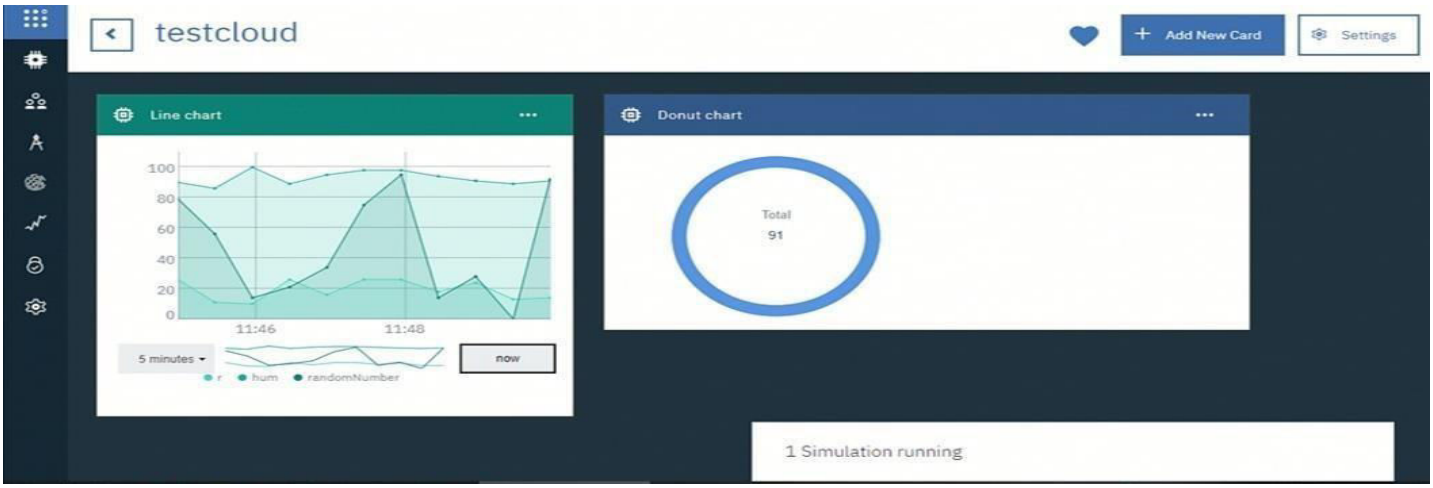
Team ID	PNT2022TMID03184
Project Name	Project – Smart farmer-IoT enabled smart farming application.

Sprint-2

Building Project
Connecting IoT Simulator to IBM Watson IoT Platform
Give the credentials of your device in IBM Watson IoT PlatformClick on connect
My credentials given to simulator are:

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

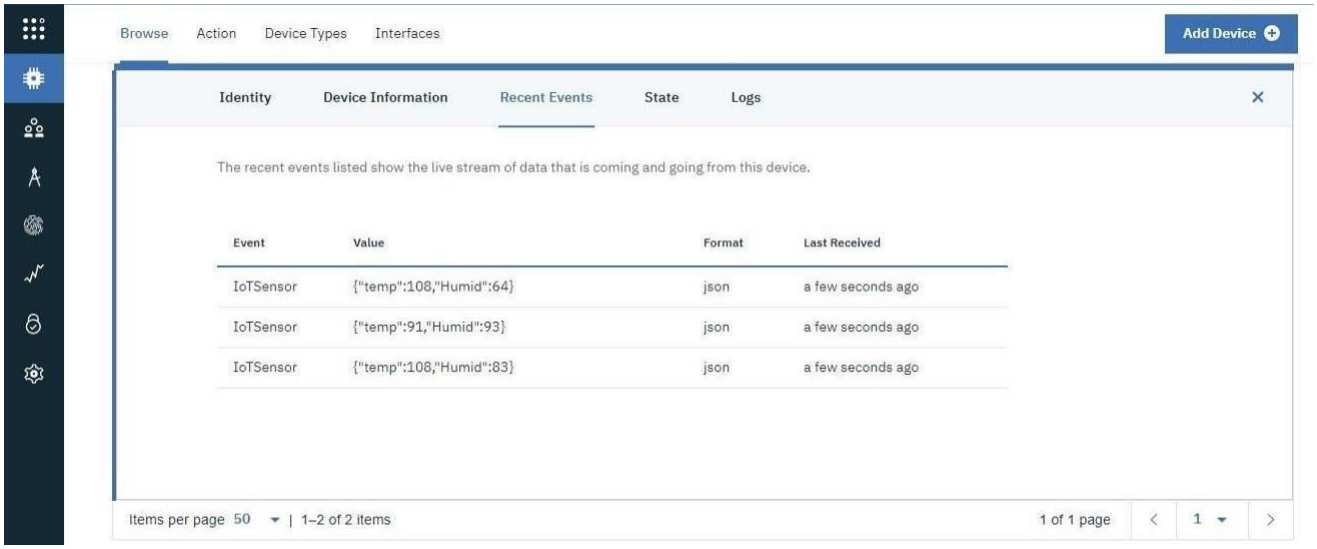


Data received in this format(json)

{
"d": {

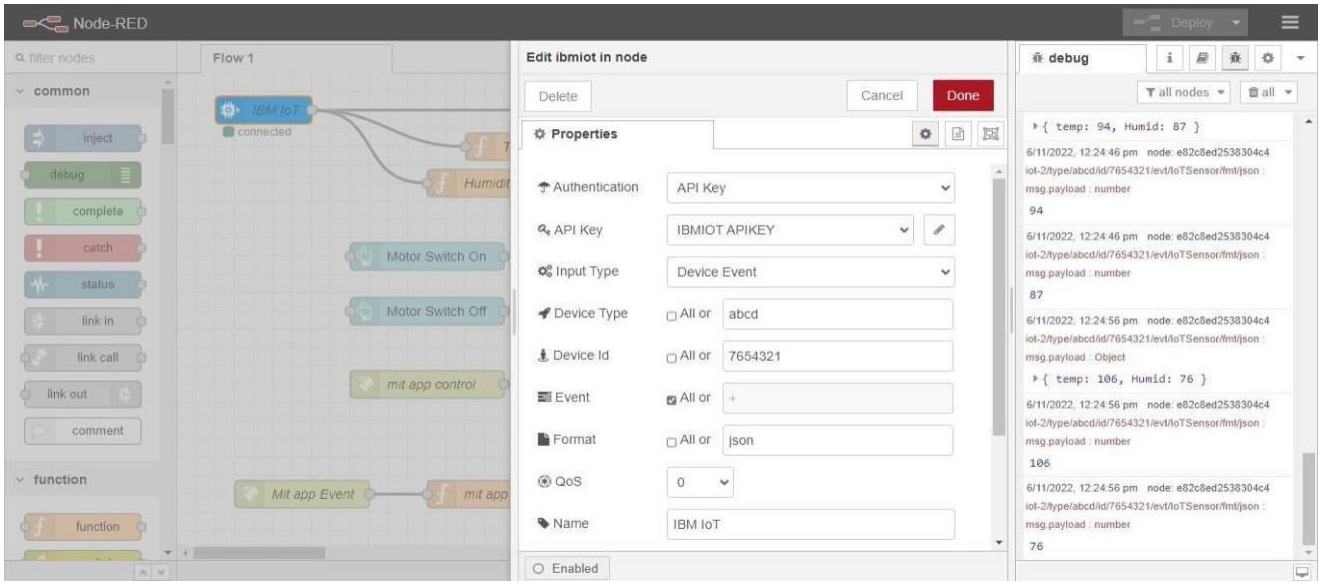
"name": "abcd",
"temperature": 17,
"humidity": 76,
"Moisture ": 25
}

}

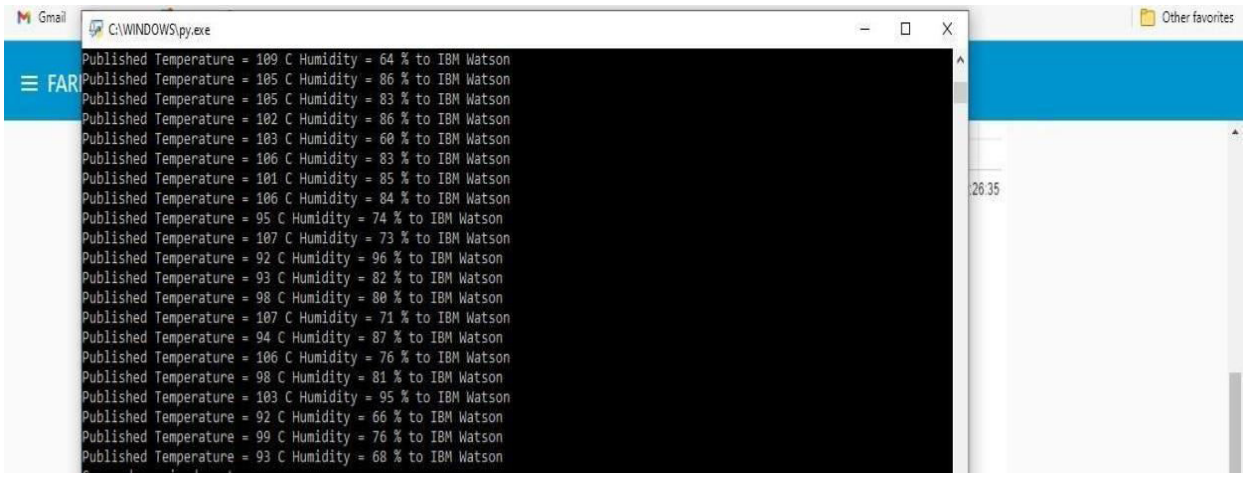


Configuration of Node-Red to collect IBM cloud data
The node IBM IoT App In is added to Node-Red workflow. Then the appropriatedevice credentials obtained earlier are entered into the node to connect and fetch device telemetry to Node-Red.

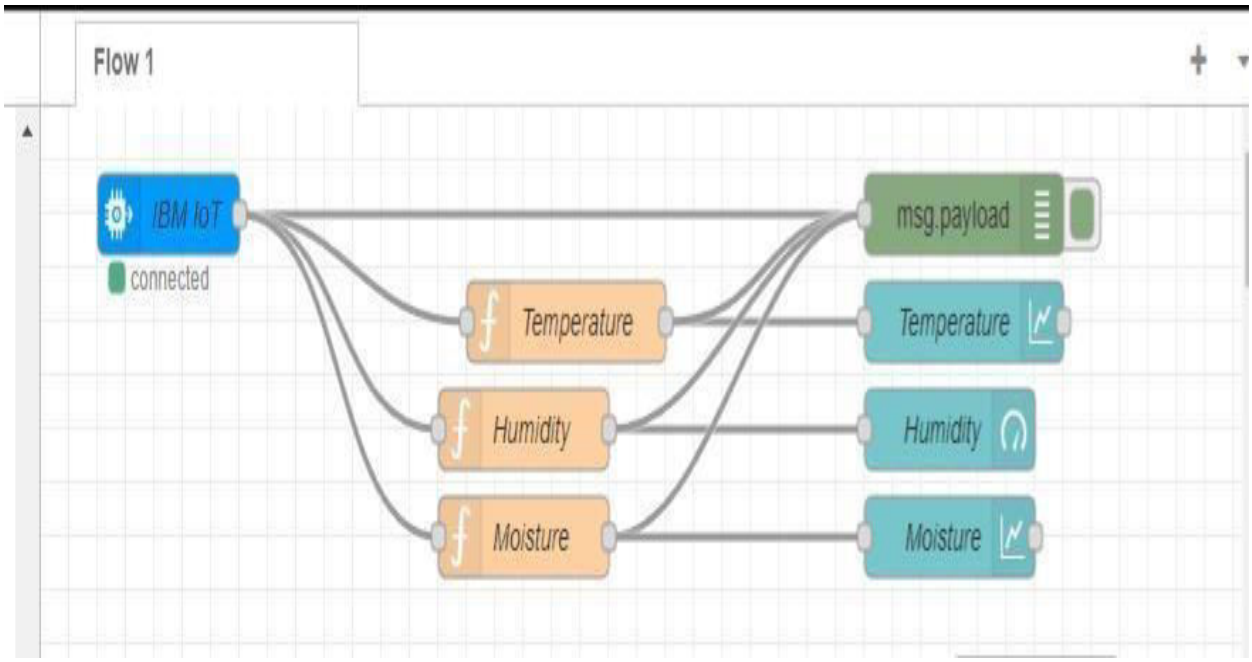
Once it is connected Node-Red receives data from the deviceDisplay the data using debug node



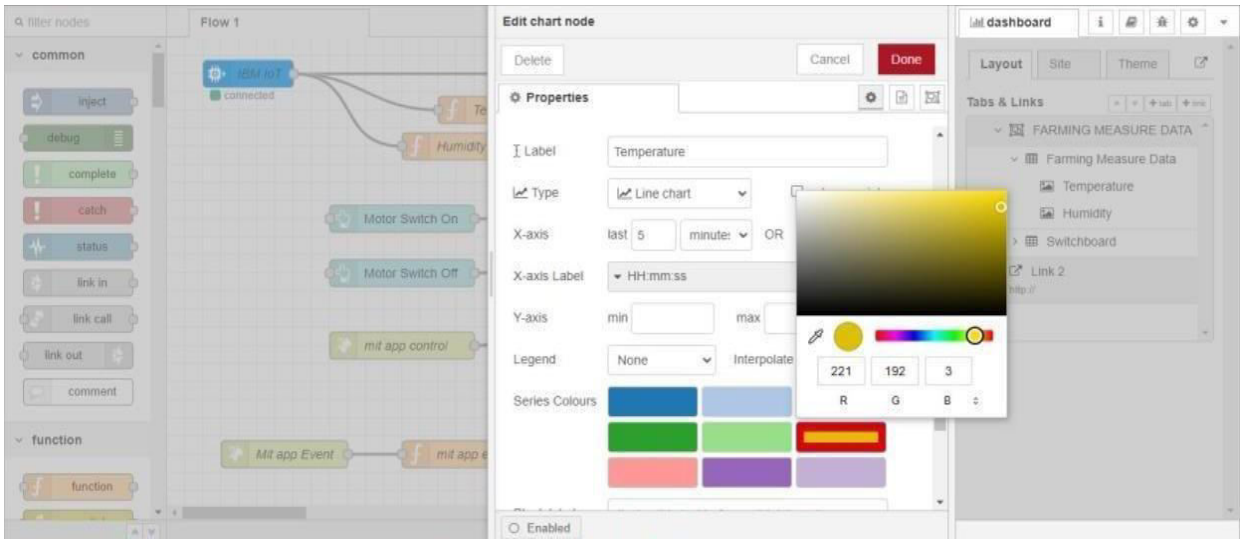
for verification
Connect function node and write the Java script code to get each readingseparately.
The Java script code for the function node is: msg.payload=msg.payload.d.temperature returnmsg;
Finally connect Gauge nodes from dashboard to see the data in UI



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 Temperatureseparately.
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.
HTTP request node is configured.

In order to parse the JSON string we use Java script functions and get eachparameters
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 variableand convert it from kelvin to Celsius
Then we add Gauge and text nodes to represent data visually in UI