

## **SPRINT DELIVERY – 3**

<b>Team ID</b>	PNT2022TMID46961
<b>Project Name</b>	IoT Enabled Smart Farming Application
<b>Date</b>	18 November 2022

## 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

The screenshot displays the Node-RED web interface in a browser. The main workspace shows a flow titled 'Smart Farming' with several nodes: 'inject', 'debug', 'complete', 'catch', 'status', 'link in', 'link call', 'link out', 'comment', 'function', 'switch', 'change', 'range', 'Motor on', 'Motor off', '[get] /sensor', '[get] /control', 'httpfunctionnode', 'http request', 'msg.payload', 'Moisture', 'Humidity', 'Temperature', and 'IBM IoT'. The 'IBM IoT' node is highlighted, and its configuration panel is open on the right. The configuration panel includes fields for API Key, Input Type, Device Type, Device Id, Event, Format, QoS, Name, and Service. The 'Input Type' is set to 'Device Event', 'Device Type' is 'All or +', 'Device Id' is 'device id e.g. ab12cd231a21', 'Event' is 'All or +', 'Format' is 'json', 'QoS' is '0', 'Name' is 'IBM IoT', and 'Service' is 'registered'. A yellow tooltip message is visible at the bottom of the configuration panel, stating: '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 right sidebar shows the 'info' tab with a search bar and a list of flows: 'Smart Farming', 'Flow 1', 'Subflows', and 'Global Configuration Nodes'. The bottom status bar shows the time as 10:50 am on 18/11/2022.

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 analyses is:

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

```
msg.payload={"command": "ON"};
```

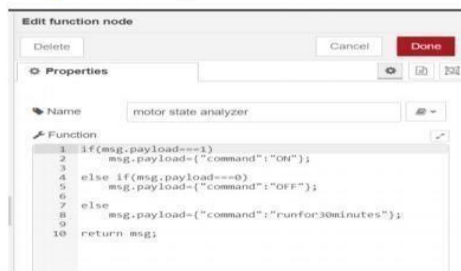
```
else if(msg.payload===0)
```

```
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.

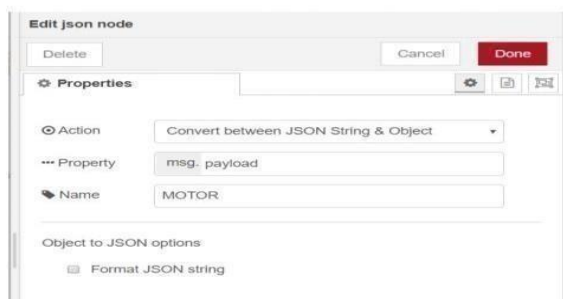
The Java script code for that function node is:

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

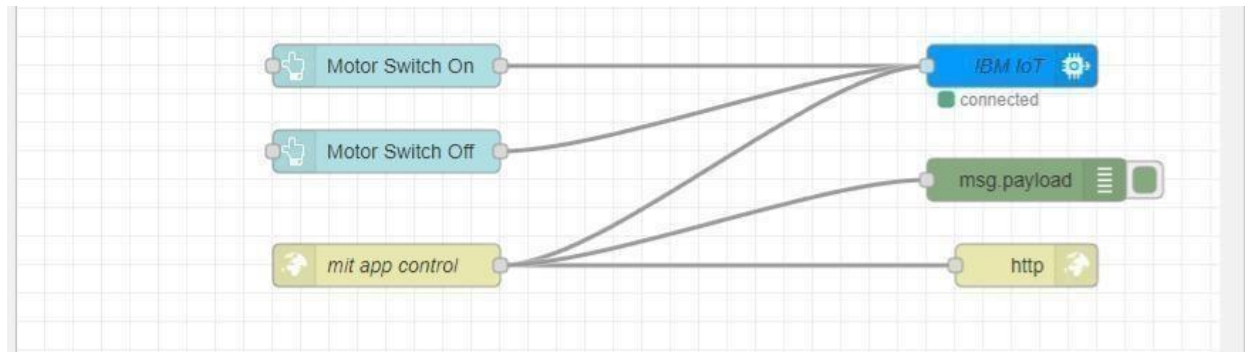


The above images show the java script codes of analyser and state function nodes.

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



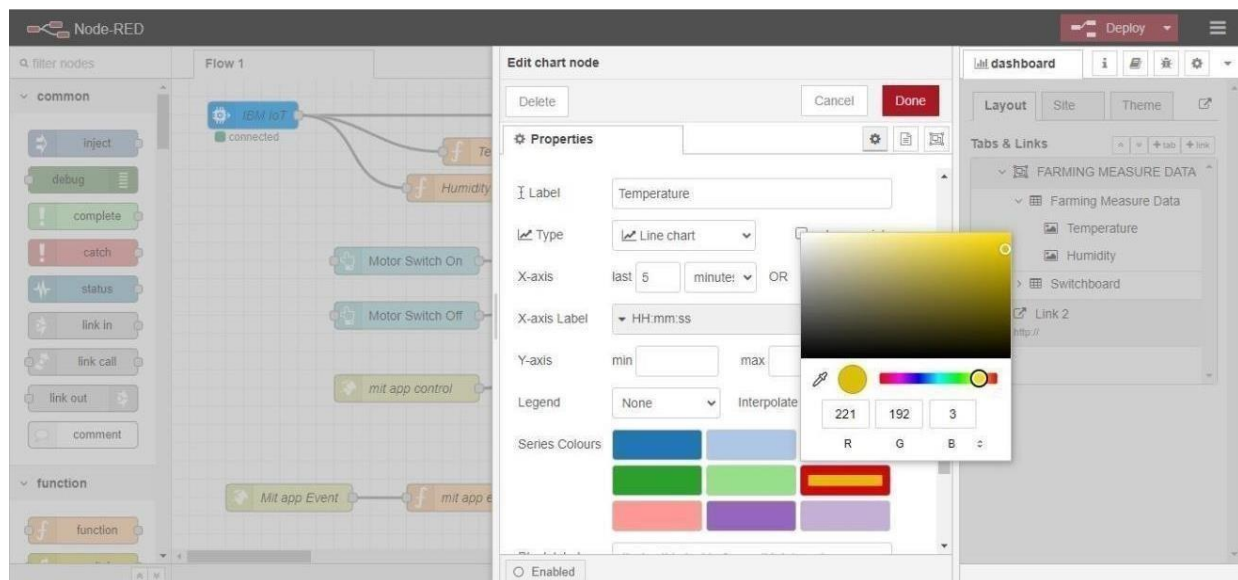
This is the program flow for sending commands to IBM cloud.

## Adjusting User Interface

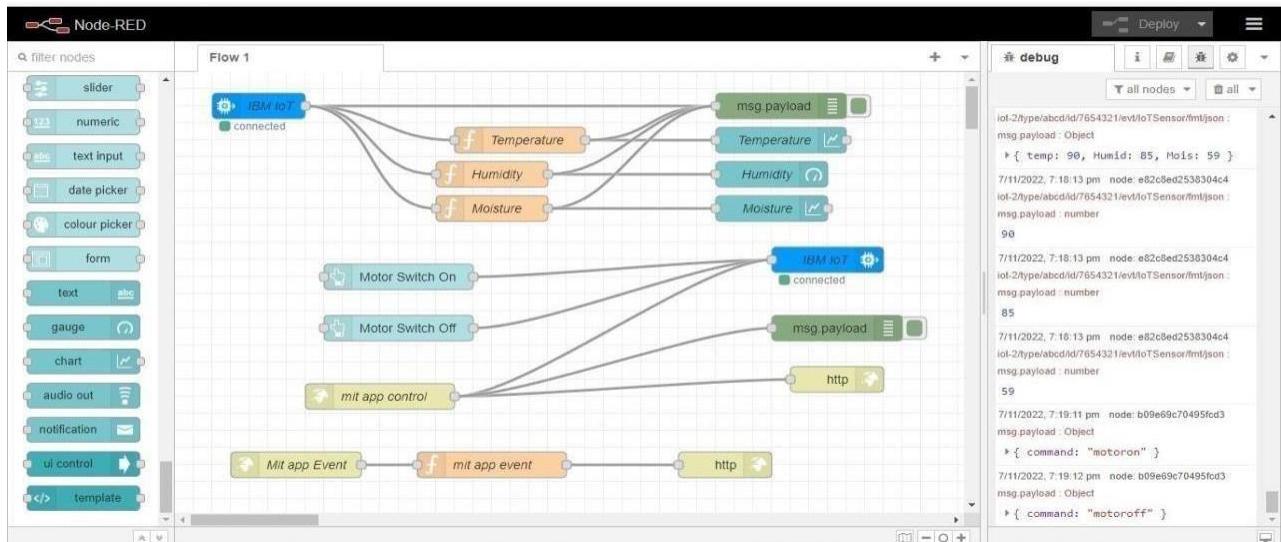
In order to display the parsed JSON data a Node-Red dashboard is created

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.

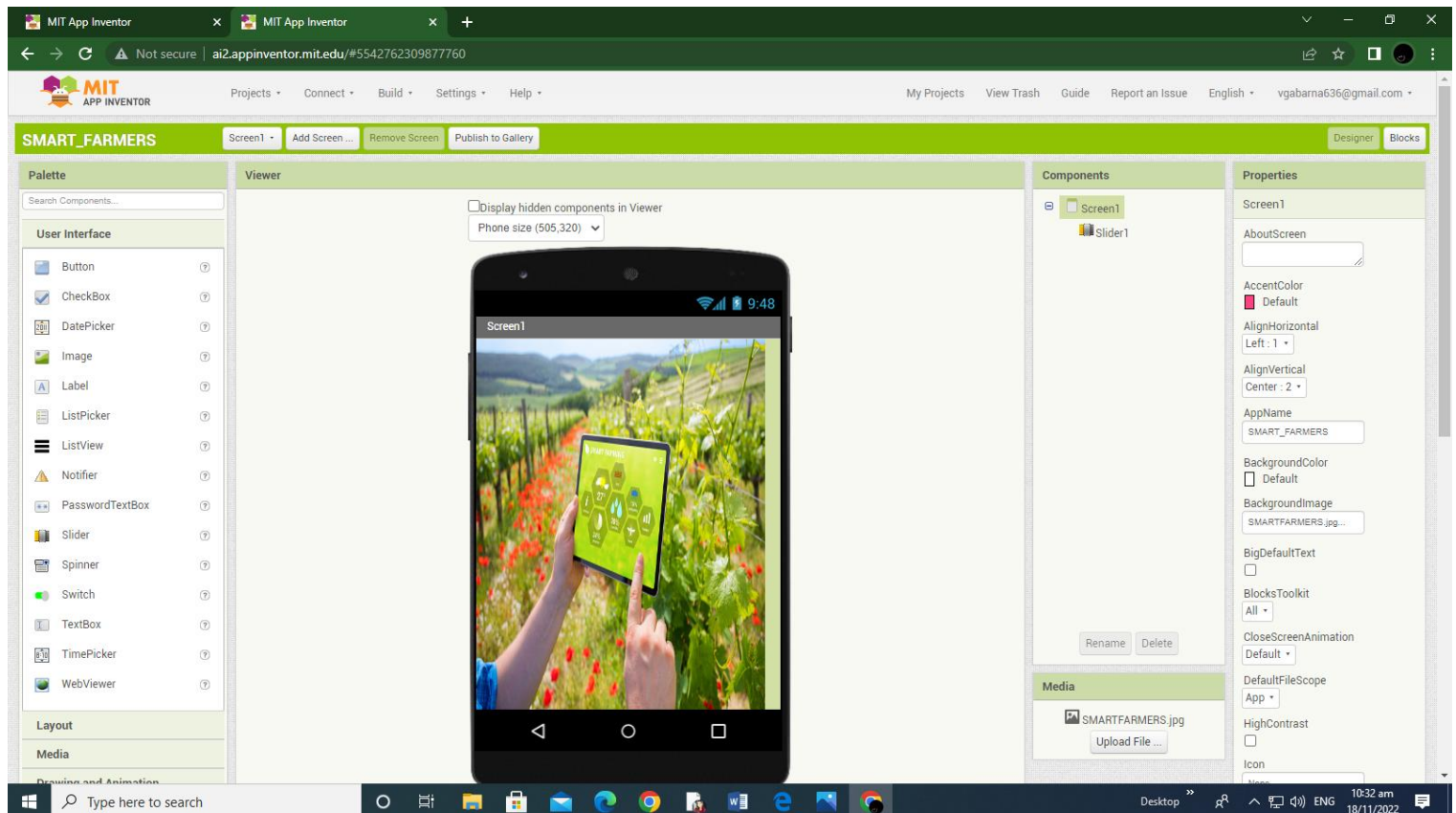


## Complete Program Flow

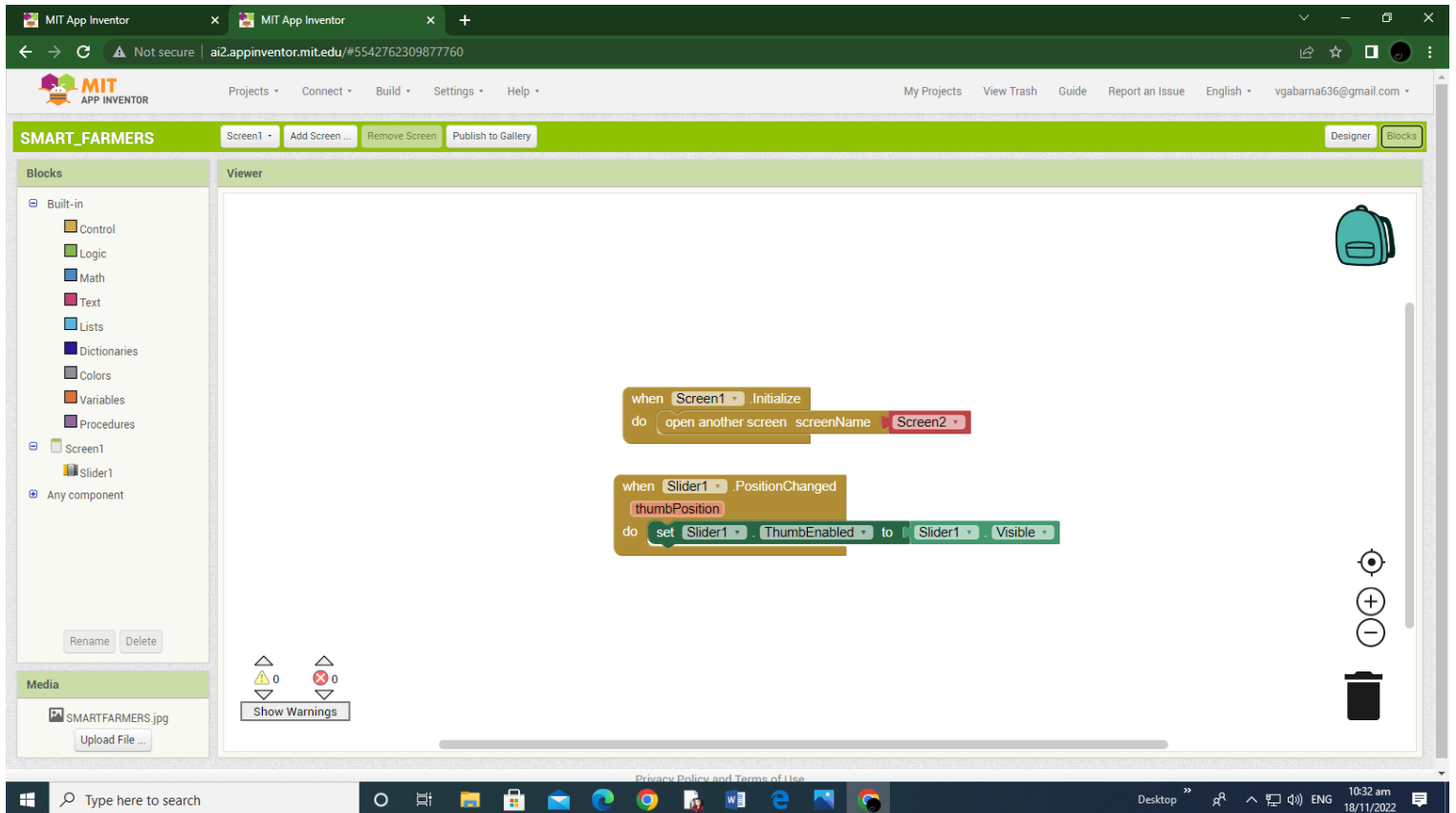


## MOBILE APP WEB

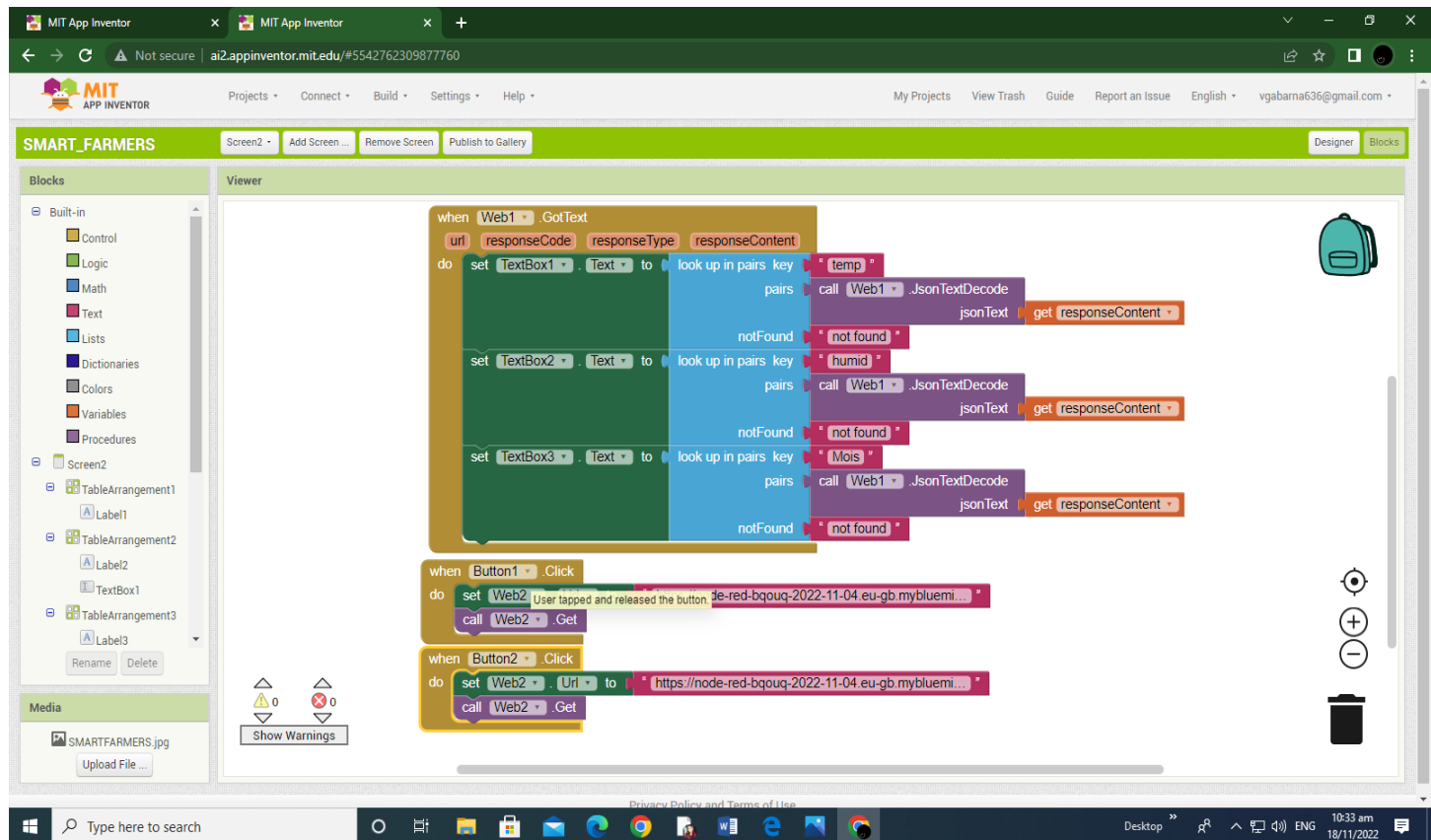
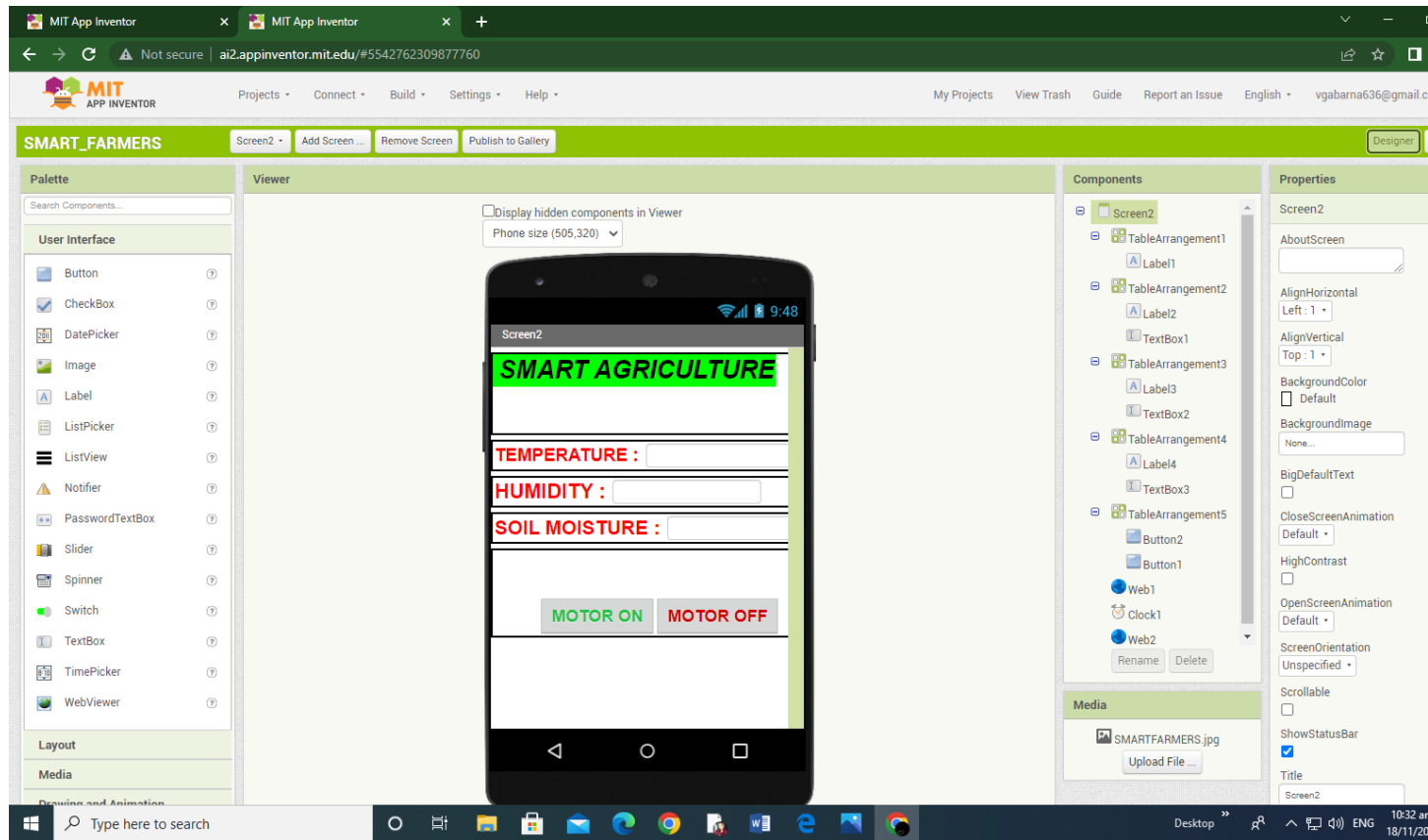
### DESIGN SCREEN 1:



# BLOCK DIAGRAM :



## DESIGN SCREEN 2 :





4G 1.5K/s

10:58 AM

4G 87%

Screen2

# SMART AGRICULTURE

TEMPERATURE :

77

HUMIDITY :

56

SOIL MOISTURE :

30

MOTOR ON

MOTOR OFF



WEB USER INTERFACE :

