

SPRINT-3

Team ID	PNT2022TMID19466
Project Title	Smart Farmer – IoT enabled Smart Farming Application
Date	16-11-2022

Smart Farming Application using NODE RED SERVICE:

IBM IOT IN NODE:

The screenshot displays the Node-RED web interface in a browser. The main workspace shows a flow diagram with an 'IBM IoT' node connected to three function nodes labeled 'Soil Moisture', 'Humidity', and 'Temperature'. These function nodes are connected to an 'http request' node, which is then connected to an 'inject' node. The left sidebar contains a palette of nodes categorized under 'common' and 'function'. The right sidebar shows the 'Edit ibmiot in node' configuration panel. The configuration includes fields for 'Authentication' (API Key), 'API Key' (2c533d245eb7b902), 'Input Type' (Device Event), 'Device Type' (All or +), 'Device Id' (0223), 'Event' (All or +), 'Format' (All or json), 'QoS' (0), 'Name' (IBM IoT), and 'Service' (registered). A yellow tooltip at the bottom of the configuration panel reads: 'Use the Input Type property to configure this node to receive Events sent by IoT Devices. Commands sent to IoT Devices. Status...'. The bottom status bar shows the system time as 12:56 PM on 16-11-2022.

SOIL MOISTURE FUNCTION:

The screenshot shows the Node-RED web interface in a browser. The main workspace displays a flow named 'Flow 1' with an 'IBM IoT' node connected to three function nodes labeled 'Soil Moisture', 'Humidity', and 'Temperature'. These function nodes are connected to a 'switch' node, which then connects to an 'http request' node. The 'Edit function node' panel for the 'Soil Moisture' node is open, showing the 'On Message' tab with the following JavaScript code:

```
1 global.set('moist',msg.payload.soil_moisture)
2 msg.payload=msg.payload.soil_moisture
3 return msg;
```

The 'Properties' section shows the node name as 'Soil Moisture'. The 'help' panel on the right provides documentation for the 'function' node, explaining that it runs a JavaScript function against incoming messages and returns a new message object.

SOIL MOISTURE GAUGE:

The screenshot shows the same Node-RED interface as above, but with the 'Edit gauge node' panel open for the 'Soil Moisture' node. The 'Properties' section is configured as follows:

- Group: [Home] Soil Moisture
- Size: auto
- Type: Gauge
- Label: Soil Moisture
- Value format: {{value}}
- Units: units
- Range: min 0 max 100
- Colour gradient: A gradient bar from green to yellow to red.
- Sectors: 0 ... optional ... optional ... 100
- Class: Optional CSS class name(s) for widget
- Name: (empty)

The 'help' panel on the right provides documentation for the 'gauge' node, explaining that it adds a gauge widget to the user interface and formats the value based on the 'Value Format' property.

HUMIDITY FUNCTION:

The screenshot shows the Node-RED web interface in a browser. The main workspace displays a flow with an 'IBM IoT' node connected to a 'switch' node, which then branches into three function nodes: 'Soil Moisture', 'Humidity', and 'Temperature'. The 'Humidity' node is selected, and its configuration panel is open. The 'Name' is set to 'Humidity'. The 'On Message' tab is active, showing a JavaScript function:

```
1 global.set('hum',msg.payload.humidity)
2 msg.payload=msg.payload.humidity
3 return msg;
```

 The right sidebar shows a list of nodes, with 'function' selected. The bottom status bar indicates the system is 'Enabled' and shows the time as 12:47 PM on 16-11-2022.

HUMIDITY CHART:

The screenshot shows the Node-RED web interface with the same flow as the previous image. The 'Humidity' node is now a 'chart' node. The configuration panel for the 'chart' node is open. The 'Group' is '[Home] Soil Moisture', 'Size' is 'auto', and 'Label' is 'Humidity'. The 'Type' is 'Line chart'. The 'X-axis' is set to 'last 1 hours' or '1000 points'. The 'X-axis Label' is 'HH:mm:ss'. The 'Y-axis' has 'min' and 'max' fields. The 'Legend' is 'None'. The 'Series Colours' are set to a grid of colors. The right sidebar shows a list of nodes, with 'chart' selected. The bottom status bar indicates the system is 'Enabled' and shows the time as 12:48 PM on 16-11-2022.

TEMPERATURE FUNCTION:

The screenshot shows the Node-RED web interface in a browser. The main workspace displays a flow named 'Flow 1' with nodes: 'inject', 'debug', 'complete', 'catch', 'status', 'link in', 'link call', 'link out', 'comment', 'function', 'switch', 'http request', 'data', and 'range'. The 'function' node is selected, and the 'Edit function node' dialog is open. The 'Name' field is set to 'Temperature'. The 'On Message' tab is active, showing the following JavaScript code:

```
1 global.set("temp",msg.payload.temperature)
2 msg.payload=msg.payload.temperature
3 return msg;
```

The right sidebar shows the 'help' panel for the 'function' node, which explains that it is a JavaScript function to run against messages, with the message body available as `msg`. It also mentions the `msg.payload` property and the `On Start` and `On Stop` tabs.

TEMPERATURE CHART:

The screenshot shows the Node-RED web interface with the same flow as the previous image. The 'chart' node is selected, and the 'Edit chart node' dialog is open. The 'Group' field is set to '[Home] Soil Moisture'. The 'Size' field is set to 'auto'. The 'Label' field is set to 'Temperature'. The 'Type' field is set to 'Line chart'. The 'X-axis' field is set to 'last 1 hours' or '1000 points'. The 'X-axis Label' field is set to 'HH:mm:ss' and 'as UTC' is checked. The 'Y-axis' field is set to 'min' and 'max'. The 'Legend' field is set to 'None' and 'Interpolate' is set to 'linear'. The 'Series Colours' field is set to a grid of colors.

The right sidebar shows the 'help' panel for the 'chart' node, which explains that it plots input values on a chart, either as a time-based line chart, a bar chart, or a pie chart. It also mentions that each input `msg.payload` value will be converted to a number, and if the conversion fails, the message is ignored. It also notes that minimum and maximum Y-axis values are optional and that the graph will auto-scale to any values received. Finally, it states that multiple series can be shown on the same chart by using a different `msg.topic` value on each input message, and that multiple bars of the same series can be shown by using the `msg.label` property.

SWITCH OPTION:

The screenshot shows the Node-RED web interface in a browser. The main workspace displays a flow named 'Flow 1'. The flow starts with an 'inject' node, followed by a 'switch' node. The 'switch' node is connected to an 'http request' node. The 'http request' node is connected to a 'data' node, which is then connected to a 'MOTOR ON' node. The 'MOTOR ON' node is connected to an 'IBM IoT' node. The 'switch' node is currently selected, and the 'Edit switch node' dialog is open. The dialog shows the 'Properties' tab with the following settings: Name (empty), Property (msg.payload), and a value of 20. The 'checking all rules' checkbox is checked. The 'Enabled' checkbox is also checked. The right sidebar shows the 'help' panel with the 'switch' node documentation.

MOTOR ON BUTTON:

The screenshot shows the Node-RED web interface in a browser. The main workspace displays a flow named 'Flow 1'. The flow starts with an 'inject' node, followed by a 'switch' node. The 'switch' node is connected to an 'http request' node. The 'http request' node is connected to a 'data' node, which is then connected to a 'MOTOR ON' node. The 'MOTOR ON' node is connected to an 'IBM IoT' node. The 'MOTOR ON' node is currently selected, and the 'Edit button node' dialog is open. The dialog shows the 'Properties' tab with the following settings: Group ([Home] Soil Moisture), Size (auto), Icon (optional icon), Label (MOTOR ON), Tooltip (optional tooltip), Color (optional text/icon color), Background (optional background color), and When clicked, send: Payload (20), Topic (msg.topic). The 'If msg arrives on input, emulate a button click' checkbox is checked. The 'Enabled' checkbox is also checked. The right sidebar shows the 'help' panel with the 'button' node documentation.

MOTOR OFF BUTTON:

The screenshot shows the Node-RED web interface in a browser. The main workspace displays a flow with a 'switch' node connected to an 'http request' node, which then connects to a 'data' node and a 'command' node. The 'command' node is connected to 'MOTOR ON' and 'MOTOR OFF' nodes. The 'Edit button node' panel is open, showing the following configuration:

- Group:** [Home] Soil Moisture
- Size:** auto
- Icon:** optional icon
- Label:** MOTOR OFF
- Tooltip:** optional tooltip
- Color:** optional text/icon color
- Background:** optional background color
- When clicked, send:**
 - Payload:** 2
 - Topic:** msg.topic
- If msg arrives on input, emulate a button click:** ☐
- Enabled:** ☐ Enabled

The right sidebar shows the 'help' panel for the 'button' node, which includes a description and a list of properties that can be set in the message object.

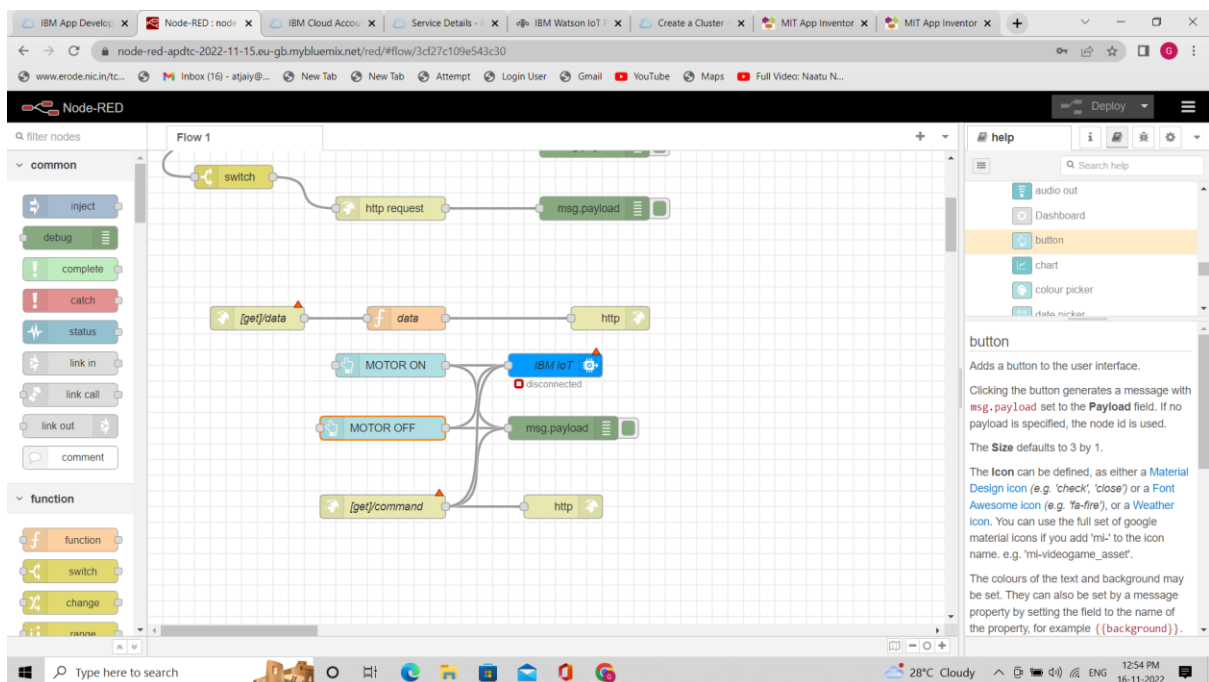
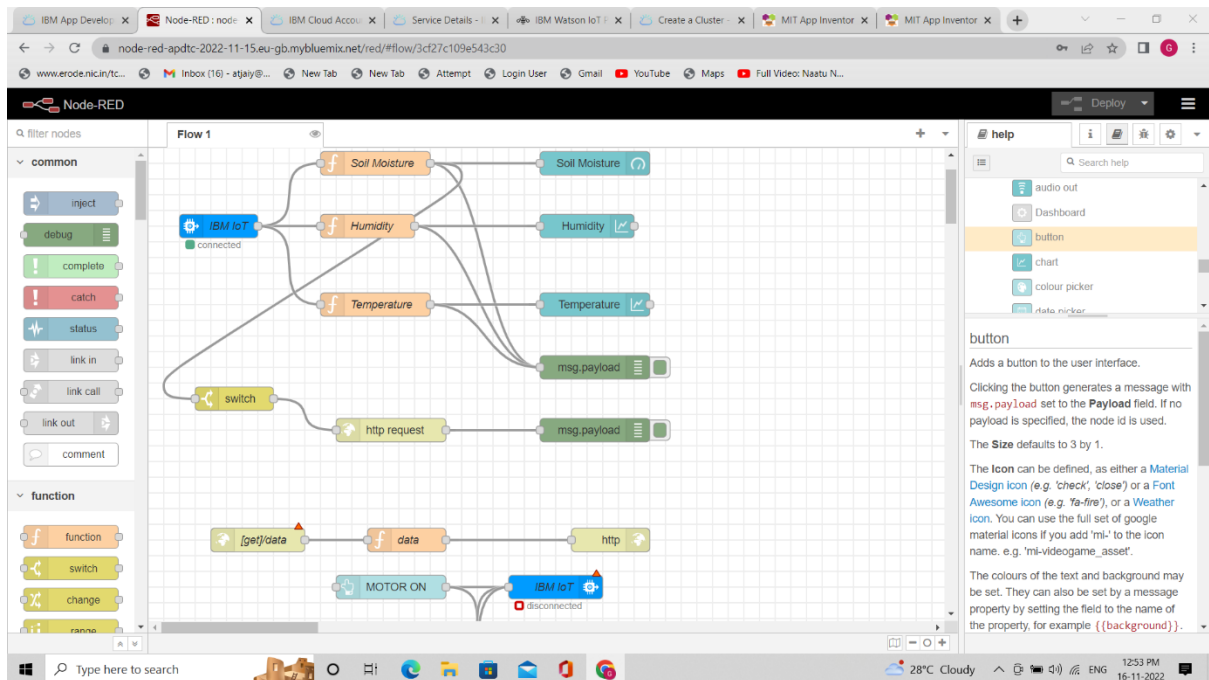
IBM IOT OUT NODE:

The screenshot shows the Node-RED web interface in a browser. The main workspace displays a flow with a 'connected' node connected to a 'Temperature' node, which then connects to a 'switch' node. The 'switch' node is connected to an 'http request' node, which then connects to a 'data' node and a 'command' node. The 'command' node is connected to 'MOTOR ON' and 'MOTOR OFF' nodes. The 'Edit ibmiot out node' panel is open, showing the following configuration:

- Authentication:** API Key
- API Key:** 2c533d245eb7b902
- Output Type:** Device Event
- Device Type:** 2.2.2
- Device Id:** device id e.g. ab12cd231a21
- Event Type:** event type e.g. blink
- Format:** json
- Data:** payload information e.g. data points
- QoS:** 0
- Name:** IBM IoT
- Service:** registered
- Enabled:** ☐ Enabled

The right sidebar shows the 'help' panel for the 'ibmiot out' node, which includes a description and a list of properties that can be set in the message object.

NODE-RED:



BULK SMS:

The screenshot shows the Fast2SMS Developer API dashboard. The left sidebar contains navigation links: Bulk SMS, DLT SMS, Quick SMS, Address Book, Delivery Reports, Transactions, Dev API (selected), Settings, and Help. The main content area is titled 'Dev API' and includes tabs for 'API Key' and 'Security'. A 'Dev API' modal is open, displaying the following configuration:

- For OTP Based SMS use 'OTP SMS API' (with a 'READ API DOCS' link)
- Method: GET
- Route: Quick SMS
- Message (NOTE: Per SMS cost ₹ 3.50): Less soil moisture detected-{{payload}}
- Language: (empty)

On the right, a black box displays the GET request details:

```
GET https://www.fast2sms.com/dev/bulkv2

Query Parameter :

authorization = pSG8ha8x5P2cq3mxMuifDCeOKWHly0nboEslTjdJkZ7UwrQo8l5rKXn1JchBzasUG4L20ih7IVZmj
route = q
message = Less soil moisture detected-{{payload}}
language = "english"
numbers =
flash = "0"

Overall URL = https://www.fast2sms.com/dev/bulkv2?authorization=pSG8ha8x5P2cq3mxMuifDCeOKWHly0nboEslTjdJkZ7UwrQo8l5rKXn1JchBzasUG4L20ih7IVZmj&route=q&message=Less%20soil%20moisture%20detected-%7B%7Bpayload%7D%7D%0A&language=english&flash=0&numbers=
```

NODE RED DASHBOARD HOME:

The screenshot shows the Node-RED dashboard home page. The top bar is blue and labeled 'Home'. The main content area displays two line graphs:

- Humidity**: A line graph showing humidity levels over time, with a y-axis from 0 to 100 and an x-axis from 11:08:00 to 11:53:00.
- Temperature**: A line graph showing temperature levels over time, with a y-axis from 0 to 100 and an x-axis from 11:08:00 to 11:53:00.

Below the graphs are two buttons: 'MOTOR ON' (green) and 'MOTOR OFF' (red). At the bottom, there is a 'Soil Moisture' label. The dashboard is running on a Node-RED instance with the URL `node-red-apdtc-2022-11-15.eu-gb.mybluemix.net`.

NODE RED DEBUG CONSOLE:

The screenshot displays the Node-RED web interface in a browser. The top navigation bar shows the URL: `node-red-apdtdc-2022-11-15-eu-gb.mybluemix.net/red/#flow/3cd27c109e543c30`. The main workspace, titled "Flow 1", contains a flow diagram. It starts with an "IBM IoT" node (labeled "connected") which branches into three function nodes: "Soil Moisture", "Humidity", and "Temperature". Each function node is connected to a corresponding output node: "Soil Moisture" to a "Soil Moisture" node, "Humidity" to a "Humidity" node, and "Temperature" to a "Temperature" node. All three output nodes are connected to a "msg.payload" node. Below this, a "switch" node is connected to an "http request" node (labeled "requesting"), which is also connected to a "msg.payload" node. At the bottom of the flow, there is a sequence of nodes: "Inet1 Irdata", "data", and "http".

On the right side, the "debug" console is open, showing a list of messages. The messages are filtered by "all nodes" and show the following details:

- Message 10: `msg.payload : number`
- Message 26: `11/17/2022, 11:56:11 AM node: 333b6df116c956a7`
`iot-2/type/Device1/d0223/event_1/fmt/json :`
`msg.payload : number`
- Message 31: `11/17/2022, 11:56:11 AM node: 333b6df116c956a7`
`iot-2/type/Device1/d0223/event_1/fmt/json :`
`msg.payload : number`
- Message 50: `11/17/2022, 11:56:11 AM node: 333b6df116c956a7`
`iot-2/type/Device1/d0223/event_1/fmt/json :`
`msg.payload : number`
- Message 2: `11/17/2022, 11:56:11 AM node: 333b6df116c956a7`
`iot-2/type/Device1/d0223/event_1/fmt/json :`
`msg.payload : number`
- Message 31: `11/17/2022, 11:57:10 AM node: 545668aa3c72d9fd`
`msg.payload : Object`
`{ command: "motoron" }`

The bottom status bar shows the system time as 11:57 AM on 17-11-2022, and the weather as 27°C Haze.