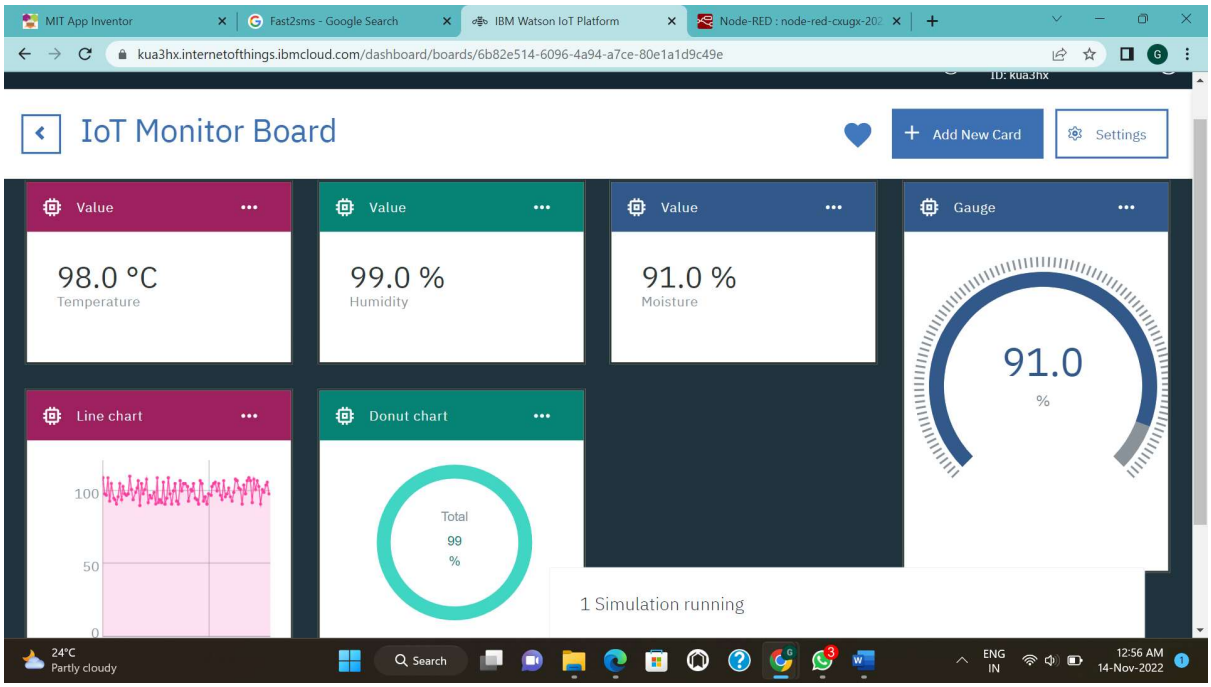
Project Development Phase

Delivery Of Sprint-2

TITLE	**Smart Farmer-IOT Enabled Smart Farming Application**
DOMAIN NAME	INTERNET OF THINGS
TEAM ID	PNT2022TMID23830
Project Name	Smart Farmer - IoT Enabled Smart Farming Application
Leader Name	GOWSALYA L
Team Members Name	DEEPIKA B K MEGAVARSHINI G MONISHA N
MENTOR NAME	THIRUPPATHI M

Building Project Connecting IoT Simulator to IBM Watson IoT Platform

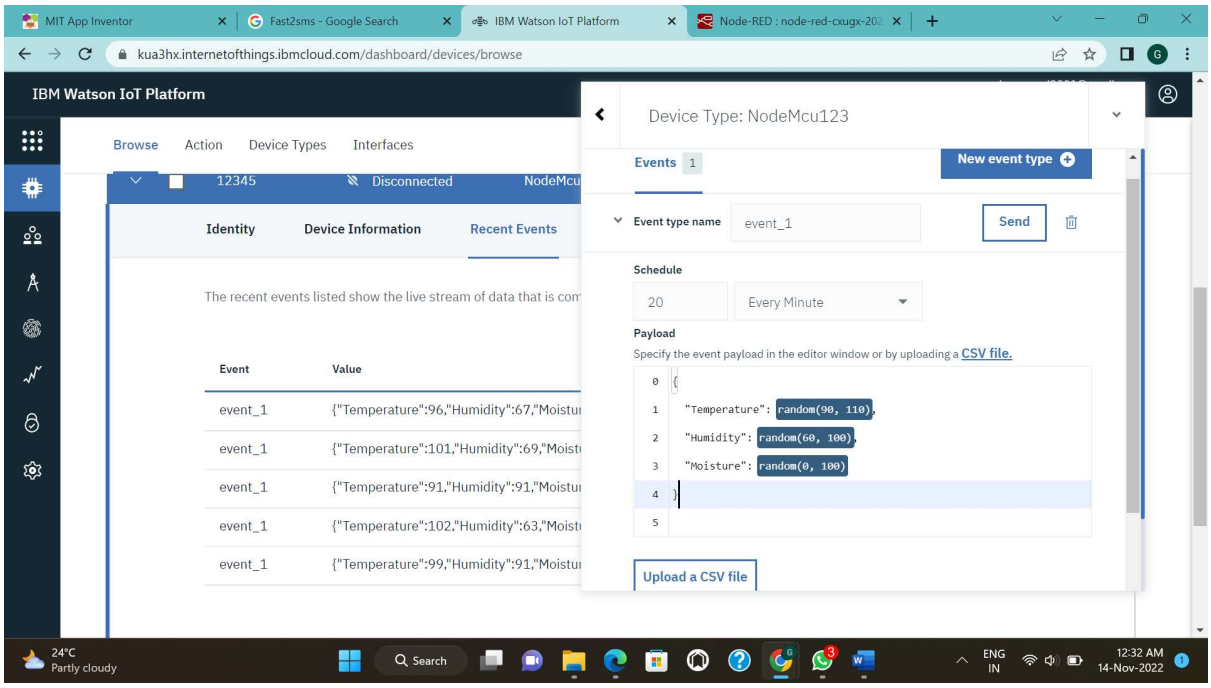
- Open link provided in below image
- Give the credentials of your device in IBM Watson Platform
- Click on connect
- My credentials given to simulator are:
  - **API**: a-kua3hx-bm5z9ikjfu
  - **Device type**: NodeMcu123
  - **Token**: 123456789
  -



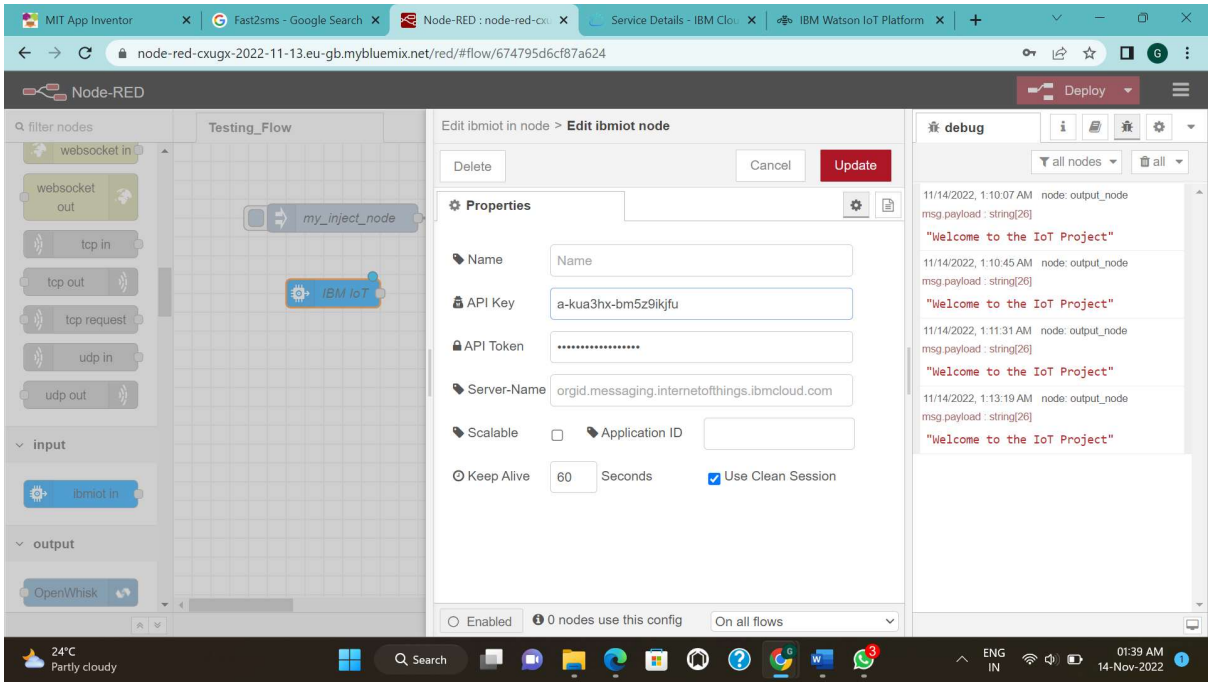
You can see the received data in graphs by creating cards in Boards tab

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

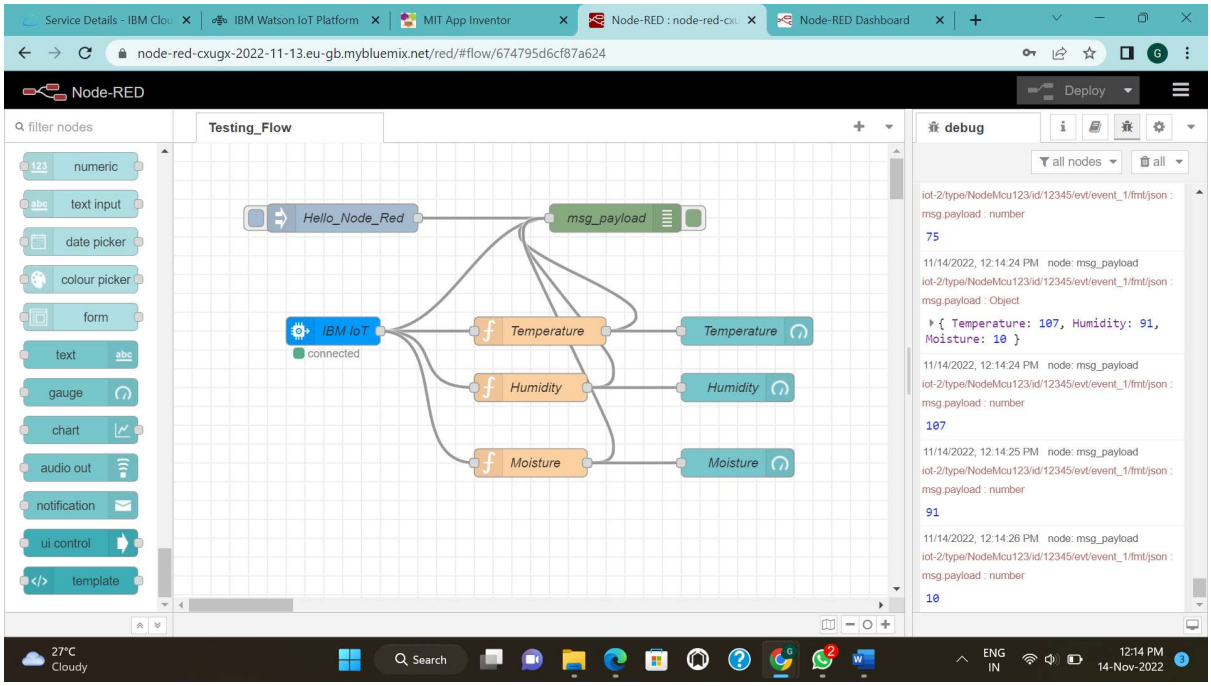
```
{
    "d":
    {
        ▪ "temperature": 96,
        ▪ "humidity": 67,
        ▪ "Moisture ": 25
    }
}
```



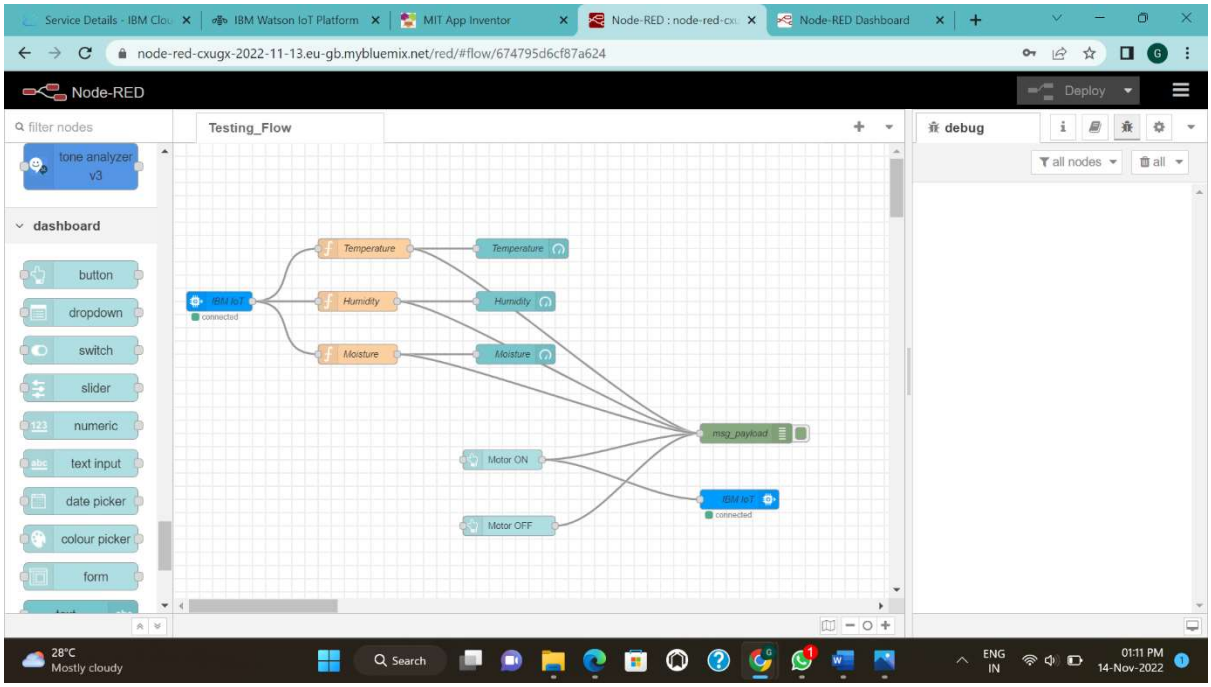
Configuration of Node-Red to collect IBM cloud data



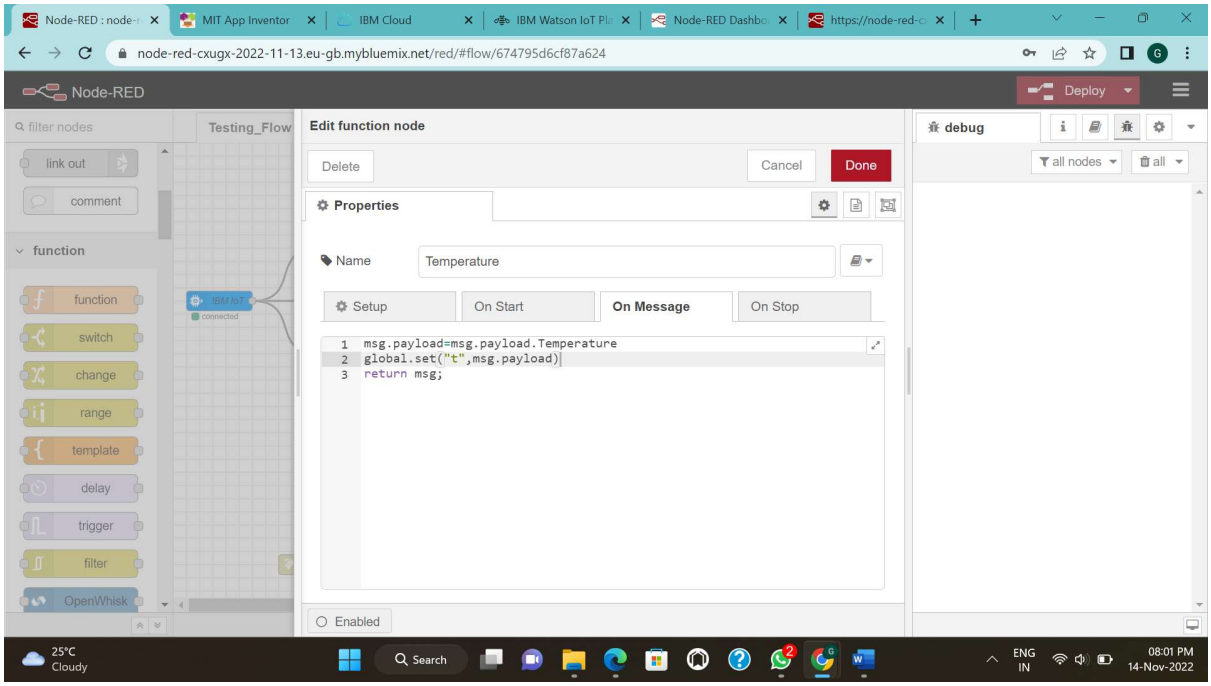
- The node IBM IoT App In is added to Node-Red workflow. Then the appropriate device credentials obtained earlier are entered into the node to connect and fetch device telemetry to Node-Red.
- Once it is connected Node-Red receives data from the device Display the data using debug node for verification
- 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.d.temperature
    - return msg;
- Finally connect Gauge nodes from dashboard to see the data in UI



➢ Data received from the cloud in Node-Red console



➢ Nodes connected in following manner to get each reading separately



This is the Java script code I written for the function node to get Temperature separately.

Configuration of Node-Red to collect data from OpenWeather

- The Node-Red also receive data from the OpenWeather API by HTTP GET request.
- An inject trigger is added to perform HTTP request for every certain interval.
- The data we receive from OpenWeather after request is in below JSON format:

{

"coord":

{

"lon":79.85,"lat":14.13},

"weather":[{"id":803,"main":"Clouds","description":"brokencloud","icon":"04n"}],

"base":"stations",

"main":{"temp":30759,

"feels_like":305.5,

"temp_min":307.59,

"temp_max":307.59,

"pressure":1002,

"humidity":35,

"sea_level":1002,

"grnd_level":1000},

"wind":{"speed":6.23,"deg":170 },

"clouds":{"all":68},

"dt":1589991979,

"sys":{"country":"IN","sunrise":1589933553",sunset":1589979720},

"timezone":19800,

"id":1270791,

"name":"Gūdūr",

"cod":20 0}

- In order to parse the JSON string we use Java script functions and get each parameters

```
var temperature = msg.payload.main.temp;
temperature = temperature-273.15;
return {payload : temperature.toFixed(2)};
```

In the above Java script code, we take temperature parameter into a new variable and convert it from kelvin to Celsius. Then we add Gauge and text nodes to represent data visually in UI