

## **SPRINT DELIVERY – 3**

**Project Name: SMART FARMER: IoT Enabled Smart Farming  
Application**

**Team ID:PNT2022TMID42977**

### **Configuration of Node-Red to send commands to IBM cloud:**

IBM IoT out node I used to send data from Node-Red to IBM Watson device.

So, after adding it to the flow we need to configure it with credentials of our Watson device.

Here we add two buttons in UI

Edit ibmiot in node

Delete

Cancel

Done

⚙ Properties

⚙

📄

🖨

🔑 Authentication

API Key

▼

🔑 API Key

IBMIOT APIKEY

▼

✎

⚙ Input Type

Device Event

▼

📡 Device Type

☐ All or

abcd

👤 Device Id

☐ All or

7654321

📋 Event

☒ All or

+

📄 Format

☐ All or

json

📶 QoS

0

▼

🏠 Name

IBM IoT

☐ Enabled

1 -> for motor

on2 -> for motor

off

We used a function node to analyse the data received and assign command to each number. The Java script code for the analysis is:

```
If (msg. payload===1)
```

```
msg. payload= {"command" :
```

```
" ON"}; else if (msg. payload===0)
```

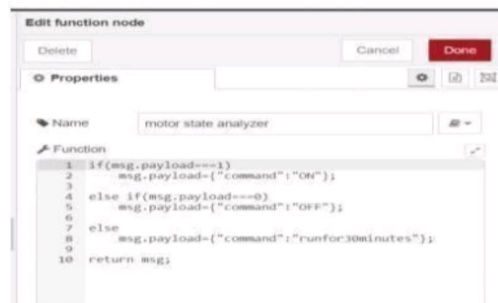
```
msg. payload= {"command" :
```

```
" OFF"};
```

Then we use another function node to parse the data and get the command and represent it visually with text node.

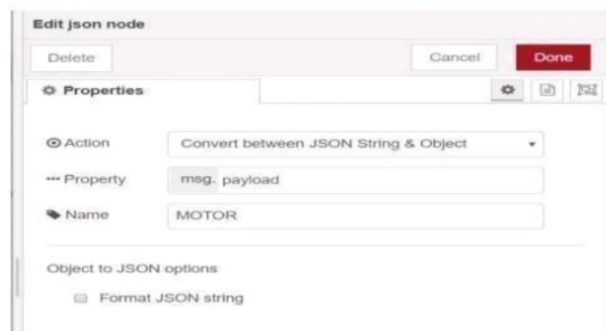
The Java script code for that function node is:

```
var state=msg.payload;  
msg.payload = state.command;  
return msg;
```

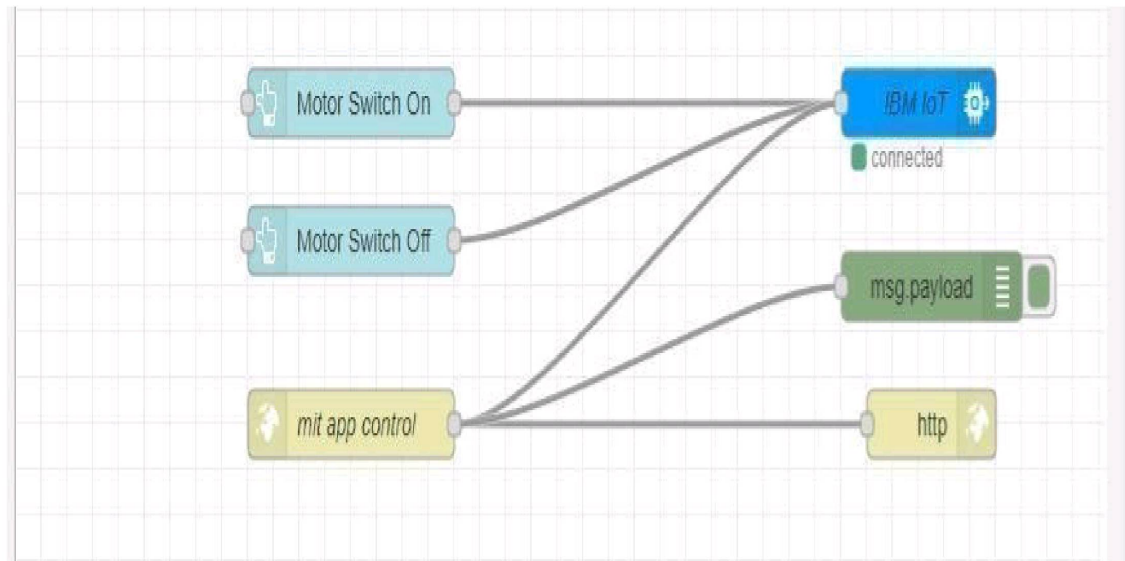


The above images show the java script codes of analyser and state function nodes.

Then we add edit Json node to the conversion between JSON string & object and finally connect it to IBM IoT Out.



Edit JSON node needs to be configured like this



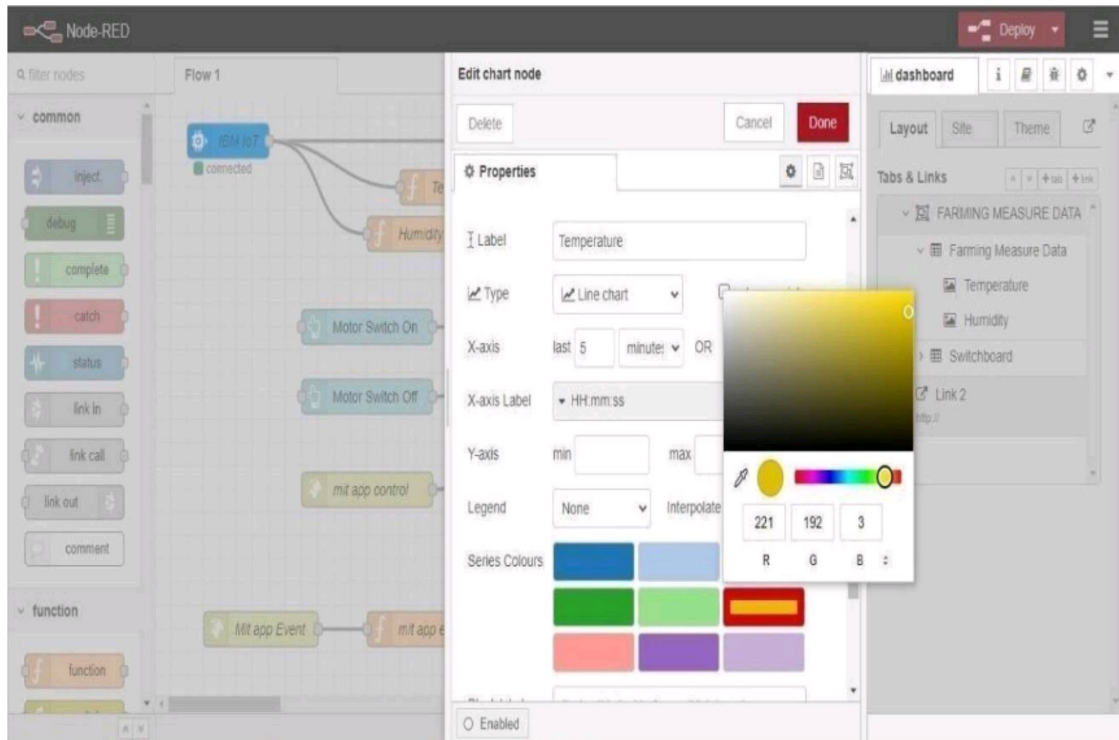
This is the program flow for sending commands to IBM cloud.

## Adjusting User Interface:

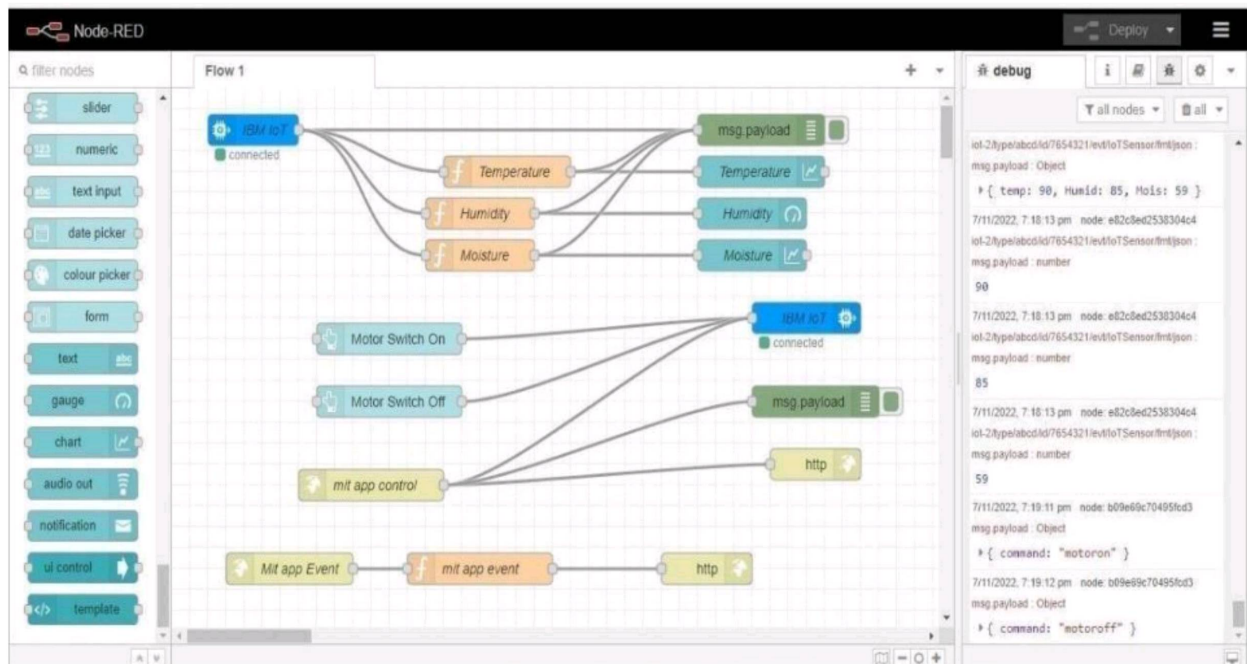
In order to display the parsed JSON data a Node-Red dashboard is created

Here we are using Gauges, text and button nodes to display in the UI and helps to monitor the parameters and control the farm equipment.

Below images are the Gauge, text and button node configurations.

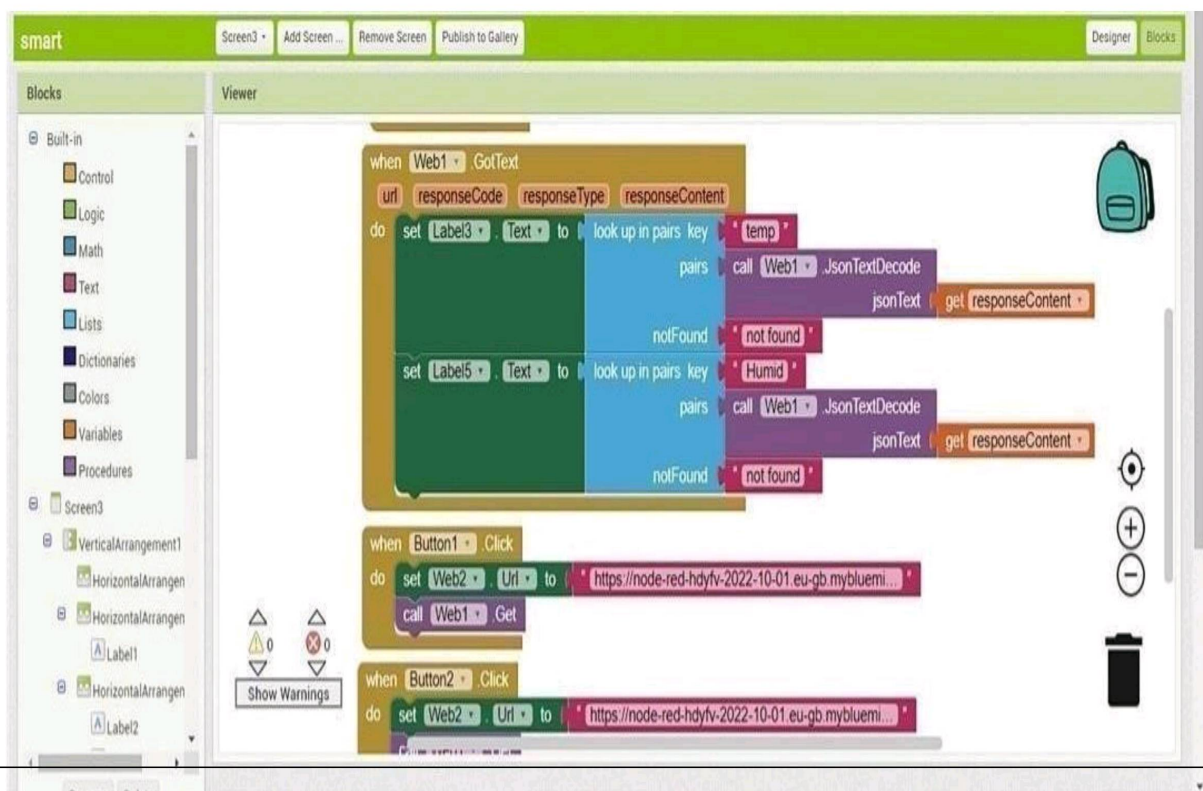


## Complete Program Flow:

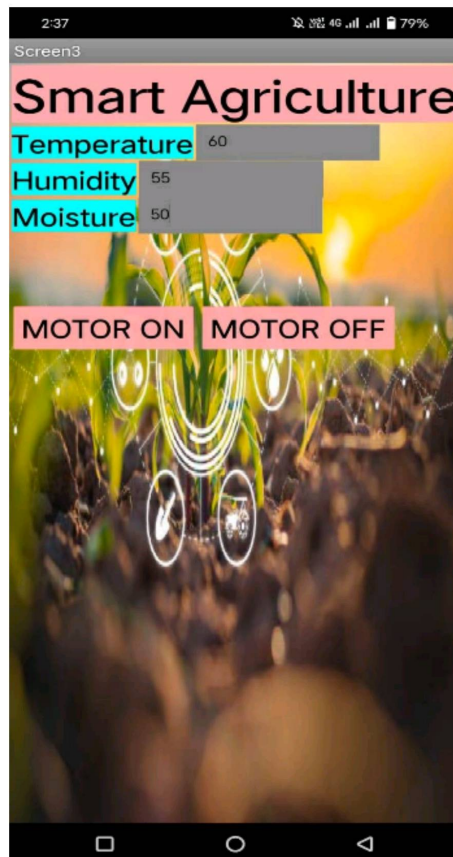
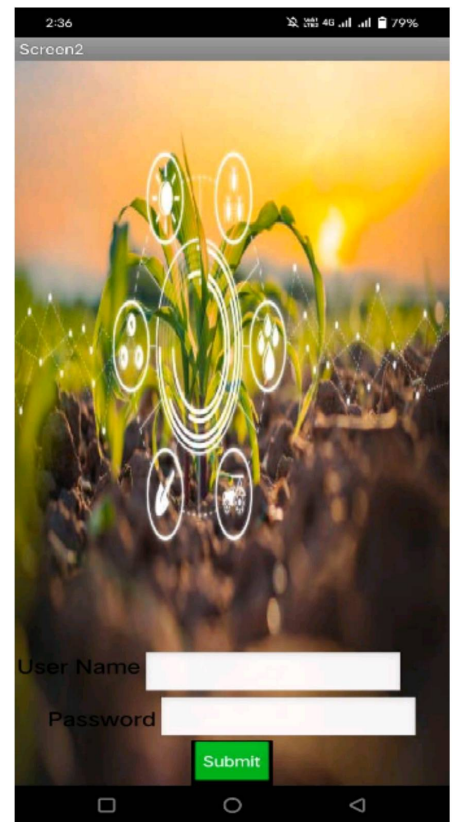


## MOBILE APP WEB:

## BLOCK DIAGRAM:









## Web APP UI Home Tab:

