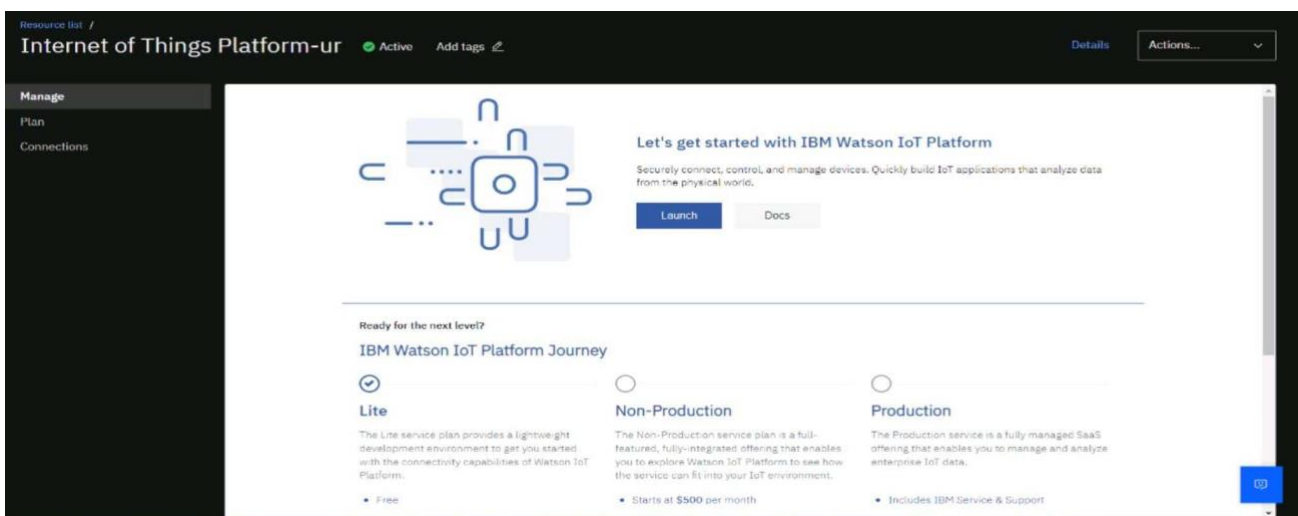


## Sprint 2

**Software-** Create device in the IoT Watson Platform,  
workflow for IoT Scenarios using Local Node

Date	8 October 2022
Team ID	PNT2022TMID23451
Project Name	Project – Smart Farmer-IoT Enabled smart Farming Application
Maximum Marks	4 Marks

### Launch IBM Watson IoT Platform:



### Steps to configure:

- Create an account in IBM cloud using your email ID
- Create IBM Watson Platform in services in your IBM cloud account
- Launch the IBM Watson IoT Platform
- Create a new device
- Give credentials like device type, device ID, Auth. Token
- Create API key and store API key and token elsewhere.

## Create a new device:

Browser Action Device Types Interfaces Add Device

Search by Device ID Device Simulator

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
12	Disconnected	abcd	Device	Oct 19, 2022 9:52 AM	

Identity Device Information Recent Events State Logs

Device ID: 12  
Device Type: abcd  
Date Added: Oct 19, 2022 9:52 AM  
Added By: dhivya0455@gmail.com  
Connection Status: Disconnected

Items per page: 50 | 1-1 of 1 item 1 of 1 page

1 Simulation running

## IoT Simulator:

In our project in the place of sensors we are going to use IoT sensor simulator which give random readings to the connected cloud. The link to simulator: <https://watson-iot-sensor-simulator.mybluemix.net/>

## Connecting IoT Simulator to IBM Watson IoT Platform:

My credentials given to simulator are:

Org: 1x108d

API: a-1x108d-p5eyywn2eu

Auth Token: GpIJ5spsrx0ZB\*RLmJ

Device Type: abcd

Device ID: 12

Device Token: 12345678

smart home Add New Card Settings

Line chart Donut chart Gauge

100 80 60 40 20 0

20:39 20:40 20:41 20:42 20:43

5 minutes temp Humid now

Total 86.0

86.0

Chat 1 Simulation running

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)

## Configuring IBM-IoT to Node-RED connection

The screenshot displays the IBM IoT Dashboard interface. On the left, a sidebar contains navigation icons. The main area is divided into two panels. The left panel shows a table of recent events for a device with ID 12, which is currently disconnected. The table lists five 'eventflow' events, each containing a JSON payload with 'randomNumber', 'temp', and 'Humid' values. The right panel shows the configuration for a device type named 'abcd'. It includes a 'Send' button, a 'Schedule' dropdown set to 'Every Minute', and a 'Payload' section with a JSON template: 

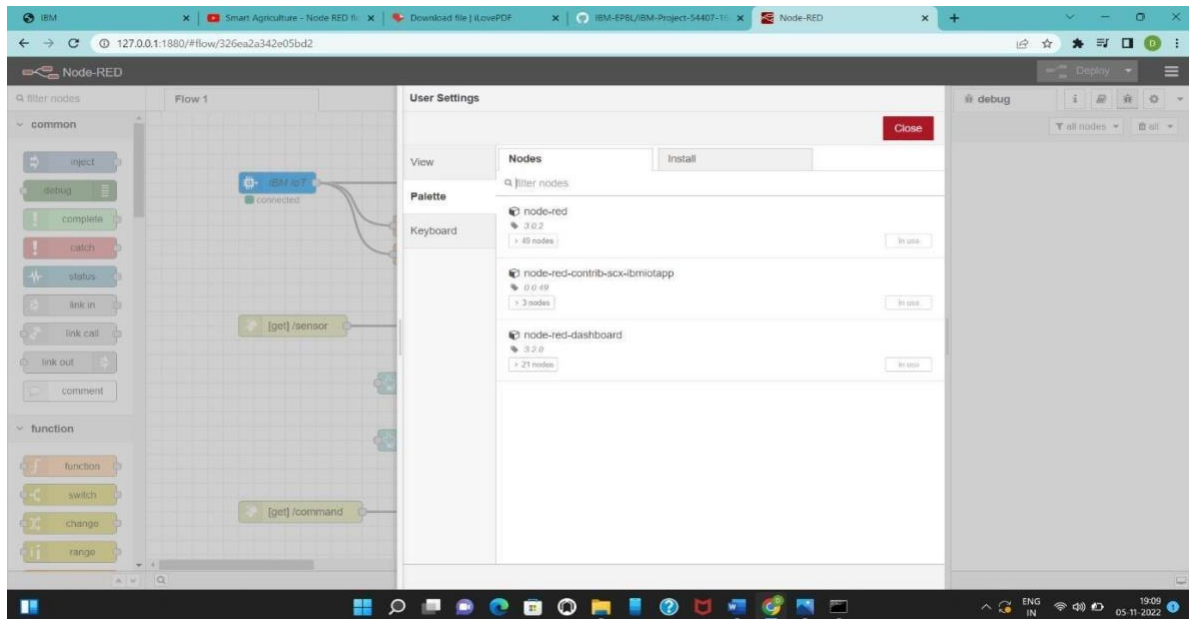
```
{
  "randomNumber": random(0, 100),
  "temp": random(90, 110),
  "Humid": random(60, 100)
}
```

. Below the payload is an 'Upload a CSV file' button and 'Cancel' and 'Save' buttons at the bottom.

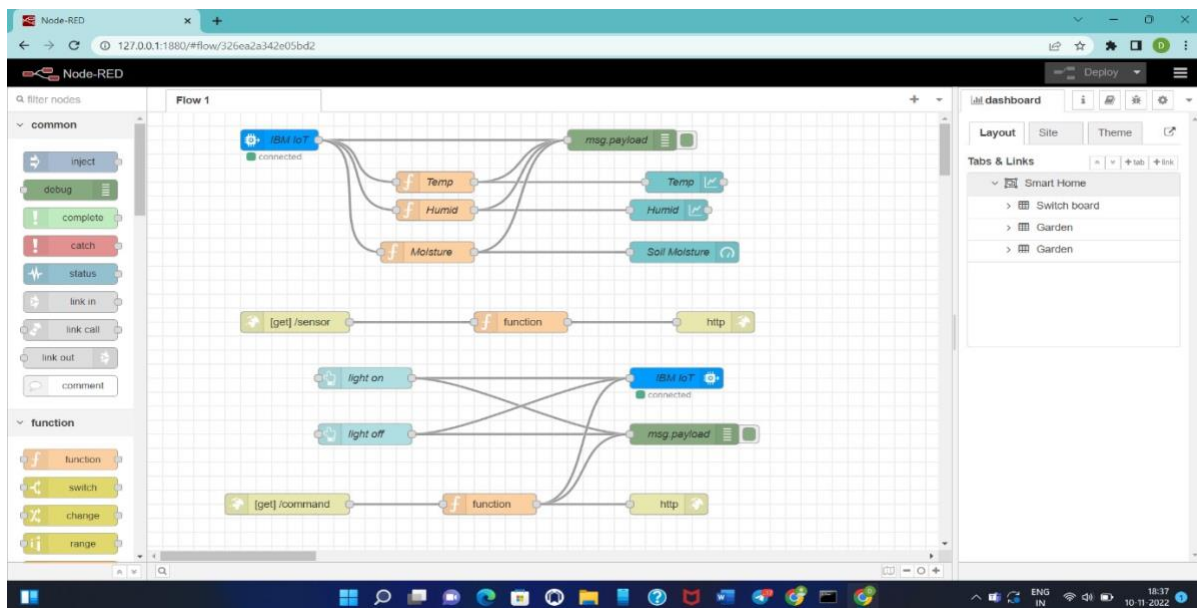
Event	Value	Format	Last Received
eventflow	{"randomNumber":98,"temp":102,"Humid":90}	json	a few seconds ago
eventflow	{"randomNumber":11,"temp":98,"Humid":78}	json	a few seconds ago
eventflow	{"randomNumber":2,"temp":99,"Humid":82}	json	a few seconds ago
eventflow	{"randomNumber":15,"temp":102,"Humid":93}	json	a few seconds ago
eventflow	{"randomNumber":34,"temp":92,"Humid":79}	json	a few seconds ago

The screenshot shows the Node-RED web interface in a browser window. The address bar displays the URL `127.0.0.1:1880/#flow/326ea2a342e05bd2`. The main workspace shows a flow with an 'IBM IoT' node connected to 'Temp' and 'Humid' nodes, which then connect to a 'function' node. The 'function' node is connected to 'light on' and 'light off' nodes, which then connect to another 'function' node. The right sidebar shows the configuration for the 'IBM IoT' node. The 'Name' is set to 'API', the 'API Key' is 'a-1x08d-p5eyyyw2eu', the 'API Token' is masked, and the 'Server Name' is 'orgid.messaging.internetofthings.ibmcloud.com'. The 'Scalable' checkbox is unchecked, and the 'Keep Alive' is set to 60 seconds. The 'Use Clean Session' checkbox is checked. The bottom status bar shows '2 nodes use this config' and 'On all flows'.

## Installing a node-red-contrib-scx-ibmiotapp and node-red dashboard

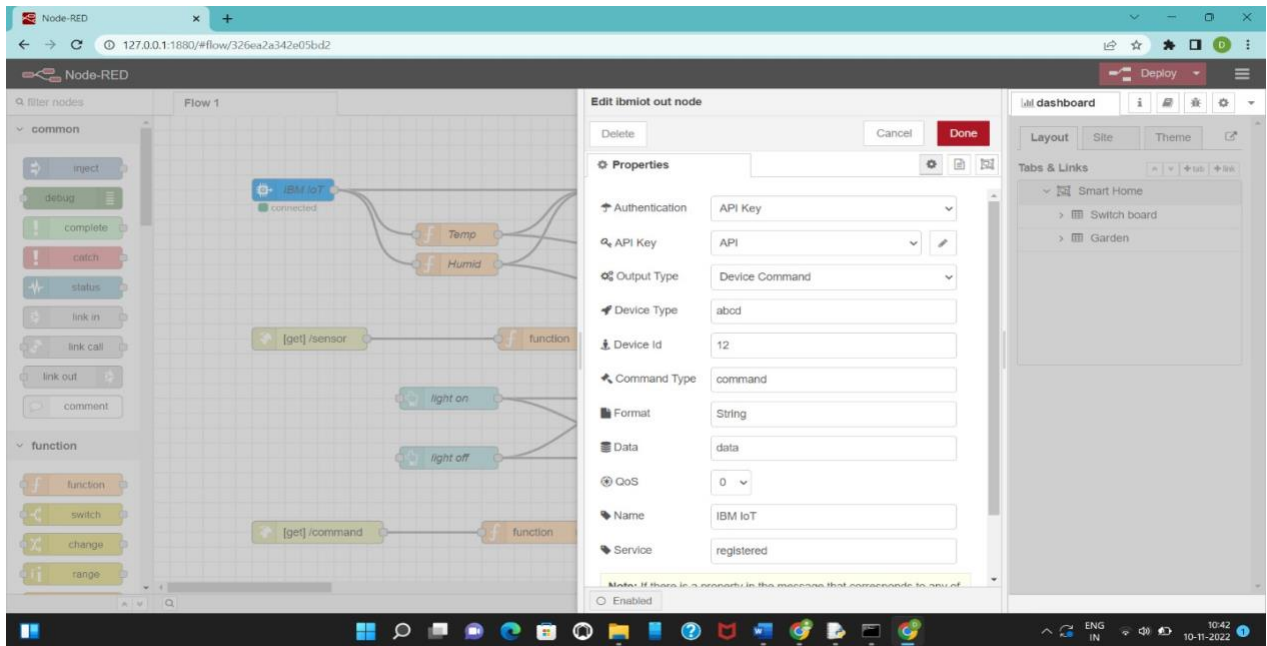


## Complete Program Flow:



## Configuration of Node-Red to collect IBM cloud data

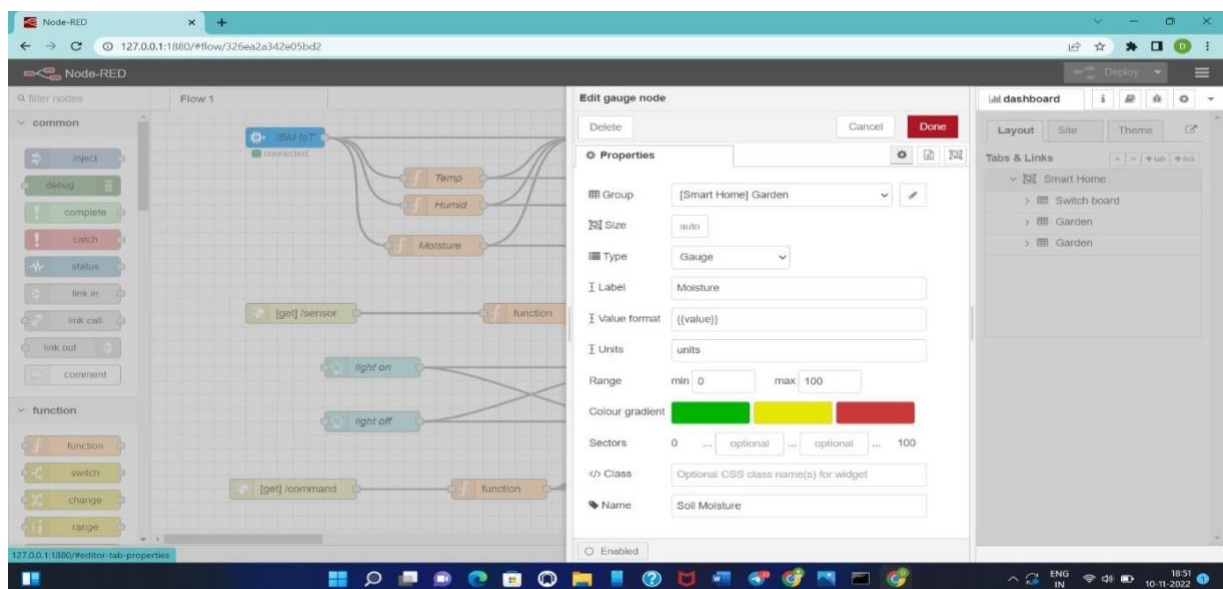
The node IBM IoT App In is added to Node-Red workflow. Then the appropriate device credentials obtained earlier are entered into the node to connect and fetch device telemetry to Node-Red.



Connect function node and The Java Script code for the function node is:

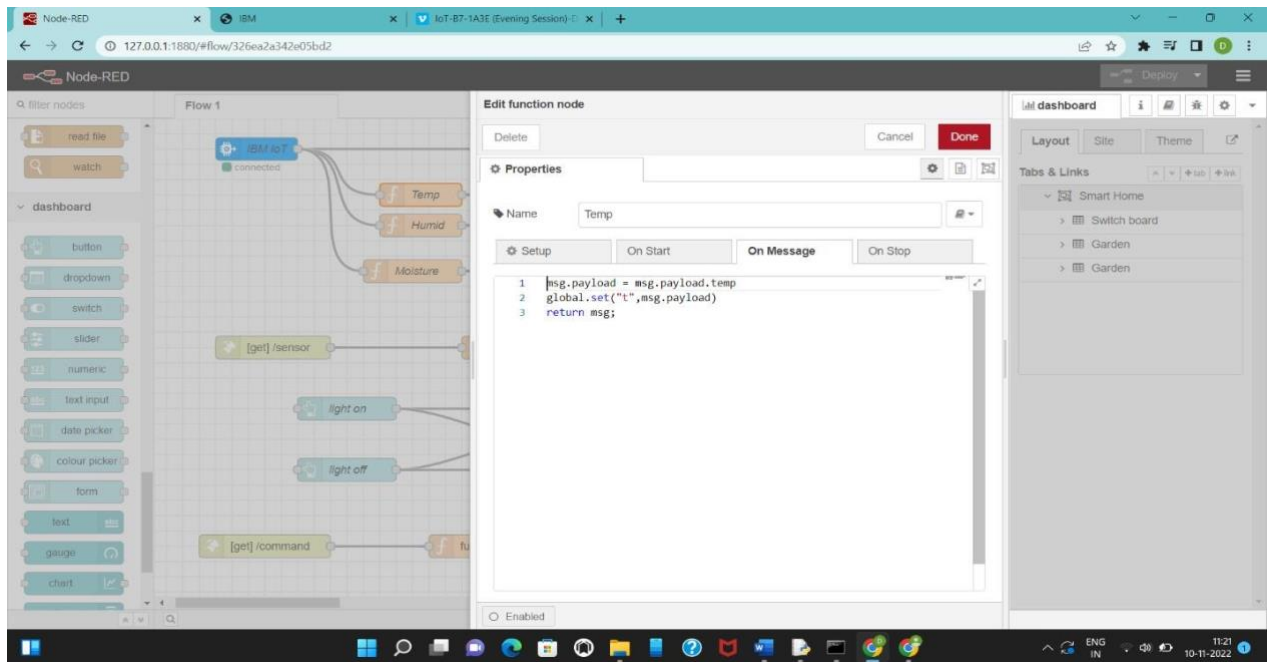
```
msg.payload=msg.payload.temp
return msg;
```

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

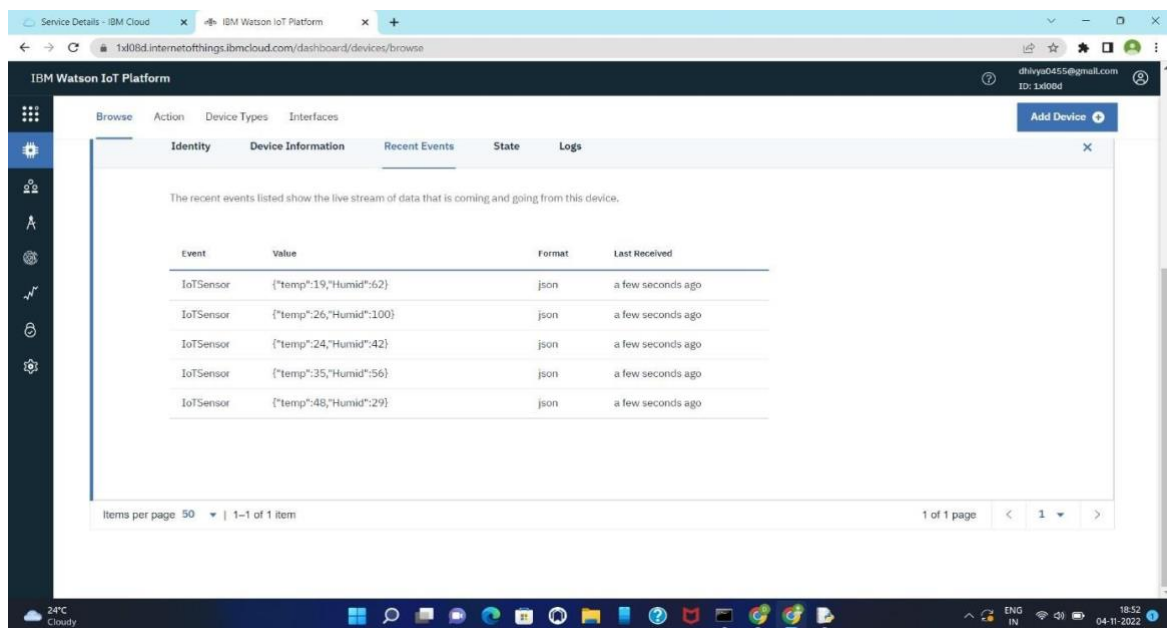


## Configuration of Node-Red to collect data from OpenWeather

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.



## Checking IoT sensor Output in IBM Watson





## Checking IoT sensor using command in Node-RED

```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (tags/v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\Sneha\AppData\Local\Programs\Python\Python37\sum.py =====
2022-11-04 21:25:30.628 IBMiotf.device.Client INFO Connected successfully
lly: dx1k100diabod112
Published Temperature = 94 C Humidity = 32 % to IBM Watson
Published Temperature = 50 C Humidity = 5 % to IBM Watson
Published Temperature = 50 C Humidity = 61 % to IBM Watson
Published Temperature = 34 C Humidity = 32 % to IBM Watson
Published Temperature = 32 C Humidity = 87 % to IBM Watson
Command received: lighton
led is on
Published Temperature = 87 C Humidity = 41 % to IBM Watson
Published Temperature = 5 C Humidity = 61 % to IBM Watson
Published Temperature = 28 C Humidity = 0 % to IBM Watson
Published Temperature = 73 C Humidity = 12 % to IBM Watson
Published Temperature = 62 C Humidity = 26 % to IBM Watson
Published Temperature = 77 C Humidity = 62 % to IBM Watson
Published Temperature = 25 C Humidity = 13 % to IBM Watson
Published Temperature = 57 C Humidity = 100 % to IBM Watson
Published Temperature = 42 C Humidity = 77 % to IBM Watson
Command received: lightoff
led is off
Published Temperature = 76 C Humidity = 13 % to IBM Watson
Published Temperature = 57 C Humidity = 21 % to IBM Watson
Published Temperature = 55 C Humidity = 57 % to IBM Watson
Published Temperature = 13 C Humidity = 47 % to IBM Watson
Published Temperature = 75 C Humidity = 60 % to IBM Watson
Published Temperature = 51 C Humidity = 73 % to IBM Watson
Published Temperature = 71 C Humidity = 34 % to IBM Watson
```

## Output in Node-RED Dashboard:

