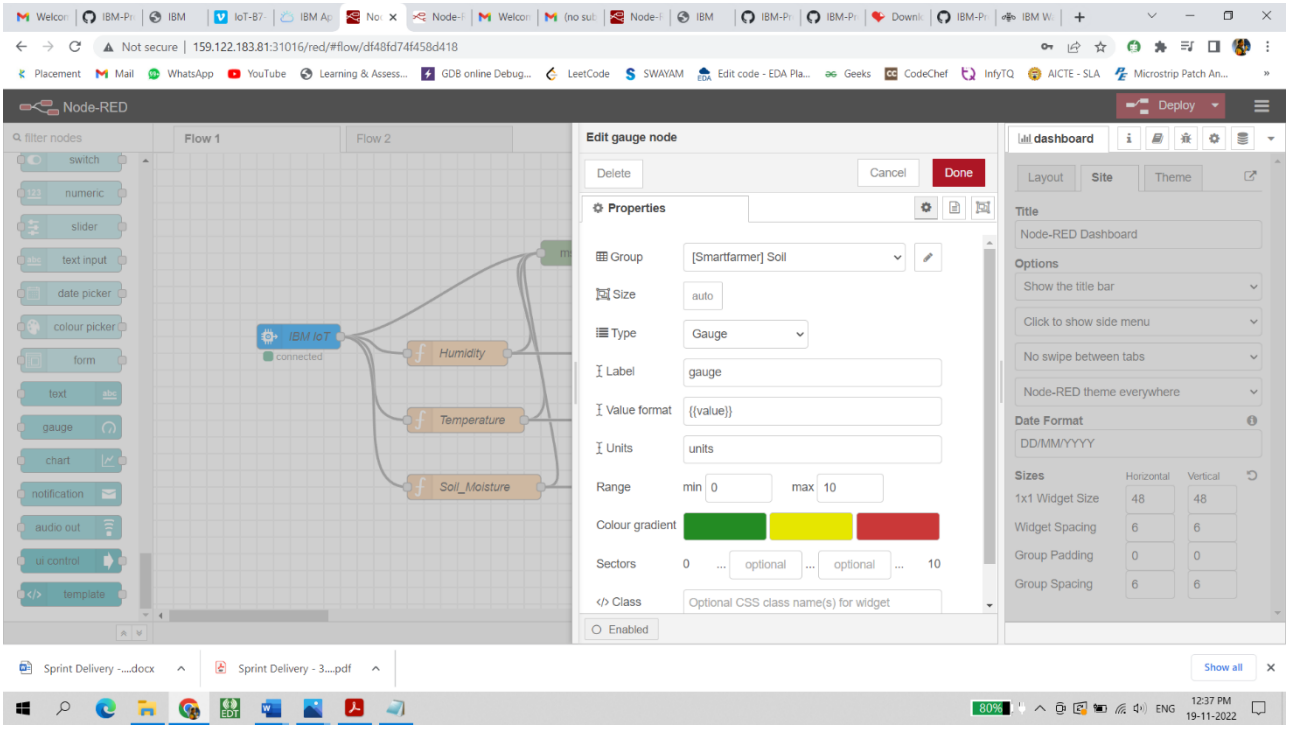
- Nodes connected in following manner to get each reading separately.

Configuration of Node-Red to collect data from Open Weather

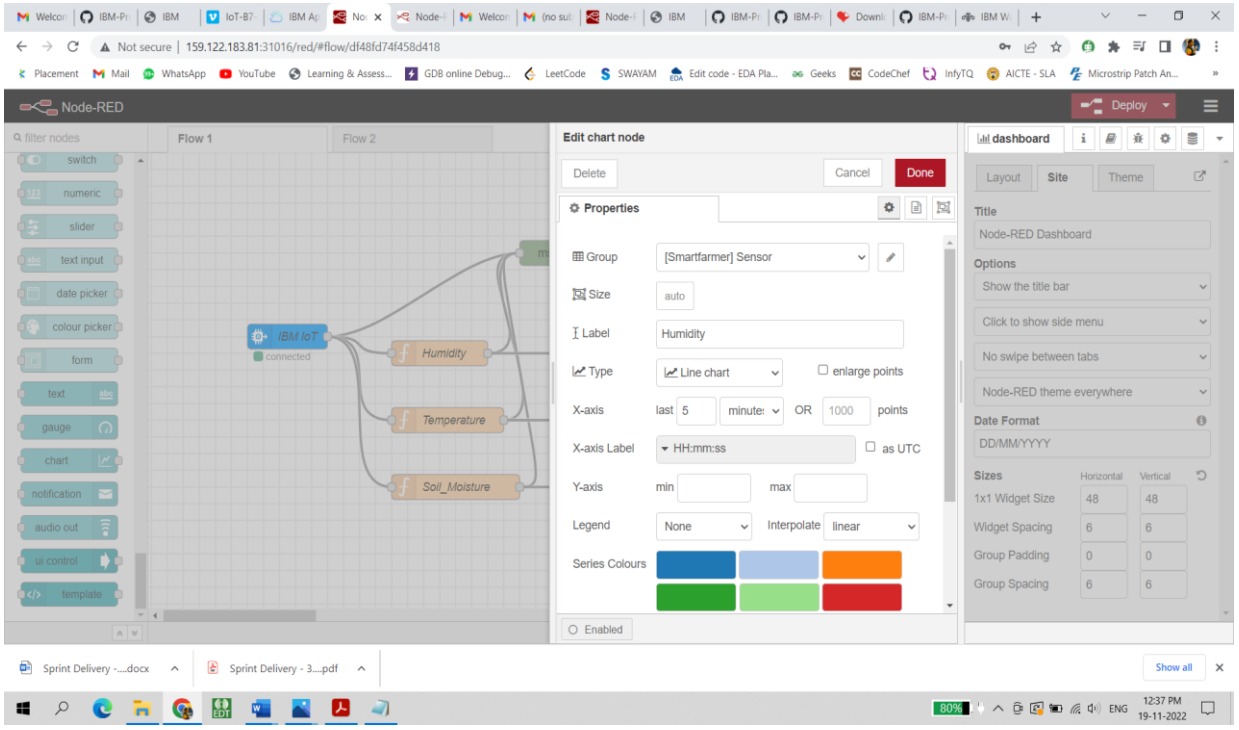
- The Node-Red also receive data from the Open Weather API by HTTPGET request. An inject trigger is added to perform HTTP request for every certain interval.
- The link to get open weather API : https://api.openweathermap.org/data/2.5/weather?lat=11.4383197&lon=77.5402674&appid=124d808d2039542453a0b1b05f37e900
- The data we receive from Open Weather after request is in below JSONformat.
- {"coord":{"lon":77.5403,"lat":11.4383},"weather":[{"id":804,"main":"Clouds","description":"overcast clouds","icon":"04d"}],"base":"stations","main":{"temp":300.33,"feels_like":303.19,"temp_min":300.33,"temp_max":300.33,"pressure":1009,"humidity":79,"sea_level":1009,"grnd_level":986},"visibility":10000,"wind":{"speed":2.3,"deg":113,"gust":3.05},"clouds":{"all":97},"dt":1668332957,"sys":{"country":"IN","sunrise":1668300334,"sunset":1668342165},"timezone":19800,"id":1270947,"name":"Gobichettipalayam","cod":200}
- In order to parse the JSON string we use Java script functions and geteach parameters

```
msg.payload = {"temp" : global.get("t") ,
                "Humid" : global.get("h") ,
                "Moisture" : global.get("m")
               }
return msg;
```

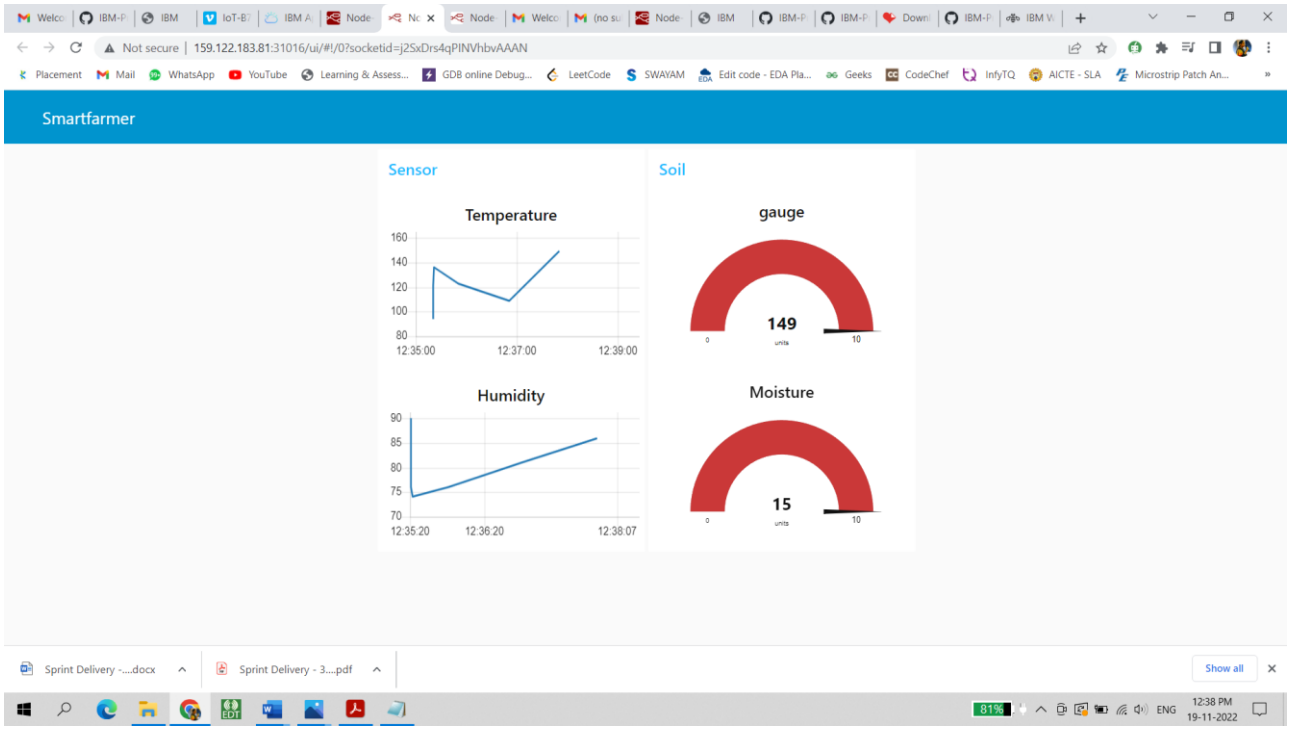
- Then we add Gauge and text nodes to represent data visually in UI



- Then we add Chart and text nodes to represent data visually in UI



- You can the data in the node-red dashboard


Not secure | 159.122.183.81:31016/ui/#!/0?socketid=j2SxDrs4qPINVhbvAAAN
Smartfarmer
Sensor
Temperature
160
140
120
100
80
12:35:00
12:37:00
12:39:00
Humidity
90
85
80
75
70
12:35:20
12:36:20
12:38:07
Soil
gauge
149
units
0
10
Moisture
15
units
0
10
Sprint Delivery -....docx
Sprint Delivery - 3....pdf
Show all
81%
ENG
12:38 PM
19-11-2022