

SPRINT DELIVERY – III

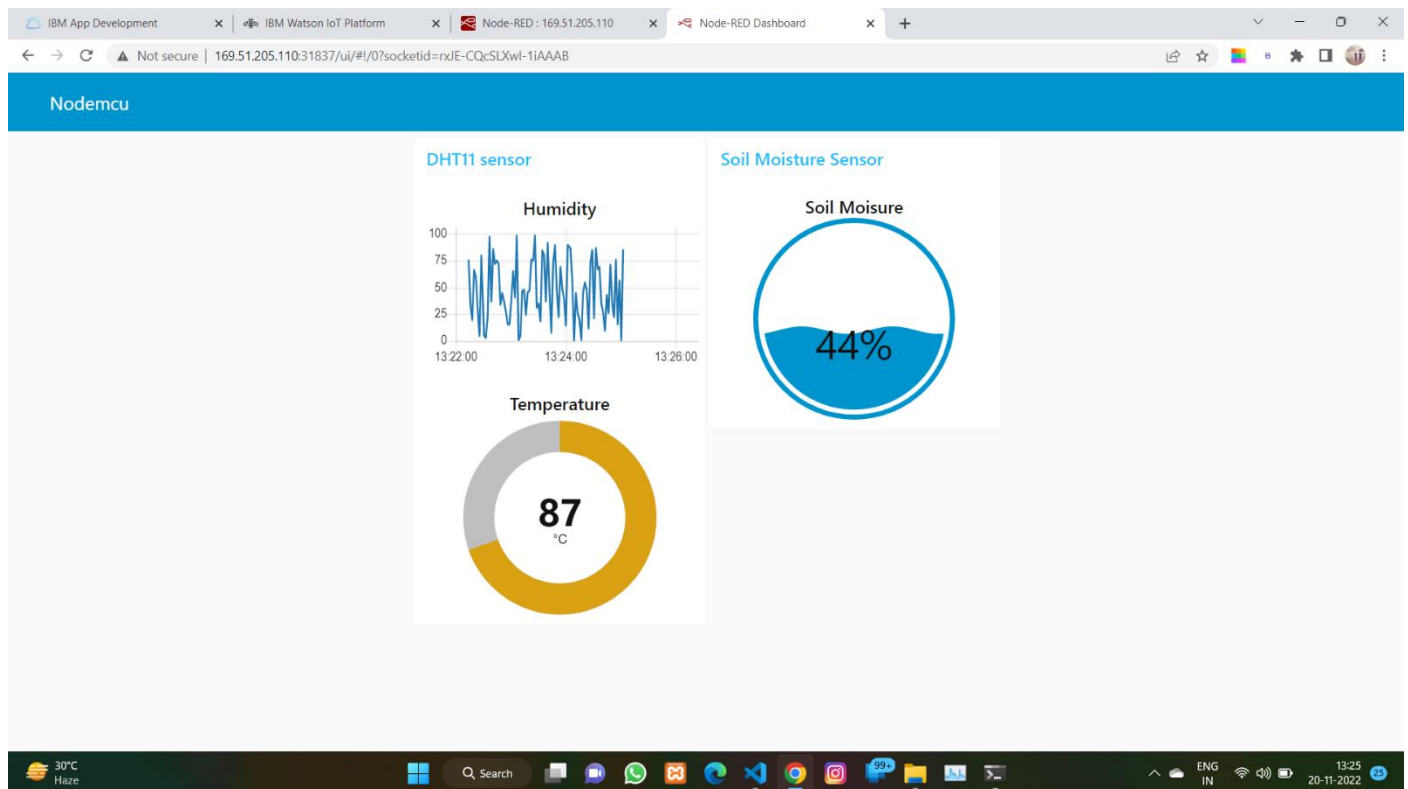
Project Name: SmartFarmer-Iot Enabled SmartFarming Application.

Team ID: PNT2022TMID32489

Build A Web Application Using Node-RED Service

- ✓ Node-Red Contains /Data and /Command Requests
- ✓ /Data command is used to access the MIT App Inventor
- ✓ /Command is used to send and receive the Motor Control response and request(on/off)
- ✓ Every one second the parameter is update
- ✓ App inventor contains blocks, with these help of these blocks we can build our application

Node-Red UI



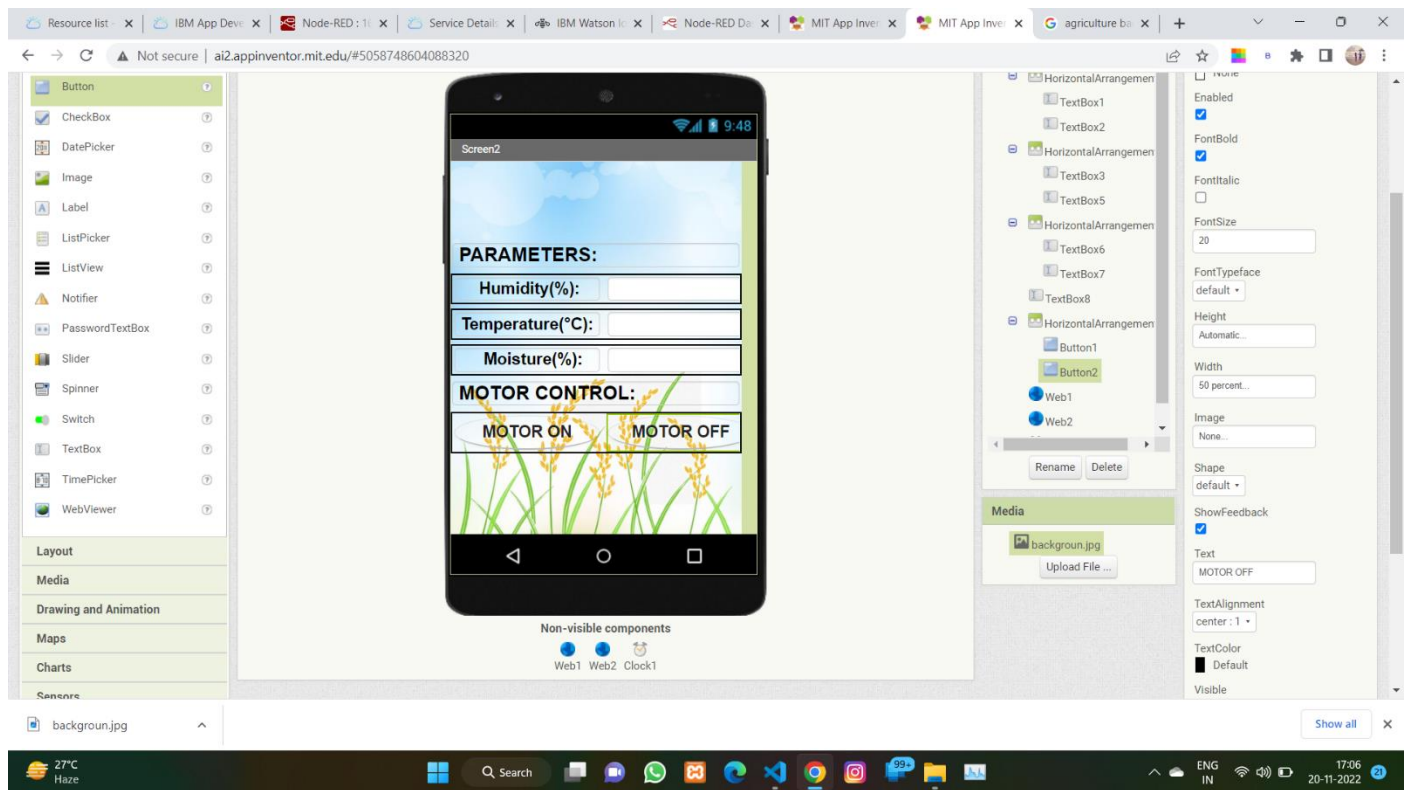
Configurations:

The screenshot displays the Node-RED web interface in a browser. The top bar shows multiple open tabs including 'Resource list', 'IBM App Dev', 'Node-RED', 'Service Details', 'IBM Watson', and 'MIT App Inventor'. The main workspace, titled 'Flow 1', contains a flow starting with an 'IBM IoT' node (labeled 'connected'). This node branches into three function nodes: 'Temperature', 'Humidity', and 'Moisture'. Each function node is connected to a corresponding 'msg.payload' node, which then feeds into a respective display node: 'Temp', 'Humidity', and 'Soil Moisture'. Below this, there is a '[get] /data' node connected to a 'data' function node, which is then connected to an 'http' node. Another section shows a 'Motor Switch' node connected to an 'IBM IoT' node (labeled 'connected') and an 'http' node. A '[get] /command' node is also connected to the 'http' node. The left sidebar shows a 'filter nodes' search bar and a 'network' category with various input/output nodes like 'mqtt in/out', 'http in/out', 'websocket in/out', 'tcp in/out', and 'udp in/out'. The right sidebar shows a 'debug' console. The bottom of the screen shows a Windows taskbar with a weather widget indicating 27°C and 'Haze', and a system clock showing 17:03 on 20-11-2022.

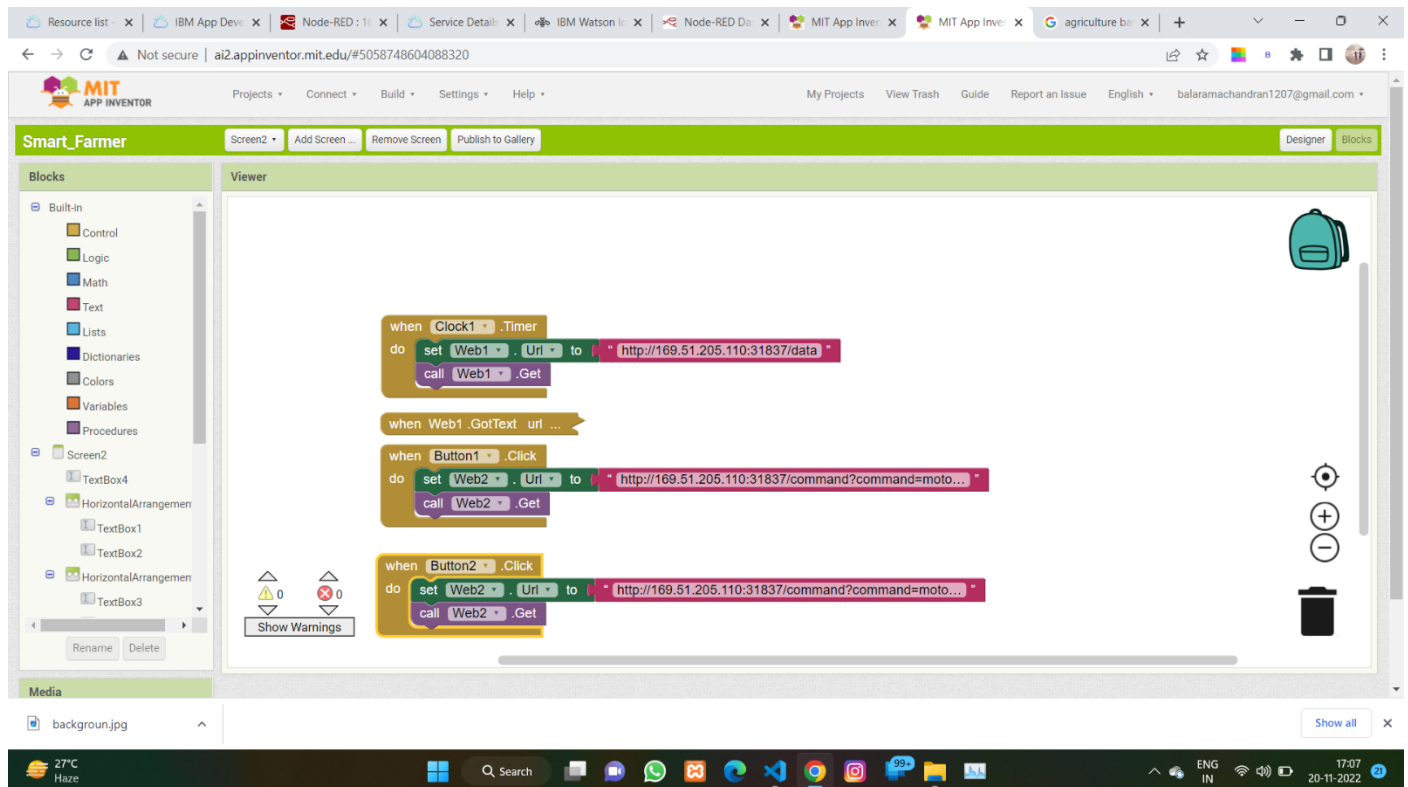
MIT App Inventor Home Page

The screenshot shows the MIT App Inventor web interface. The top bar includes tabs for 'Resource list', 'IBM App Dev', 'Node-RED', 'Service Details', 'IBM Watson', 'MIT App Inventor', and 'agriculture ba'. The main workspace is divided into four panels: 'Palette', 'Viewer', 'Components', and 'Properties'. The 'Palette' panel on the left lists various components under categories like 'User Interface', 'Layout', 'Media', 'Drawing and Animation', 'Maps', 'Charts', 'Sensors', 'Social', 'Storage', 'Connectivity', 'LEGO® MINDSTORMS®', 'Experimental', and 'Extension'. The 'Viewer' panel in the center shows a mobile app design for 'Screen1' with a background image of rice plants and the text 'Welcome To SmartFarmer App' and 'Home Page'. The 'Components' panel on the right shows a tree view of the app's components, including 'Screen1', 'TextBox1', and 'Button1'. The 'Properties' panel on the far right shows the properties for 'Button1', including 'BackgroundColor', 'Enabled', 'FontBold', 'FontItalic', 'FontSize', 'FontTypeface', 'Height', 'Width', 'Image', 'Shape', 'ShowFeedback', 'Text', and 'TextAlignment'. The bottom of the screen shows a Windows taskbar with a weather widget indicating 27°C and 'Haze', and a system clock showing 17:06 on 20-11-2022.

MIT App Inventor Dashboard



App Configuration



Resource list x IBM App Dev x Node-RED : 1 x Service Detail x IBM Watson x Node-RED Da x MIT App Inver x MIT App Inver x agriculture ba x

Not secure | ai2.appinventor.mit.edu/#505874860408320

MIT APP INVENTOR

Projects Connect Build Settings Help My Projects View Trash Guide Report an Issue English balaramachandran1207@gmail.com

Smart_Farmer Screen2 Add Screen Remove Screen Publish to Gallery Designer Blocks

Blocks

- Built-in
 - Control
 - Logic
 - Math
 - Text
 - Lists
 - Dictionaries
 - Colors
 - Variables
 - Procedures
- Screen2
 - TextBox4
 - HorizontalArrangemen
 - TextBox1
 - TextBox2
 - HorizontalArrangemen
 - TextBox3

Media

backgroun.jpg Show all x

Viewer

when Clock1.Timer

do

- set Web1.Uri to http://169.51.205.110:31837/data
- call Web1.Get

when Web1.GetText

do

- set TextBox2.Text to look up in pairs key humidity command:moi pairs call Web1.JsonTextDecode jsonText get responseContent notFound Returns the value associated with the key in the list of pairs
- set TextBox5.Text to look up in pairs key temperature pairs call Web1.JsonTextDecode jsonText get responseContent notFound not found
- set TextBox7.Text to look up in pairs key moisture pairs call Web1.JsonTextDecode jsonText get responseContent notFound not found

Show Warnings 0 0

27°C Haze 17:07 20-11-2022