

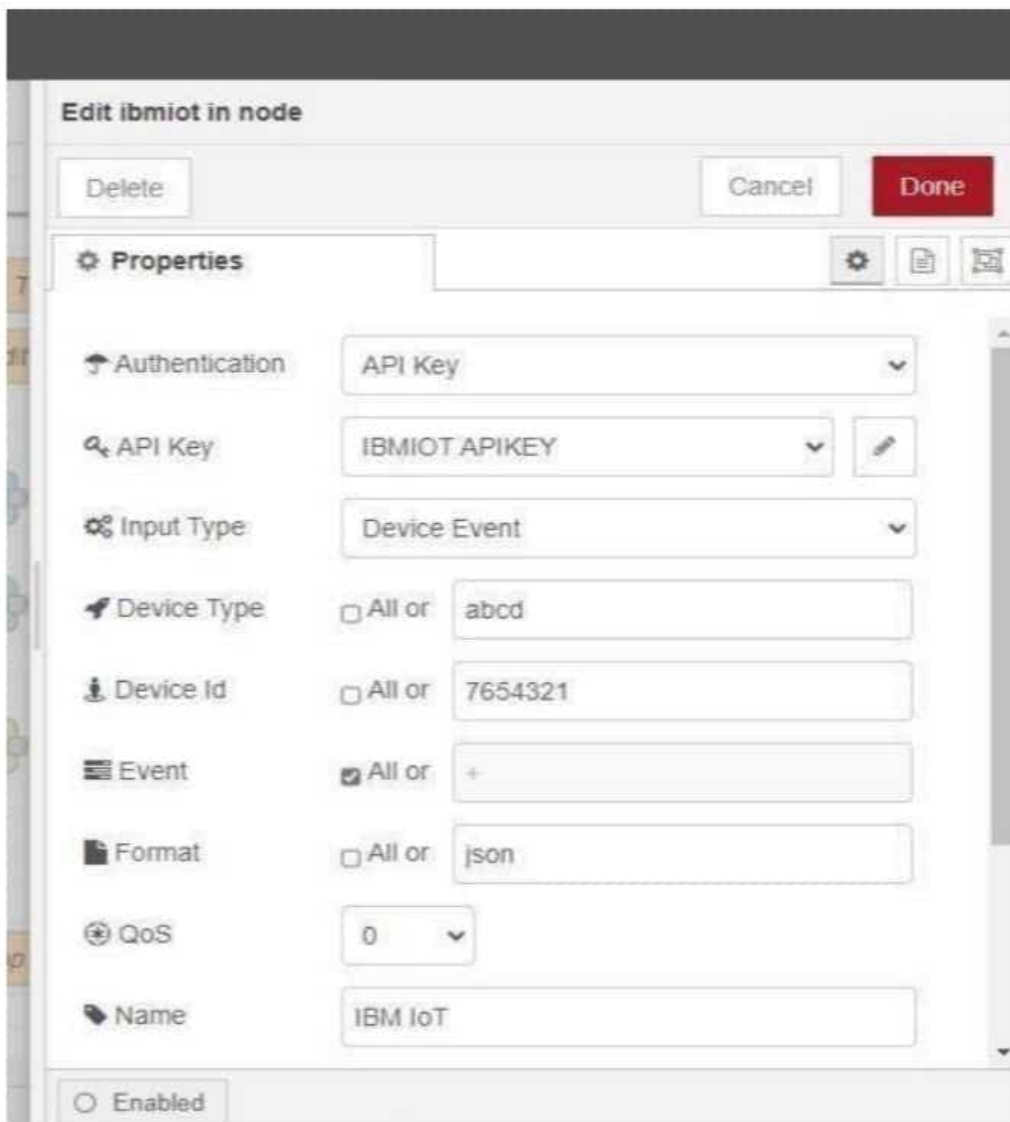
## **SPRINT DELIVERY – 3**

<b>Team ID</b>	PNT2022TMID43969
<b>Project Name</b>	IoT Enabled Smart Farming Application
<b>Date</b>	15 November 2022

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

ibmiot 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



The screenshot shows the 'Edit ibmiot in node' configuration window. At the top, there are three buttons: 'Delete', 'Cancel', and 'Done'. Below these is a 'Properties' section with a gear icon and a search icon. The configuration fields are as follows:

- Authentication:** A dropdown menu set to 'API Key'.
- API Key:** A text input field containing 'IBMIOT-APIKEY' with a search icon to its left and a pencil icon to its right.
- Input Type:** A dropdown menu set to 'Device Event'.
- Device Type:** A checkbox labeled 'All or' followed by a text input field containing 'abcd'.
- Device Id:** A checkbox labeled 'All or' followed by a text input field containing '7654321'.
- Event:** A checkbox labeled 'All or' followed by a text input field containing '+'.
- Format:** A checkbox labeled 'All or' followed by a text input field containing 'json'.
- QoS:** A dropdown menu set to '0'.
- Name:** A text input field containing 'IBM IoT'.

At the bottom left, there is a checkbox labeled 'Enabled' which is currently checked.

1 -> for motor on

2 -> 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 analyse 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;
```

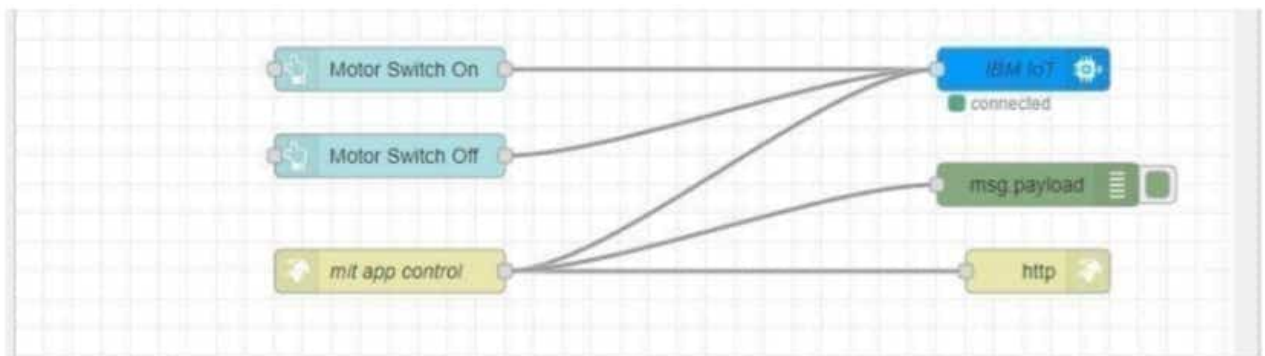


The above images show the java script codes of analyser and state function nodes.

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.



smart Screen3 Add Screen Remove Screen Publish to Gallery Designer Blocks

Blocks

- Built-in
  - Control
  - Logic
  - Math
  - Text
  - Lists
  - Dictionaries
  - Colors
  - Variables
  - Procedures
- Screen3
  - VerticalAnordnung1
  - HorizontalAnzeigen
    - Label1
    - HorizontalAnzeigen
      - Label2

Viewer

when Web1 -> GotText

url responseCode responseType responseContent

do

set Label3 -> Text -> to look up in pairs key pairs

call Web1 -> JsonTextDecode jsonText get responseContent ->

not found not found

set Label5 -> Text -> to look up in pairs key pairs

call Web1 -> JsonTextDecode jsonText get responseContent ->

not found not found

when Button1 -> Click

do

set Web2 -> Url -> to https://node-red-hdyfz-2022-10-01-eu-gb.mybluemix.net

call Web1 -> Get

when Button2 -> Click

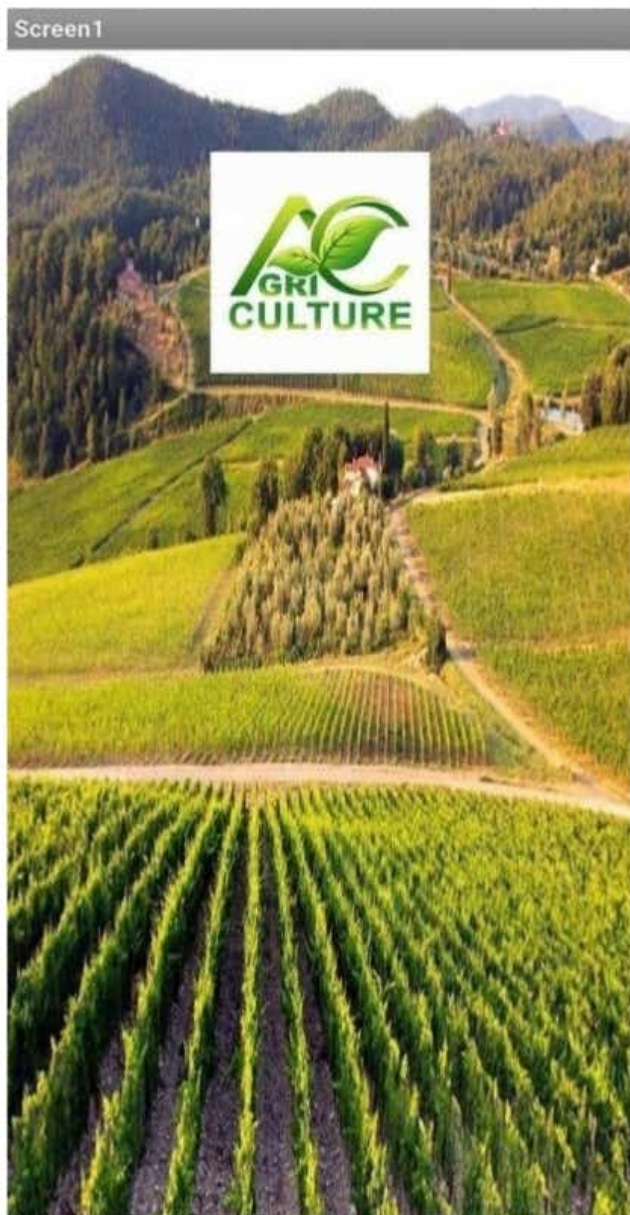
do

set Web2 -> Url -> to https://node-red-hdyfz-2022-10-01-eu-gb.mybluemix.net

Show Warnings



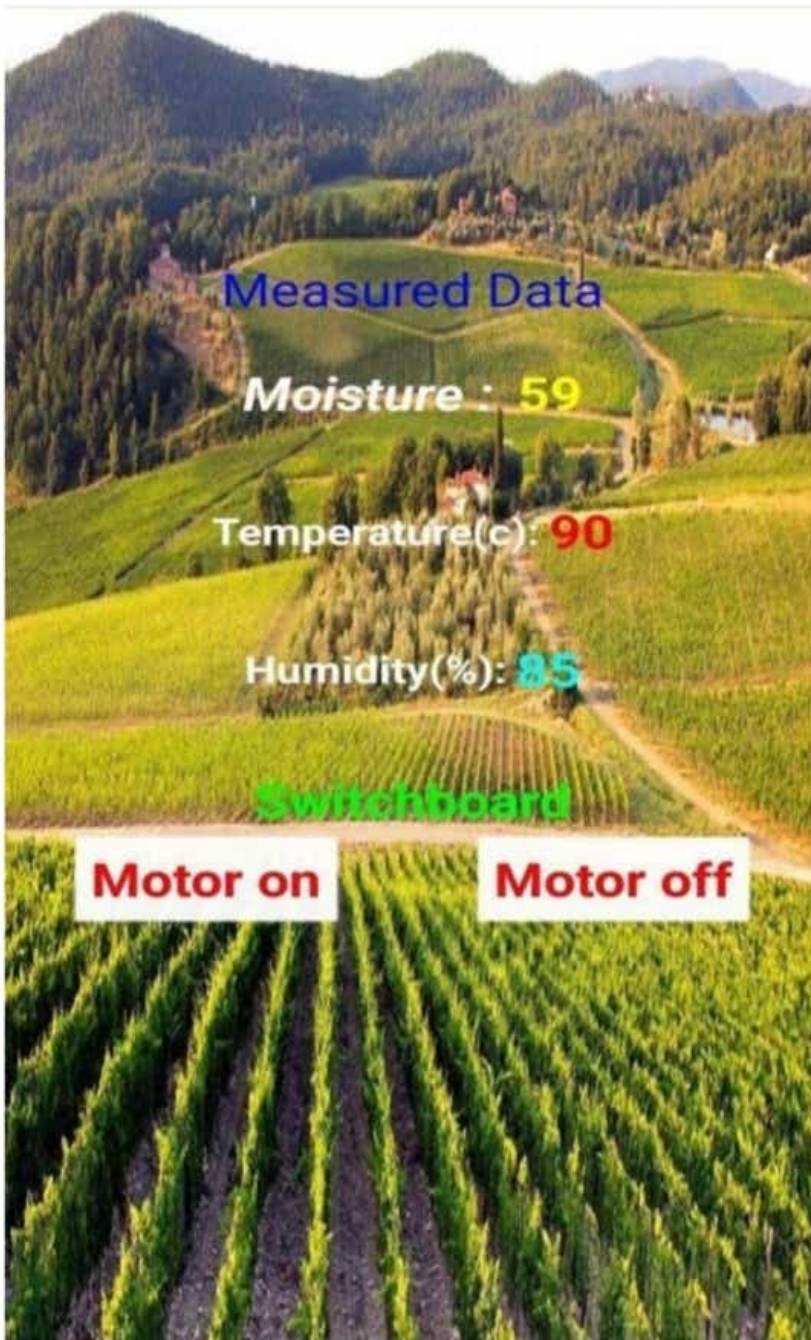
Screen1



SCREEN – 1



SCREEN - 2



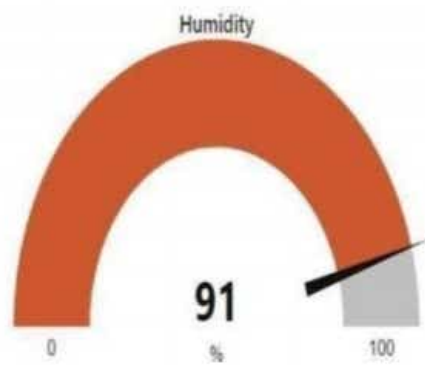
### SCREEN - 3

Web APP UI Home Tab



## FARMING MEASURE DATA

### Farming Measure Data



### Switchboard

MOTOR SWITCH ON

MOTOR SWITCH OFF