

IBM NALAIYATHIRAN SMART FARMER-IOT ENABLED SMART FARMING APPLICATION

SPRINT 3

Title	Smart farmer-IoT enabled smart farming application
Domain	Internet of Things
Team ID	PNT2022TMID44170
Project Name	Project – Smart Farmer-IoT Enabled smartFarming Application

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.

The screenshot displays the Node-RED web interface in a browser. The main workspace shows a flow named 'Flow 1' with several nodes: 'motor switch on', 'motor switch off', 'msg payload', 'IBM IoT', 'mit app control', 'Moisture', 'temperature', and 'humidity'. The 'IBM IoT' node is highlighted, and its configuration panel is open on the right. The configuration panel includes fields for 'Authentication' (API Key), 'API Key' (hasna09), 'Input Type' (Device Event), 'Device Type' (All or hasnarahah09), 'Device Id' (All or hasna09), 'Event' (All or +), 'Format' (All or json), 'QoS' (0), 'Name' (IBM IoT), and 'Service' (registered). A yellow warning box at the bottom of the configuration panel states: 'Use the Input Type property to configure this node to receive Events sent by IoT Devices, Commands sent to IoT Devices, Status Messages referring to IoT Devices, or Status Messages referring to IoT Applications. Check the info tab, to get more information about each of the fields'. The bottom of the screen shows a Windows taskbar with the date and time as 10:51 AM on 16-Nov-22.

We used a function node to analyse the data received and assign command to each number.

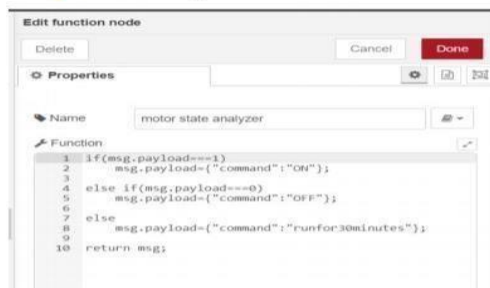
The Java script code for the analysis is:

```
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.

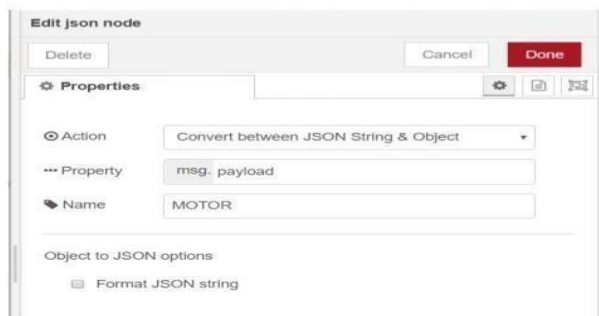
The Java script code for that function node is:

```
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.

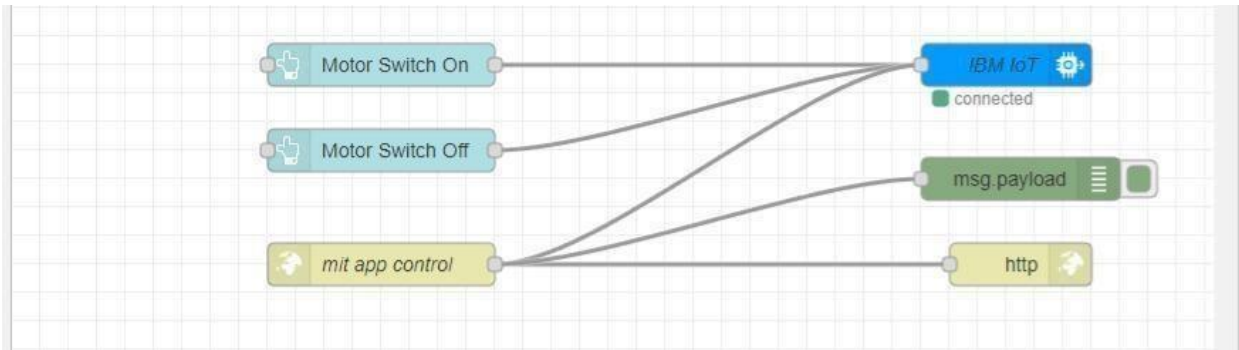


Edit JSON node needs to be configured like this

Here we add two buttons in UI

- for motor on

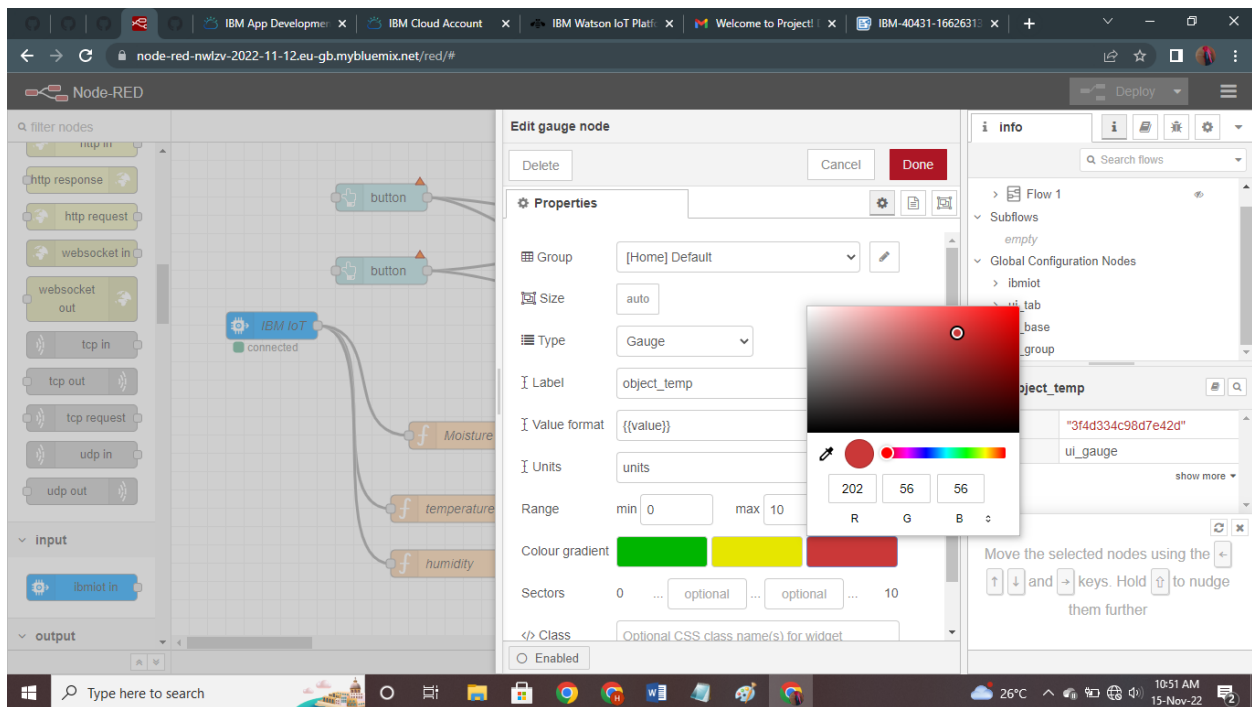
- for motor off

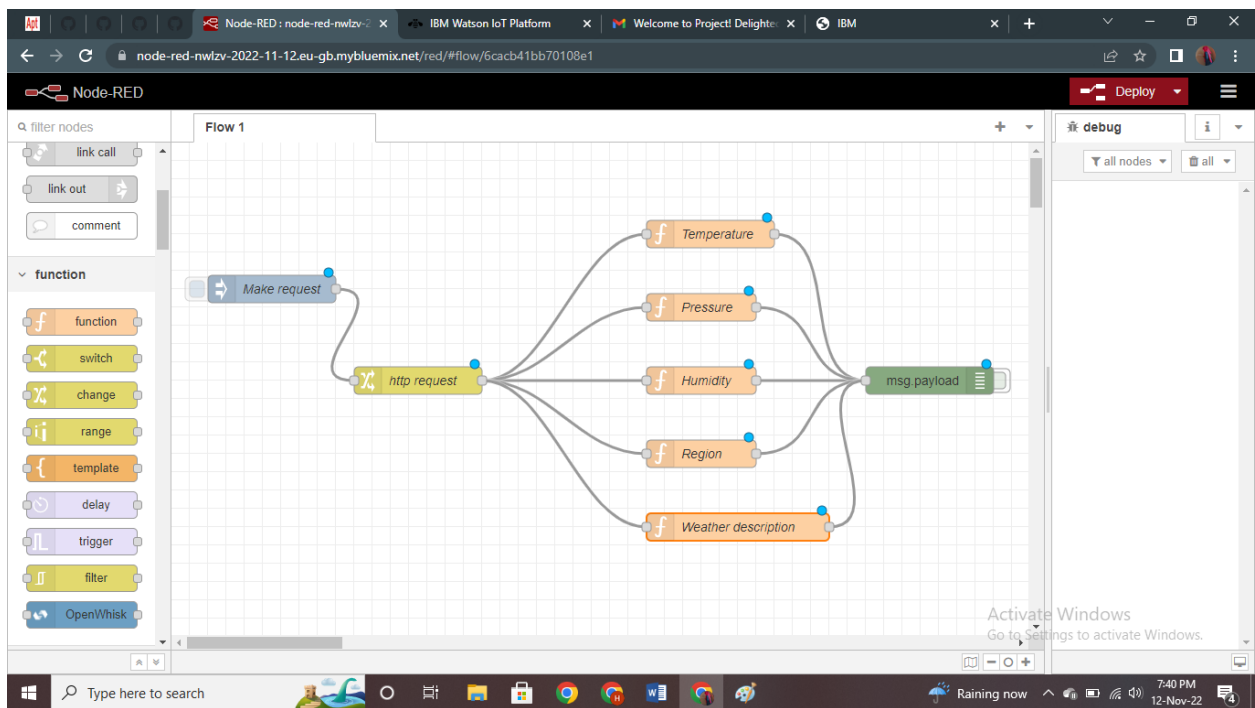


This is the program flow for sending commands to IBM cloud.

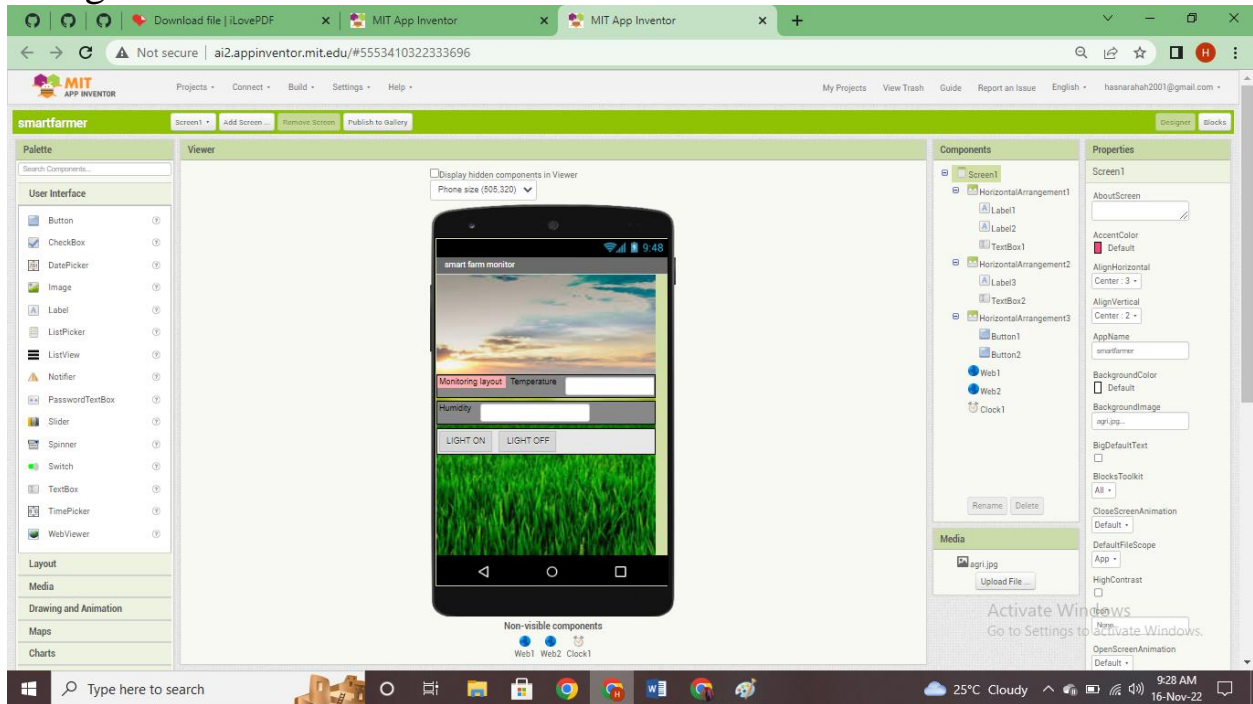
Adjusting User Interface

- 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.

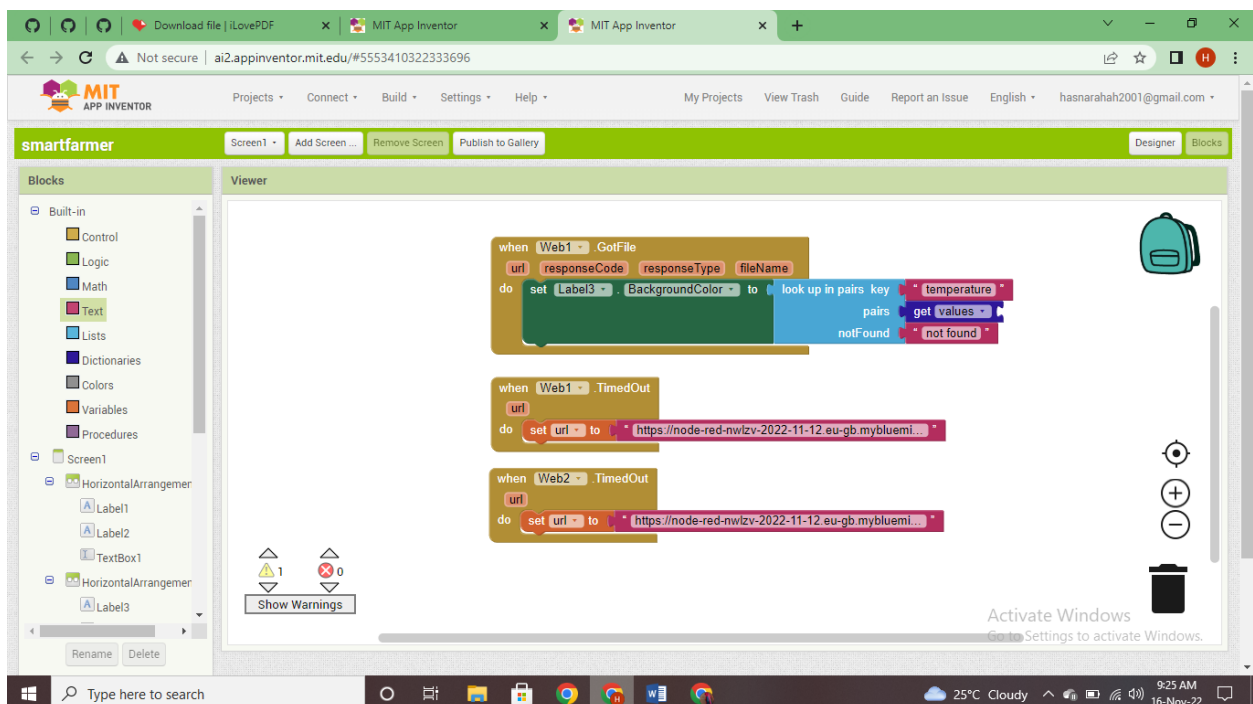




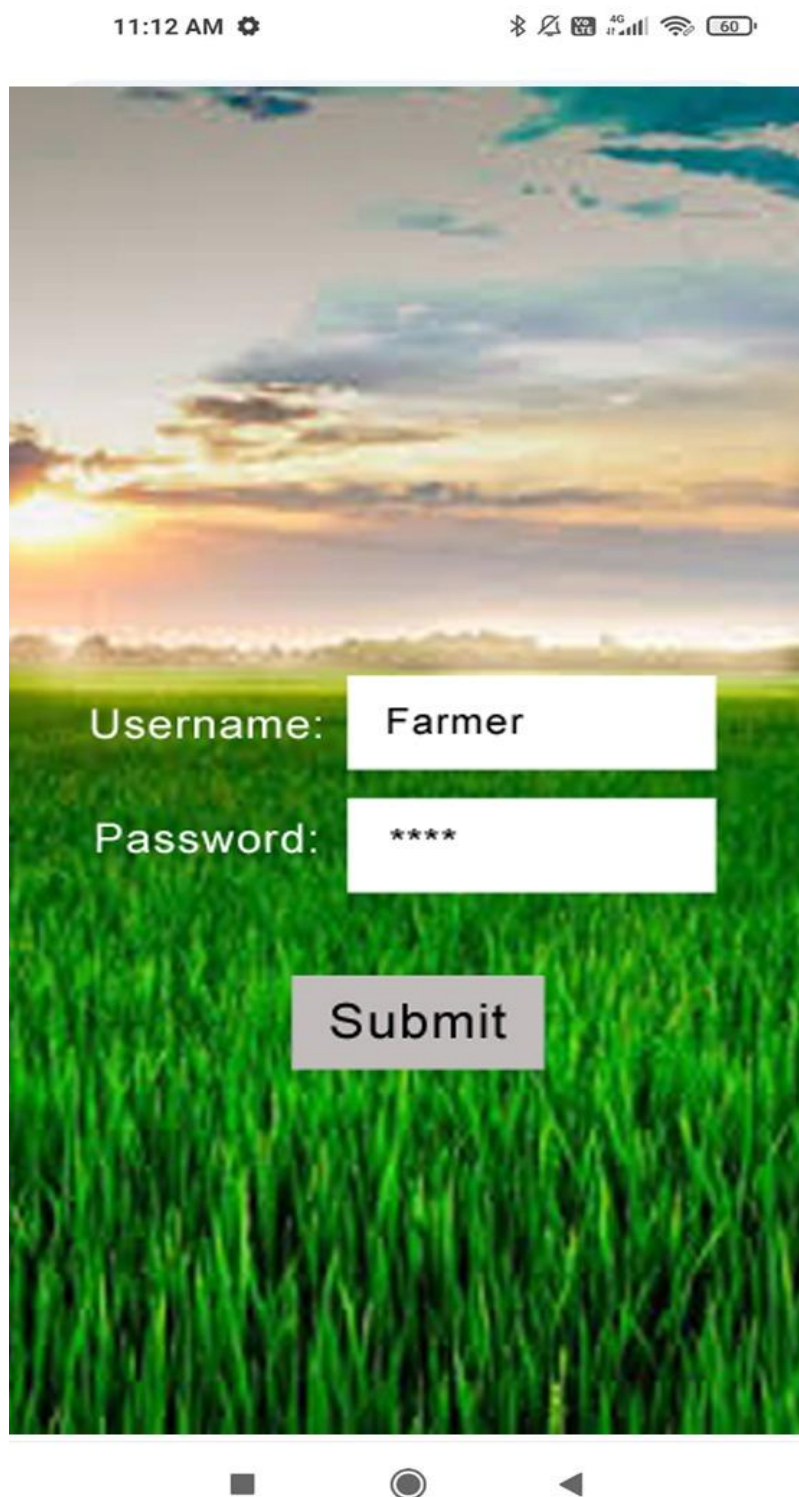
MIT APP INVENTOR: Designer screen



Blocks screen:



DEVELOPED APP MOBILE SCREEN SCREEN1



SCREEN2

11:12 AM ⚙



Measured Data

Moisture : 59

Temperature(c) : 90

Humidity (%) : 85

Switchboard

Motor on

Motor off

