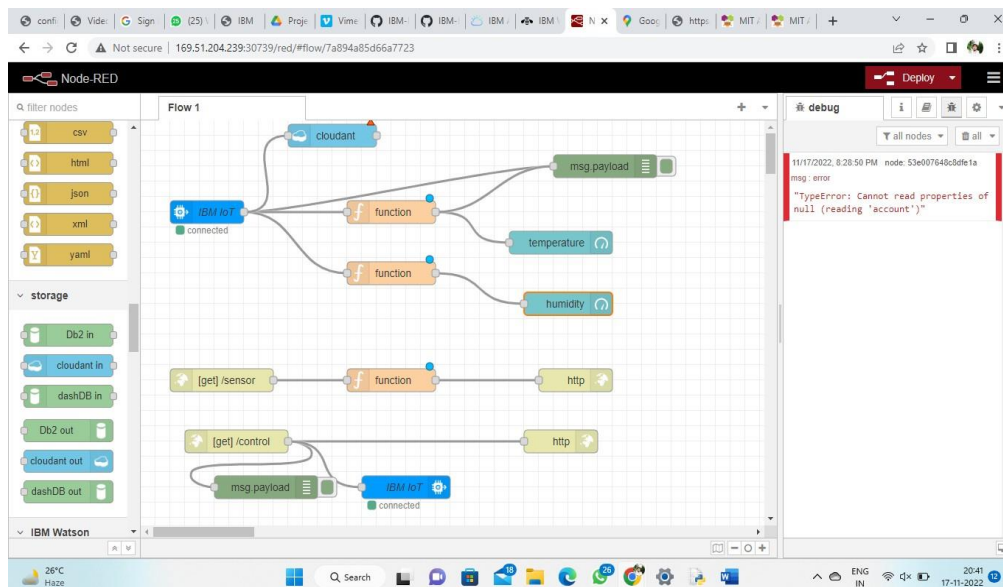


Configure The Application To Receive The Data From Cloud

Date	17 November 2022
Team Id	PNT2022TMID26592
Title	Hazardous Area Monitoring for Industrial Plant using IoT

Node red flow created to get values



Configuring function to fetch the desired value

The screenshot shows the Node-RED web interface in a browser. The left sidebar contains a 'filter nodes' search bar and a list of nodes including 'csv', 'html', 'json', 'xml', 'yaml', and a 'storage' section with various database nodes. The main workspace displays a flow with an 'IBM IoT' node connected to a function node. The function node is selected, and its configuration panel is open. The 'Properties' tab shows the node name as 'Name'. The 'On Message' tab is active, displaying a JavaScript snippet:

```
1 msg.payload = msg.payload.temp;
2 global.set('t',msg.payload)
3 return msg;
```

The right sidebar shows the 'debug' console with a log entry from 11/17/2022, 8:28:50 PM, indicating a 'msg : error' and a 'TypeError: Cannot read properties of null (reading 'account')'.

This screenshot shows the same Node-RED interface as the first, but with the function node's JavaScript snippet updated to fetch the 'humid' value from the message payload:

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

The 'debug' console on the right still shows the same error message from the previous state.

App Blocks to render the values and display it in app

The screenshot displays the MIT App Inventor web application interface. The browser address bar shows the URL `ai2.appinventor.mit.edu/#5439689839214592`. The interface includes a top navigation bar with links for Projects, Connect, Build, Settings, and Help. The main workspace is divided into two panels: 'Blocks' on the left and 'Viewer' on the right. The 'Blocks' panel shows a list of built-in components, including Control, Logic, Math, Text, Lists, Dictionaries, Colors, Variables, and Procedures. The 'Viewer' panel displays the code for the selected screen, 'Screen3'. The code is organized into three sections: a timer event, a web request event, and a button click event. The timer event sets the URL of 'Web1' to `http://169.51.204.239:30739/sensor` and calls `Web1.Get`. The web request event for 'Web1' processes the response, decoding JSON data and setting the text of 'TextBox1' and 'TextBox2' based on the response content. The button click event for 'Button1' sets the URL of 'Web2' to `http://169.51.204.239:30739/control?command=alar...` and calls `Web2.Get`. The 'Show Warnings' button at the bottom left indicates 0 warnings and 2 errors. The Windows taskbar at the bottom shows the date 17-11-2022 and time 23:06.