

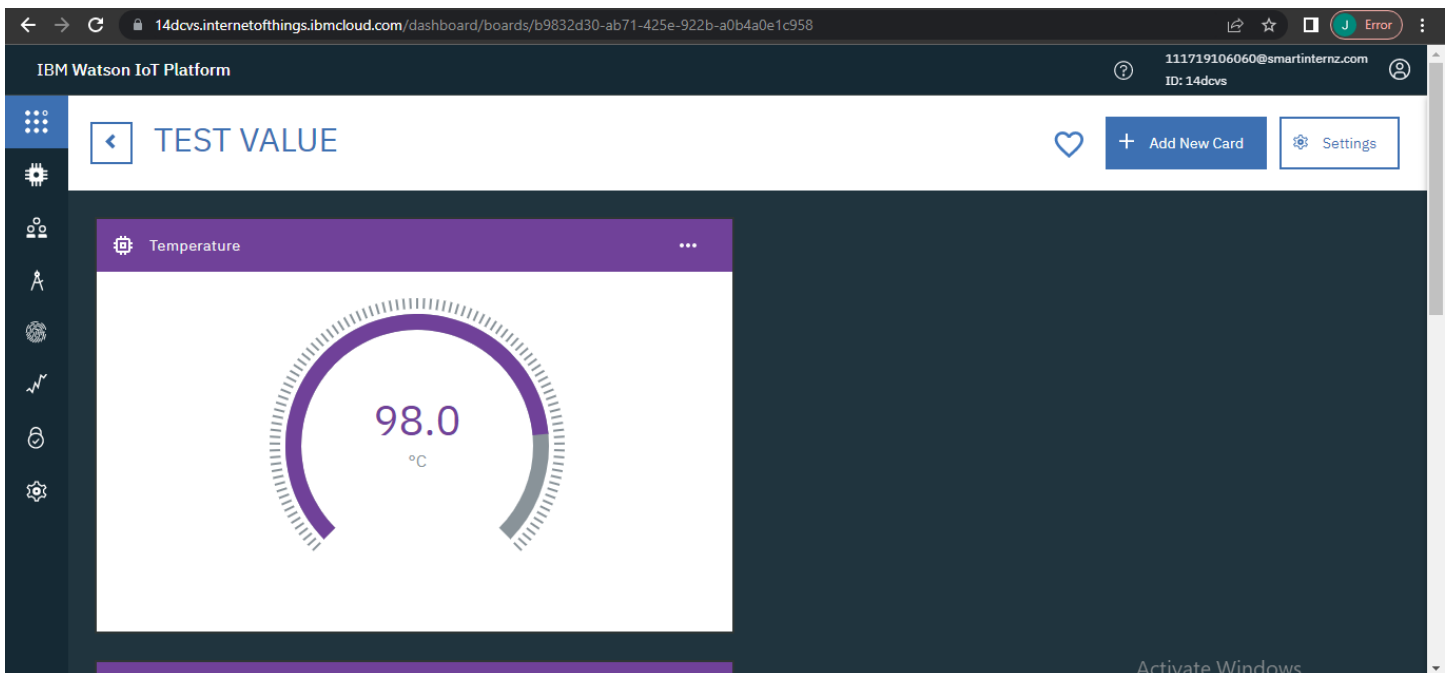
Sprint Delivery – 2

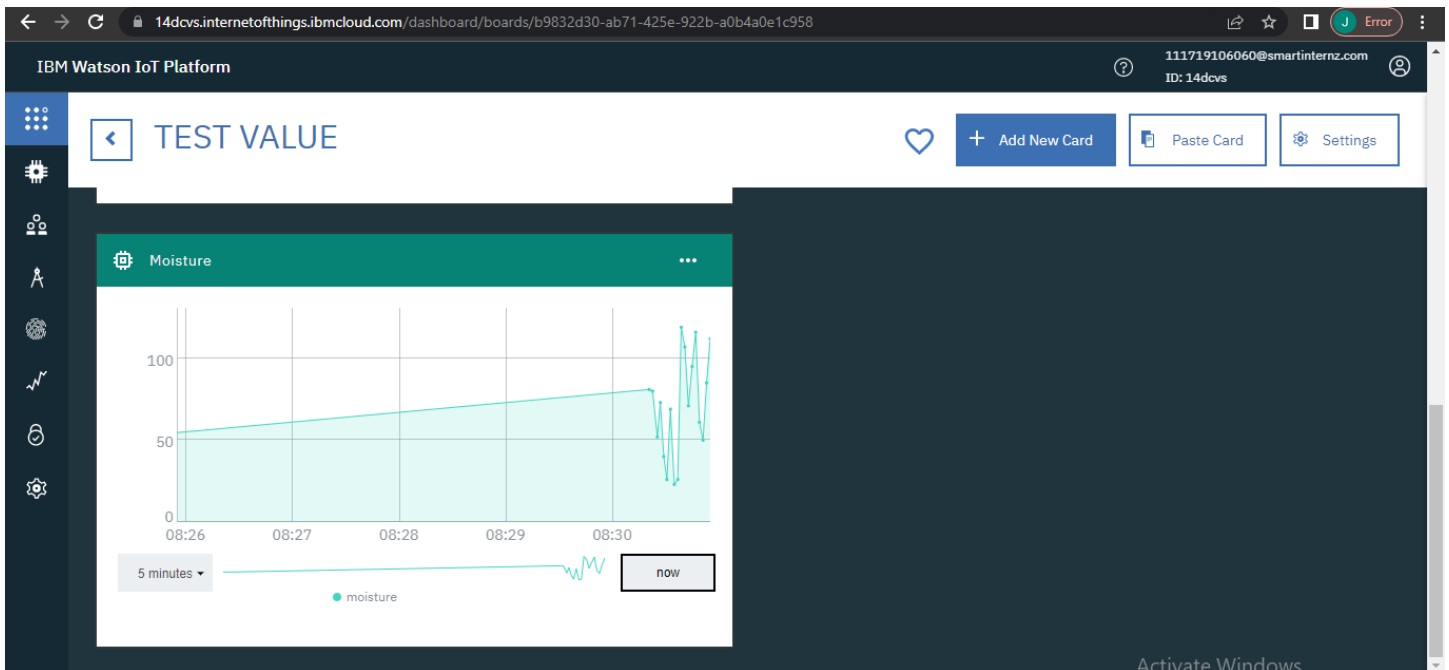
Team Id	PNT2022TMID16026
Project Name	Smart Farmer – IOT Enabled Smart Farming Application

5. Building Project Connecting IOT Simulator to IBM Watson IOT Platform

- Open link provided in above section 4.3
- Give the credentials of your device in IBM Watson IOT Platform
- Click on connect My credentials given to simulator are:
- OrgID: 14dcvs
- Api key : a-14dcvs-xzoonjld1n
- Device type: Device1
- Authentication token: FSaB@(rp7jt2hUXdul
- Device ID: 12345
- Device Token: 87654321
- 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": "Device1",  
  "temperature": 71,  
  "humidity": 84,  
  "Moisture ": 78 } }
```

IBM Watson IoT Platform

111719106060@smartinternz.com
ID: 14dcvs

Browse Action Device Types Interfaces

Add Device

12345 Connected Device1 Device Nov 17, 2022 5:44 PM

Identity Device Information Recent Events State Logs

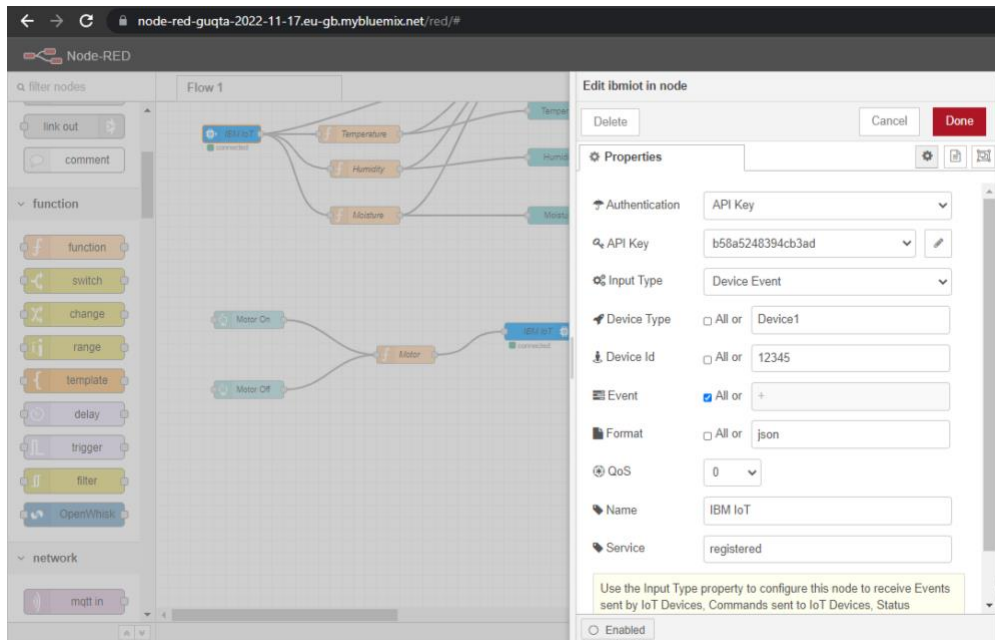
The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
status	{"temperature":2,"humidity":51,"moisture":70}	json	a few seconds ago
status	{"temperature":15,"humidity":55,"moisture":61}	json	a few seconds ago
status	{"temperature":24,"humidity":16,"moisture":89}	json	a few seconds ago
status	{"temperature":76,"humidity":66,"moisture":108}	json	a few seconds ago
status	{"temperature":-18,"humidity":45,"moisture":50}	json	a few seconds ago

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

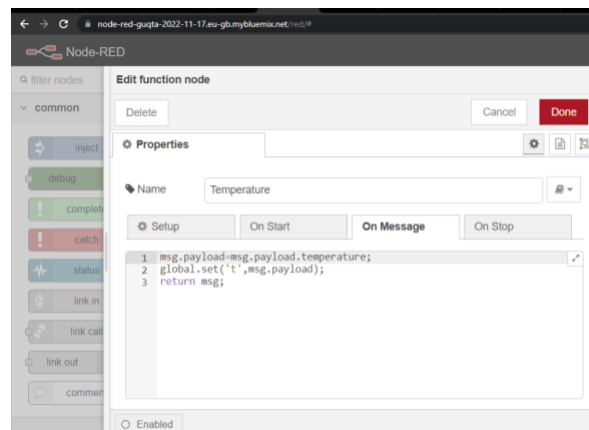
Once it is connected Node-Red receives data from the device, Display the data using debug node for verification. Connect function node and write the Java script code to get each reading separately.



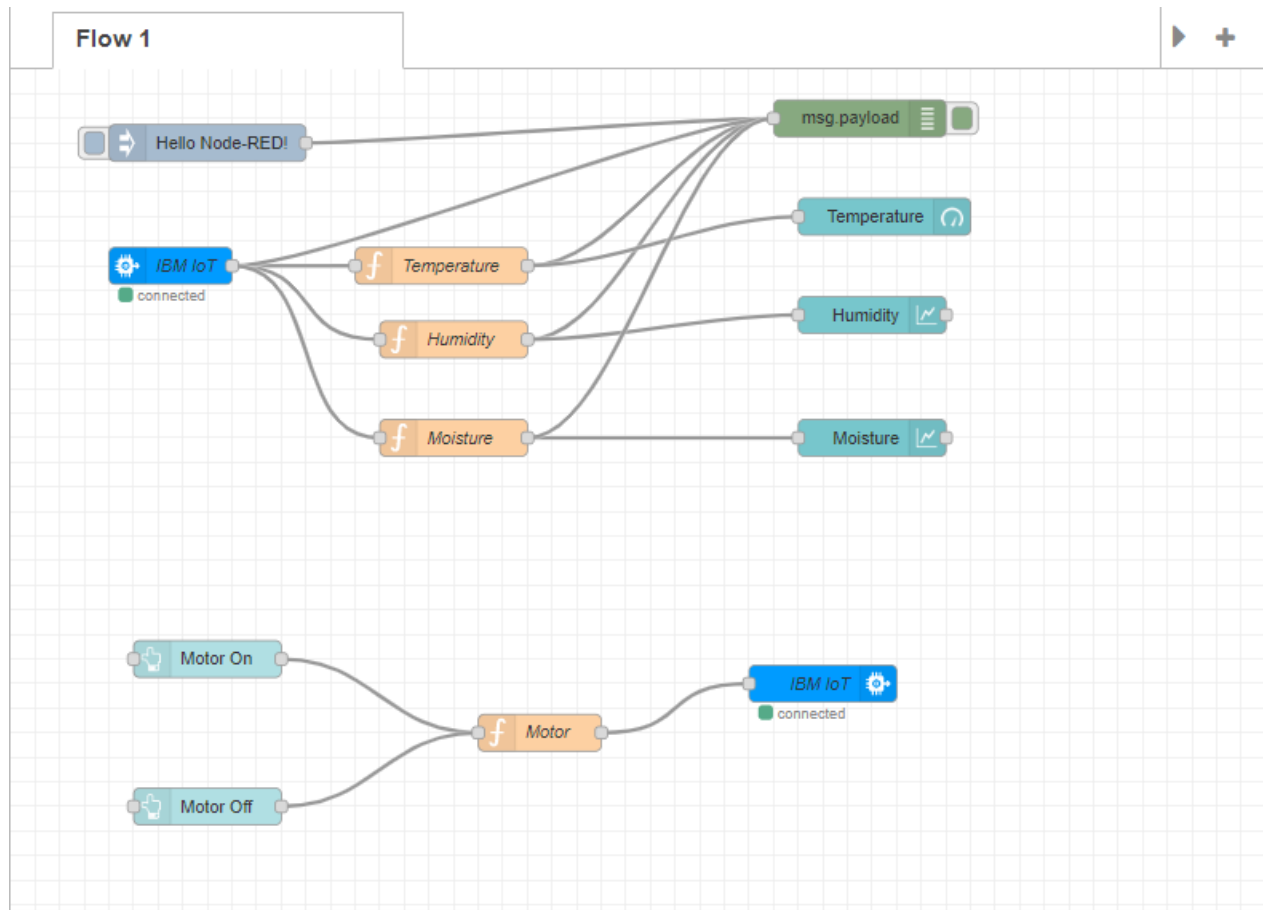
The Java script code for the function node is:

```
msg.payload = msg.payload.temperature;  
  
return msg;
```

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



Nodes connected in following manner to get each reading separately.



Data received from the cloud in Node-Red console.

The screenshot shows the Node-RED interface with the flow diagram from the previous image. The debug console on the right displays the following messages:

```
11/18/2022, 8:40:34 AM node: f2f2649a.0d0d98  
iot-2/type/Device1/id/12345/evt/status/fmt/json :  
msg.payload : number  
95  
11/18/2022, 8:40:35 AM node: f2f2649a.0d0d98  
iot-2/type/Device1/id/12345/evt/status/fmt/json :  
msg.payload : number  
89  
11/18/2022, 8:40:36 AM node: f2f2649a.0d0d98  
iot-2/type/Device1/id/12345/evt/status/fmt/json :  
msg.payload : number  
62  
11/18/2022, 8:40:37 AM node: f2f2649a.0d0d98  
iot-2/type/Device1/id/12345/evt/status/fmt/json :  
msg.payload : Object  
{ temperature: 107, humidity: 22,  
moisture: 36 }  
11/18/2022, 8:40:38 AM node: f2f2649a.0d0d98  
iot-2/type/Device1/id/12345/evt/status/fmt/json :  
msg.payload : number
```

Data from Node-Red to MIT app Inventor

