

| | |
|---------------|---------------------------------------|
| Date | 04-11-2022 |
| Team ID | PNT2022TMID47454 |
| Project name | Smartfarmer-IOT enabled smart farming |
| Maximum marks | 20 marks |

| | | |
|----------|------|---|
| Sprint-2 | US-1 | Configure the connection security and create API keys that are used in the Node-RED service for accessing the IBM IoT Platform. |
| Sprint-2 | US-2 | Create a Node-RED service. |

US-1 Configure the connection security and create API keys that are used in the Node-red service for accessing the IBM IOT platform

The screenshot displays the IBM Watson IoT Platform user interface. At the top, the header shows 'IBM Watson IoT Platform' and a user profile for '910019106029@smartinternz.com' with ID '7um9ms'. A left-hand navigation menu contains various icons. The main content area features a success message: 'The API key has been added.' Below this, a warning states: 'Authentication tokens are non-recoverable. If you misplace this token, you will need to re-register the API key to generate a new authentication token.'

The interface is divided into two sections: 'Generated Details' and 'API Key Information'. The 'Generated Details' section lists the 'API Key' as 'a-7um9ms-9ppip5cd2t' and the 'Authentication Token' as 'm+Et+BaAZ(UUM?OQAK'. The 'API Key Information' section shows the 'Description' as '-', 'Role' as 'Standard Application', and 'Expires' as 'Never'. A warning icon and text advise: 'Make a note of the generated authentication token. Lost authentication tokens cannot be recovered. If you lose the token, you must reregister the API to generate a new token.'

At the bottom right of the message box are three buttons: 'View API Key', 'Add Another', and 'Close'. Below the message box, the 'Browse API Keys' section is visible, showing a table with one entry: '1 Simulation running'.

US-2 Create a Node-red service

The screenshot shows the Node-RED web interface. On the left, the 'filter nodes' sidebar lists various widgets. The main workspace, 'Flow 1', contains an 'IBM IoT' node connected to a 'debug 1' node, and a 'gauge' node. The 'Edit gauge node' panel on the right is open, showing the following properties:

- Group: [Home] Monitoring
- Size: auto
- Type: Gauge
- Label: Temperature
- Value format: {{value}}
- Units: C
- Range: min 0, max 100
- Colour gradient: A gradient bar with green, yellow, and red segments.
- Sectors: 0, optional, optional, 100
- Class: Optional CSS class name(s) for widget
- Enabled: ☐ Enabled

The 'debug' console on the right shows a series of JSON messages received from the IoT node, including temperature and humidity data.

The screenshot shows the Node-RED web interface with a different configuration. The main workspace, 'Flow 1', contains an 'IBM IoT' node connected to a 'debug 1' node, and a 'function 1' node connected to a 'Temperature' node. The 'Edit function node' panel on the right is open, showing the following properties:

- Name: Temperature Node
- Setup: ☐ Setup
- On Start: ☐ On Start
- On Message: ☒ On Message
- On Stop: ☐ On Stop
- Code:

```
1 msg.payload = msg.payload.Temperature
2 return msg;
```
- Enabled: ☐ Enabled

The 'debug' console on the right shows a series of JSON messages received from the IoT node, including temperature and humidity data.

IBM Watson IoT Platform

910019106029@smartinternz.com
ID: 7um9ms

Browse Action Device Types Interfaces

Add Device

The recent events listed show the live stream of data that is coming and going from this device.

| Event | Value | Format | Last Received |
|---------|---|--------|-------------------|
| event_1 | {"Temperature":35,"Soil moisture":23,"Humidity":... | json | a few seconds ago |
| event_1 | {"Temperature":19,"Soil moisture":61,"Humidity":... | json | a few seconds ago |
| event_1 | {"Temperature":59,"Soil moisture":6,"Humidity":... | json | a few seconds ago |
| event_1 | {"Temperature":43,"Soil moisture":34,"Humidity":... | json | a few seconds ago |
| event_1 | {"Temperature":74,"Soil moisture":22,"Humidity":... | json | a few seconds ago |

Items per page 50 | 1-1 of 1 item

1 of 1 page

1 Simulation running

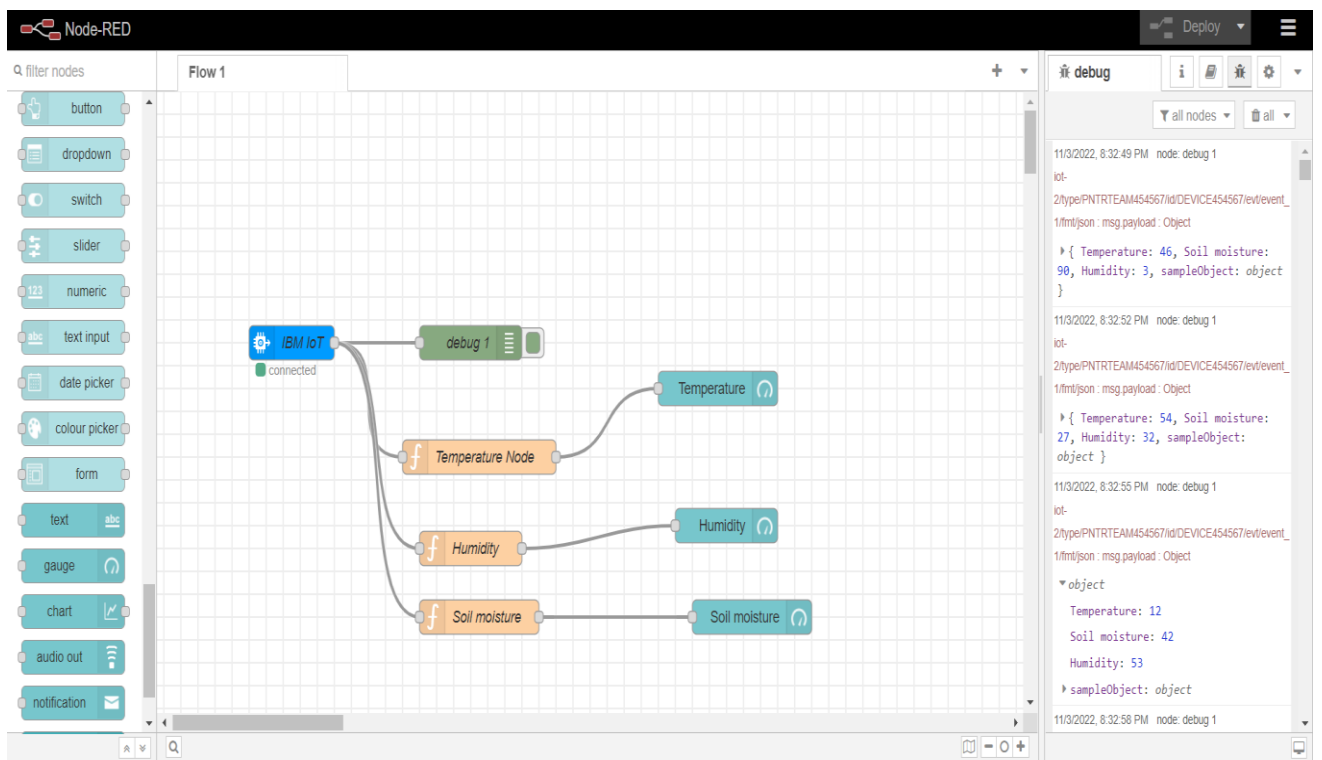


Fig: Monitoring the sensor values- temperature, humidity, soil moisture.

Node-RED interface showing a flow for sending data to IBM IoT. The flow starts with an **ibmiot in** node, which connects to a **debug 1** node and three function nodes: **Temperature Node**, **Soil moisture**, and **Humidity**. These function nodes then connect to a **Motor on** node, which connects to a **Motor off** node, and finally to a **debug 2** node.

The **Edit ibmiot out node** configuration panel shows the following settings:

- Authentication: API Key
- API Key: 180ab8184e1e556d
- Output Type: Device Command
- Device Type: PNTRTEAM454567
- Device Id: DEVICE454567
- Command Type: onoff
- Format: json
- Data: data
- QoS: 0
- Name: IBM IoT

The **debug** console shows the following log entries:

```
11/3/2022, 8:35:19 PM node: debug 1
iot-
2/type:PNTRTEAM454567/id:DEVICE454567/ev:event_
1/fmt/json : msg payload : Object
{ Temperature: 78, Soil moisture: 75, Humidity: 72, sampleObject: object }
11/3/2022, 8:35:22 PM node: debug 1
iot-
2/type:PNTRTEAM454567/id:DEVICE454567/ev:event_
1/fmt/json : msg payload : Object
{ Temperature: 52, Soil moisture: 58, Humidity: 82, sampleObject: object }
11/3/2022, 8:35:25 PM node: debug 1
iot-
2/type:PNTRTEAM454567/id:DEVICE454567/ev:event_
1/fmt/json : msg payload : Object
{ Temperature: 73, Soil moisture: 94, Humidity: 73, sampleObject: object }
11/3/2022, 8:35:28 PM node: debug 1
iot-
```

Node-RED interface showing a flow for receiving data from IBM IoT. The flow starts with an **ibmiot in** node, which connects to a **debug 1** node and three function nodes: **Temperature Node**, **Soil moisture**, and **Humidity**. These function nodes then connect to a **httpfunctionnode** node, which connects to a **Motor on** node, which connects to a **Motor off** node, and finally to a **debug 2** node.

The **Edit ibmiot in node** configuration panel shows the following settings:

- Authentication: API Key
- API Key: 180ab8184e1e556d
- Input Type: Device Event
- Device Type: ☐ All or PNTRTEAM454567
- Device Id: ☐ All or DEVICE454567
- Event: ☒ All or +
- Format: ☐ All or json
- QoS: 0
- Name: IBM IoT
- Service: registered

The **debug** console shows the following log entries:

```
11/4/2022, 9:10:44 PM node: debug 2
msg payload : Object
{ command: "motoron" }
11/4/2022, 9:10:45 PM node: debug 2
msg payload : Object
{ command: "motoron" }
11/4/2022, 9:10:46 PM node: debug 2
msg payload : Object
{ command: "motoron" }
11/4/2022, 9:10:50 PM node: debug 2
msg payload : Object
{ command: "motoroff" }
11/4/2022, 9:10:50 PM node: debug 2
msg payload : Object
{ command: "motoroff" }
```

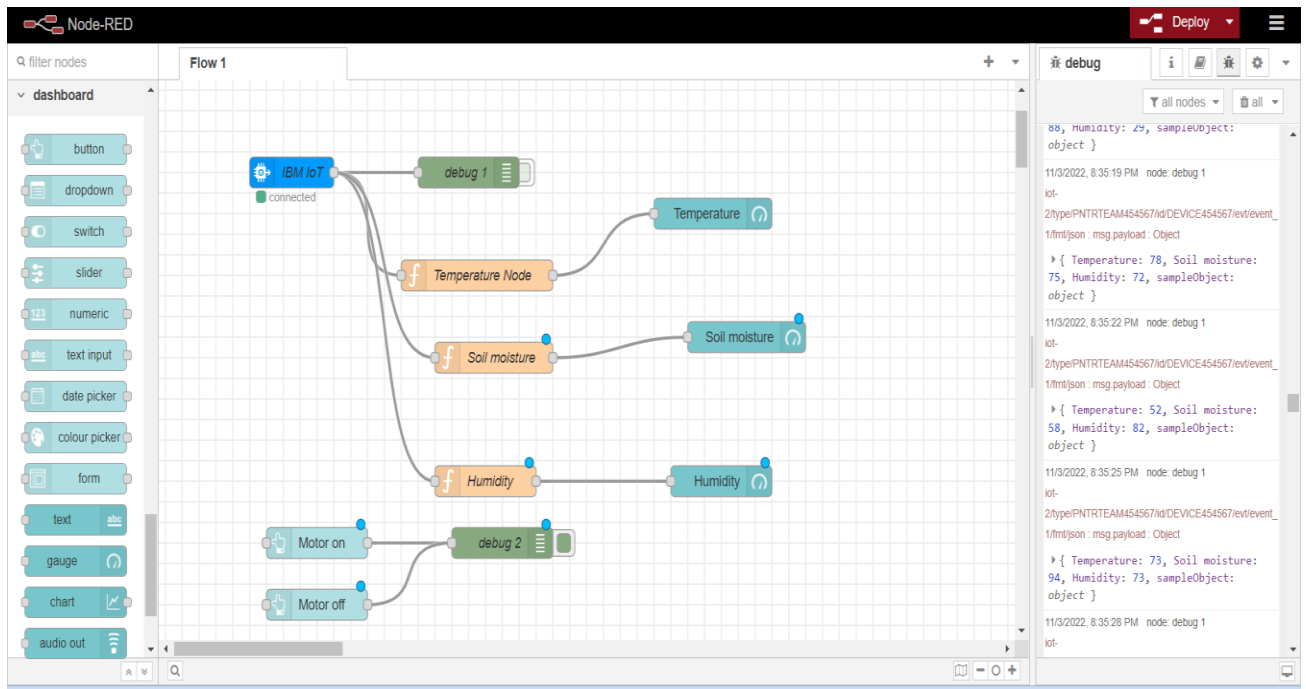
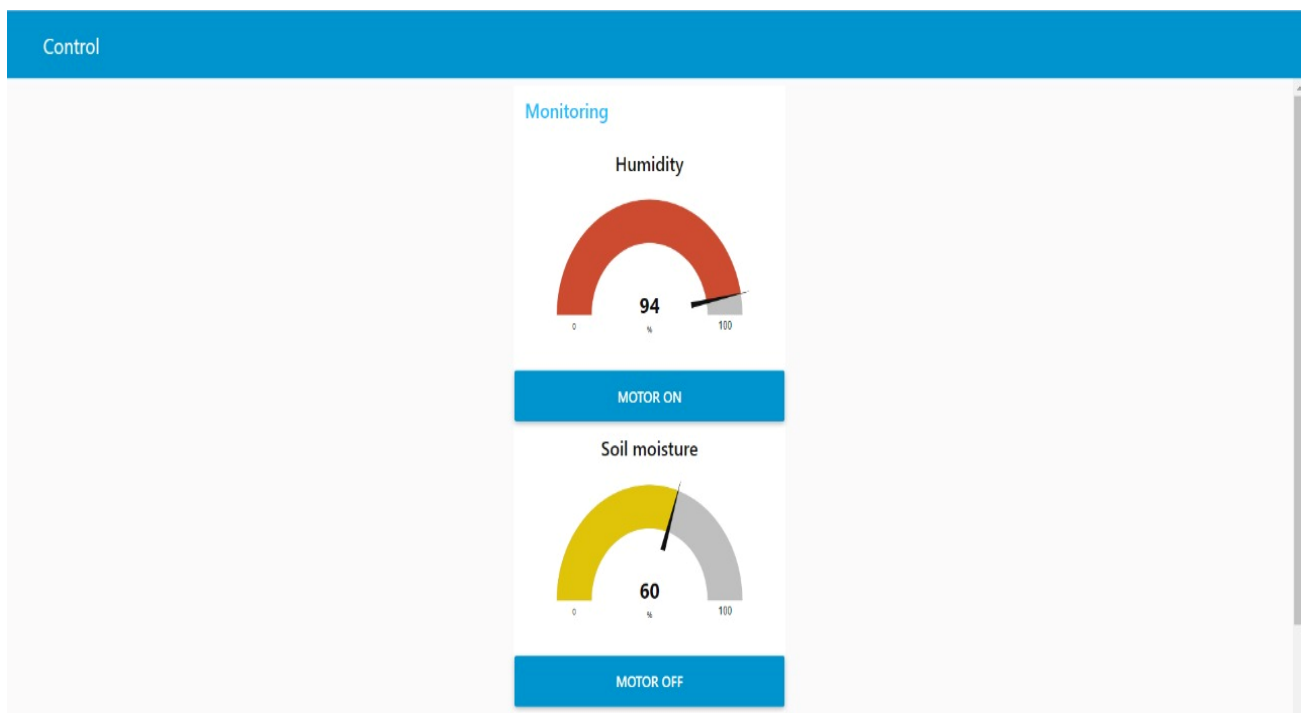


Fig: Control buttons (motor ON, motor OFF) and Properties



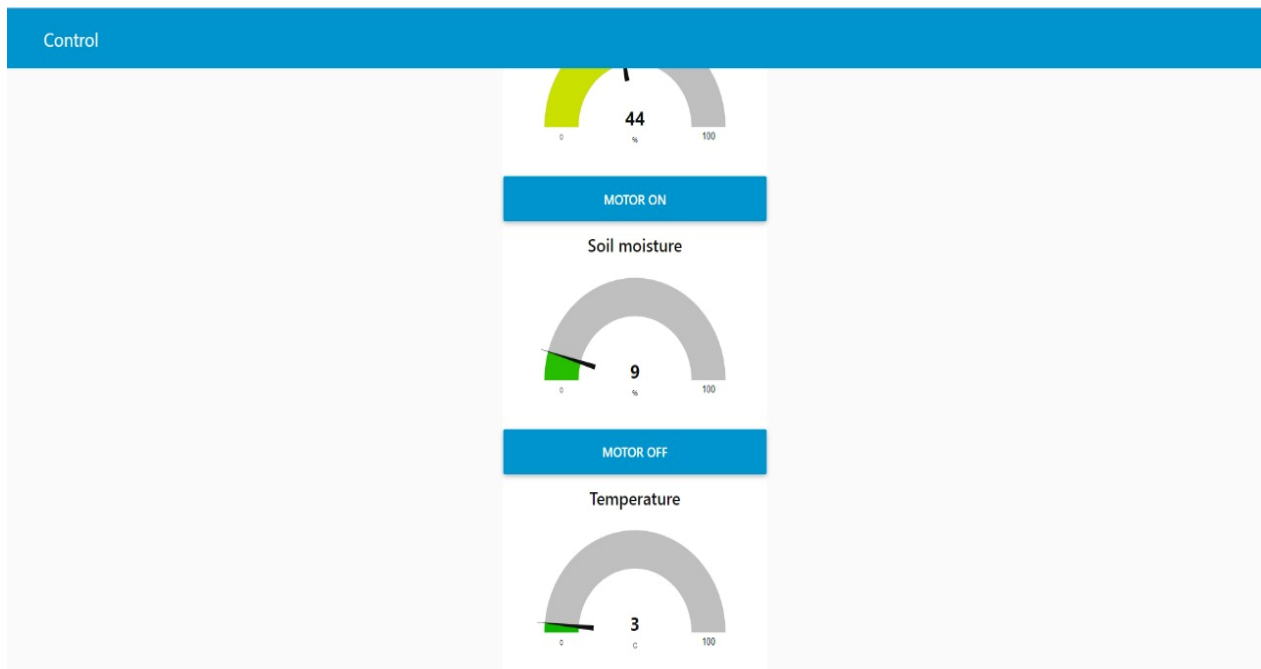


Fig: Output from recent event

MIT APP INVERTER TO DESIGN THE APP:

