Problem Statement

Inefficient organization and management of household items can lead to inconvenience and wastage of resources. Traditional drawers lack intelligence and fail to provide timely reminders or monitor item quantities. There is a need for an innovative approach to household automation and organization that addresses these challenges. The aim of this project is to develop smart drawers equipped with reminder systems and weight sensing technologies to improve convenience, optimize inventory management, and enhance overall household automation. By integrating smart features into drawers, we seek to enhance user experience, reduce wastage, and streamline everyday tasks.

Main Objectives

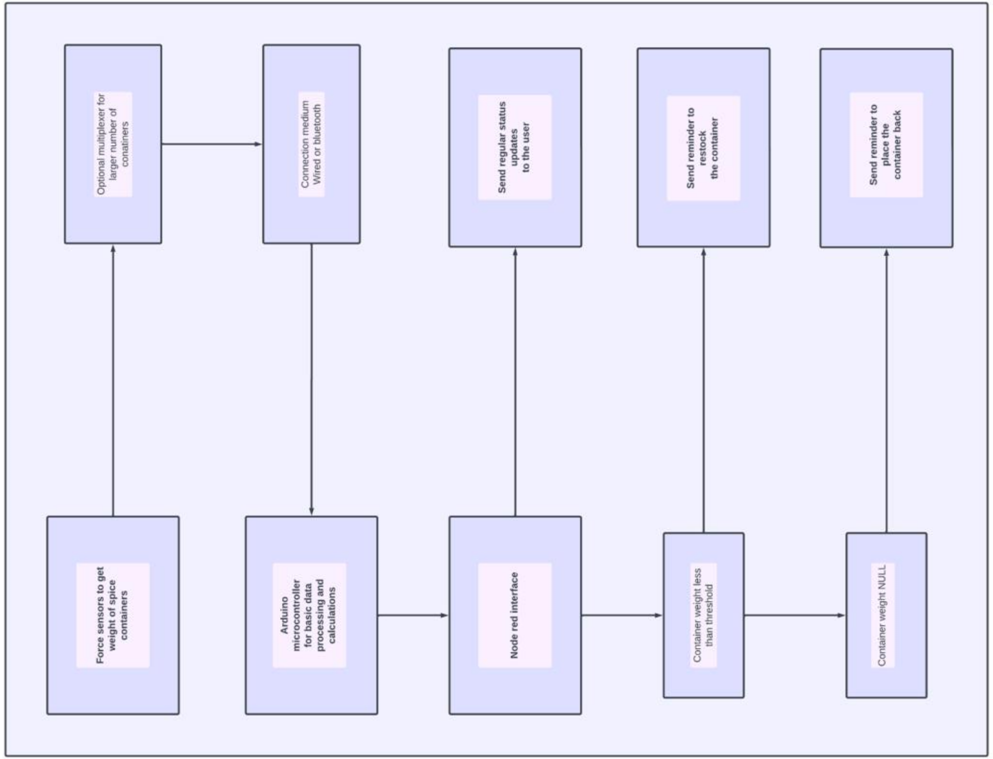
The main objectives of the project include:

- Increasing home organization, Smart drawers help in keeping the drawer organized by sending a reminder to the user when the object is not put back in its place. This feature can also be turned off in case the items are not meant to be kept back in its place.
- Restocking, Smart drawers update the users about the status of its containers based on its weight. The capacity of the container is derived based on predefined maximum weight and current weight, which is updated regularly and sent to the user. This makes it easier for the user to restock the containers.

Apparatus Required

- Force Sensors
- Arduino Uno
- Makeshift “Drawer” (shoebox)
- 10k ohm Resistors

Block Diagram


Force sensors to get weight of spice containers
Optional multiplexer for larger number of conatiners
Connection medium Wired or bluetooth
Arduino microcontroller for basic data processing and calculations
Node red interface
Send regular status updates to the user
Container weight less than threshold
Send reminder to restock the container
Container weight NULL
Send reminder to place the container back

PROTOTYPE

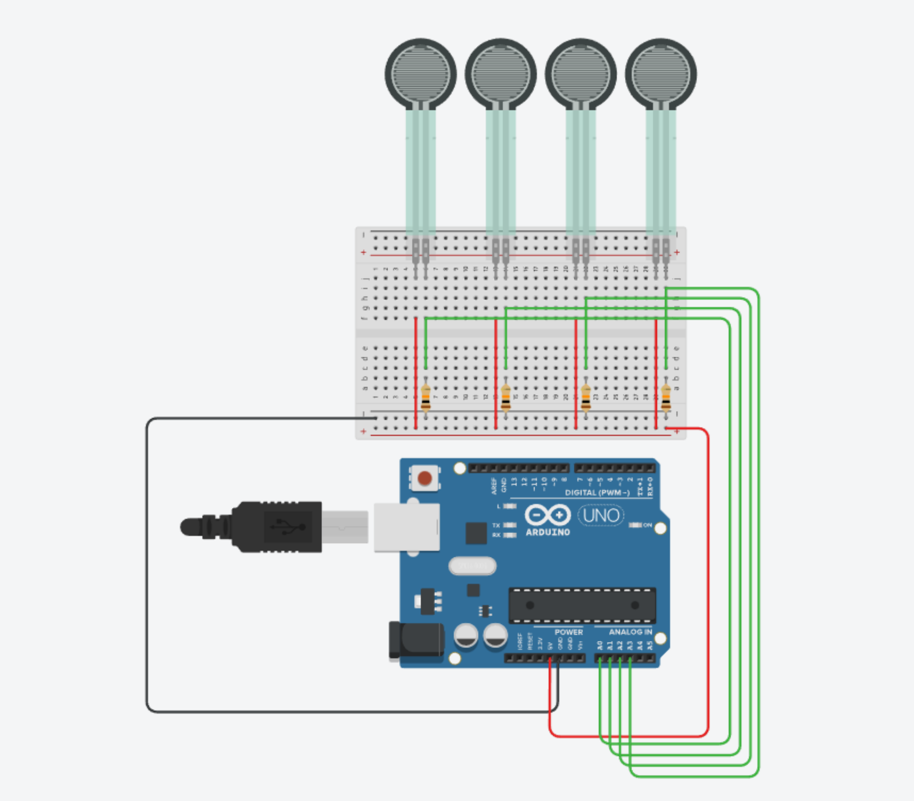
tinkercad

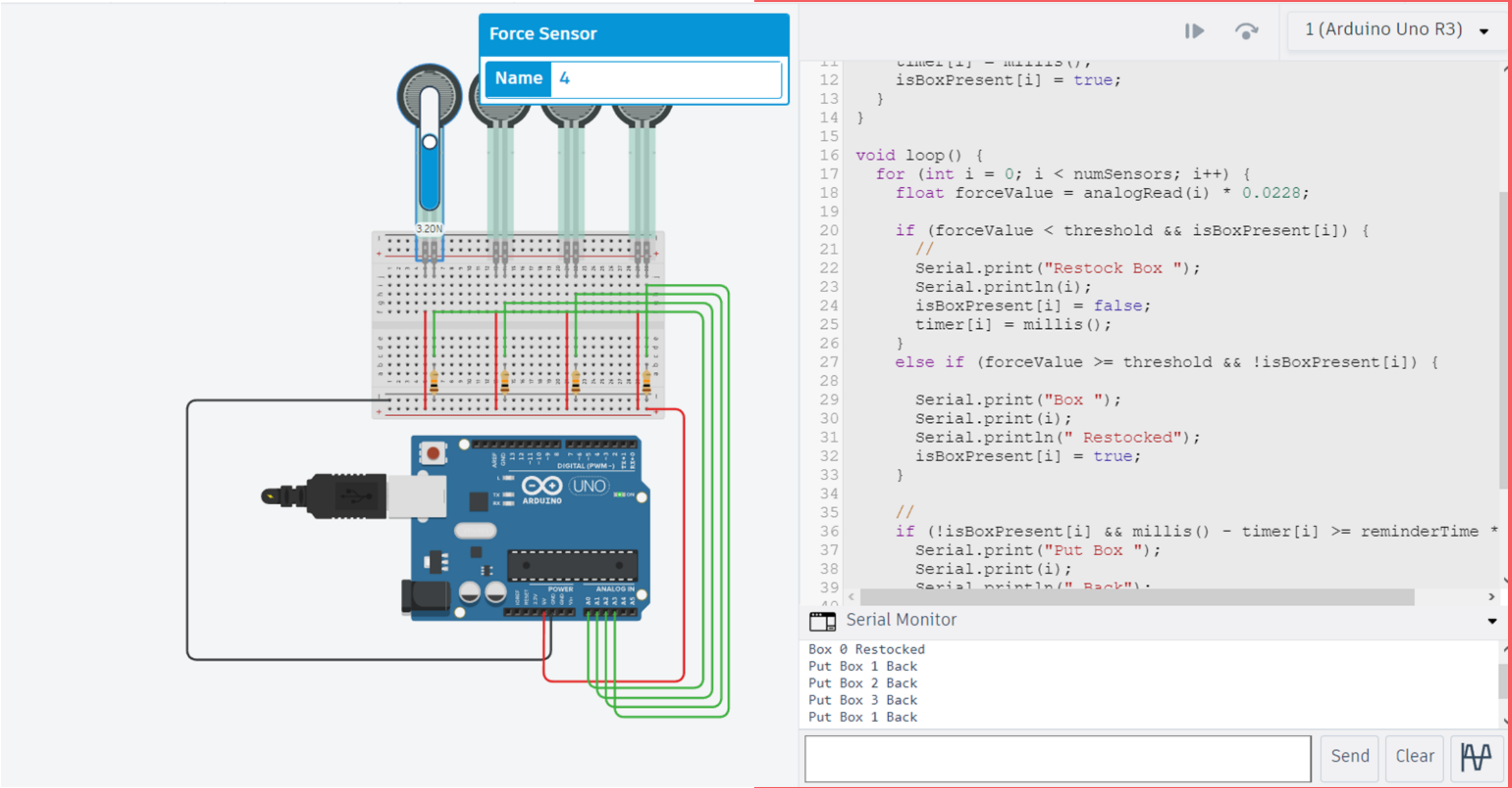
The process of creating the prototype was challenging, yet enjoyable. With prior knowledge to the working principle of Tinkercad, the circuit was made relatively easily. The hardest part was the implementation of force sensors in Tinkercad. Along with trying to understand how the force sensor worked, the coding had to be tweaked in order to convert the reading from newtons to grams. One obstacle that was overcome was the alerts given by the circuit. Initially, the alerts only occurred when the force was 0 newtons. However, by changing small parts of the code, we were able to send alerts when the force was above a specific value.

Code

```
const int numSensors = 4;
const float threshold = 10.0;
const int reminderTime = 10;
unsigned long timer[numSensors];
bool isBoxPresent[numSensors];
void setup() {
 Serial.begin(9600);
 for (int i = 0; i < numSensors; i++) {
   pinMode(i, INPUT);
   timer[i] = millis();
   isBoxPresent[i] = true;
 }
}

void loop() {
 for (int i = 0; i < numSensors; i++) {
   float forceValue = analogRead(i) * 0.0228;

   if (forceValue < threshold && isBoxPresent[i]) {
     //
     Serial.print("Restock Box ");
     Serial.println(i);
     isBoxPresent[i] = false;
     timer[i] = millis();
   }
   else if (forceValue >= threshold && !isBoxPresent[i]) {

     Serial.print("Box ");
     Serial.print(i);
     Serial.println(" Restocked");
     isBoxPresent[i] = true;
   }

   //
   if (!isBoxPresent[i] && millis() - timer[i] >= reminderTime * 1000) {
     Serial.print("Put Box ");
     Serial.print(i);
     Serial.println(" Back");
     timer[i] = millis();
   }
 }

 delay(100);
}
```

Circuit Diagram





```
      timer[i] = millis();
      isBoxPresent[i] = true;
    }
}

void loop() {
  for (int i = 0; i < numSensors; i++) {
    float forceValue = analogRead(i) * 0.0228;

    if (forceValue < threshold && isBoxPresent[i]) {
      //
      Serial.print("Restock Box ");
      Serial.println(i);
      isBoxPresent[i] = false;
      timer[i] = millis();
    }
    else if (forceValue >= threshold && !isBoxPresent[i]) {

      Serial.print("Box ");
      Serial.print(i);
      Serial.println(" Restocked");
      isBoxPresent[i] = true;
    }

    //
    if (!isBoxPresent[i] && millis() - timer[i] >= reminderTime *
      Serial.print("Put Box ");
      Serial.print(i);
      Serial.println(" Back");
```

```
Box 0 Restocked
Put Box 1 Back
Put Box 2 Back
Put Box 3 Back
Put Box 1 Back
```

EXECUTION

arduino IDE

Executing using Arduino

PROCEDURE:

1. Create the circuit using a 10k ohm resistor and force sensor.
2. Connect this circuit to the computer.
3. Open Arduino IDE and write the code.
4. Go to tools and select the correct port.
5. Verify and upload the code.
6. Observe the output in the Serial Monitor.

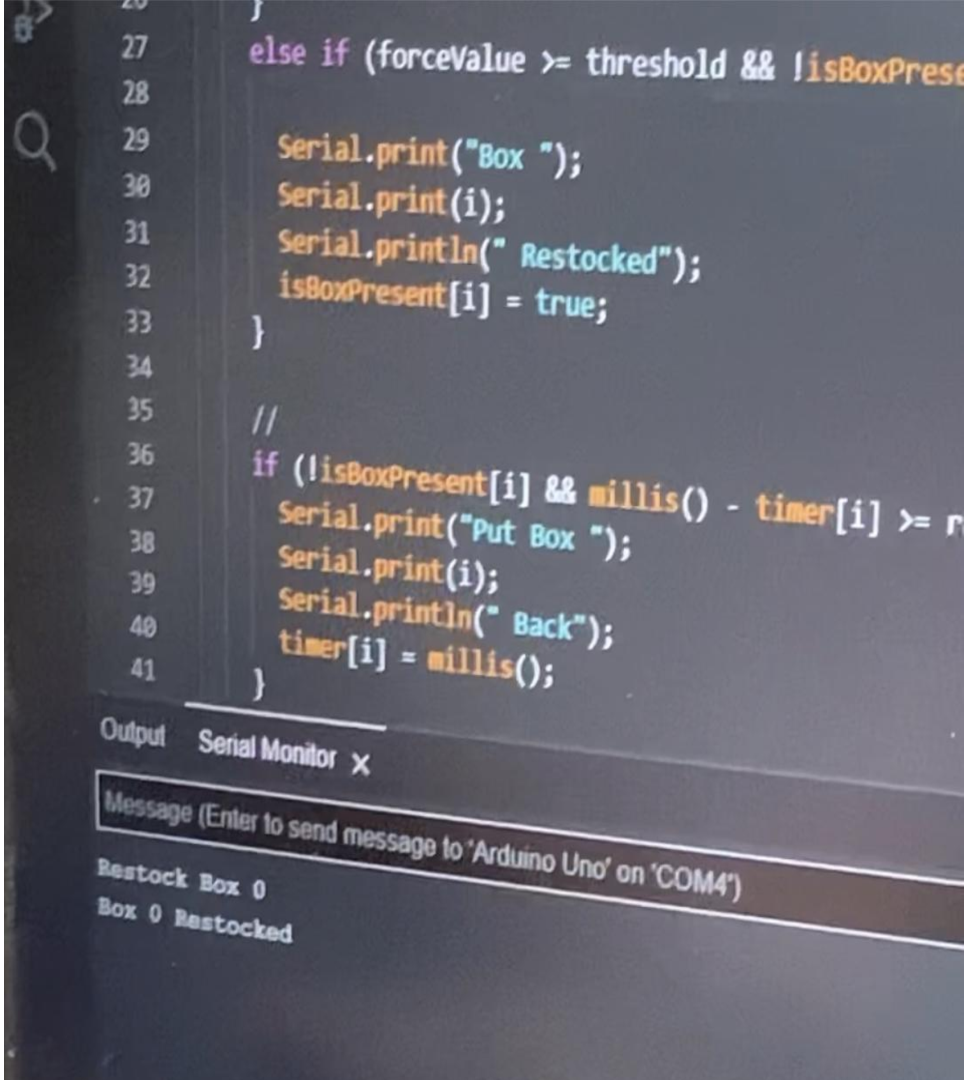
Executing using Arduino

For this experiment, we use a ketchup bottle to test if the Smart Drawer can detect it’s weight. The thresholds in the code had to be adjusted in order for the weight to be correctly detected by the sensor. In order for the weight of the ketchup bottle to be detected by the sensor, we put erasers at the bottom of the box, where the sensor is, and put a cardboard square on top of the erasers (like a false bottom). We place the ketchup bottle on the top of the ‘false bottom’.

When the sensor detects a weight below the mentioned threshold, a message appears on the Serial Monitor “Put Box 0 Back”, and after a time delay, “Restock Box 0”. When the sensor detects a weight above the mentioned threshold, a message appears on the Serial Monitor “Box 0 Restored”.

OUTPUT

of arduino IDE



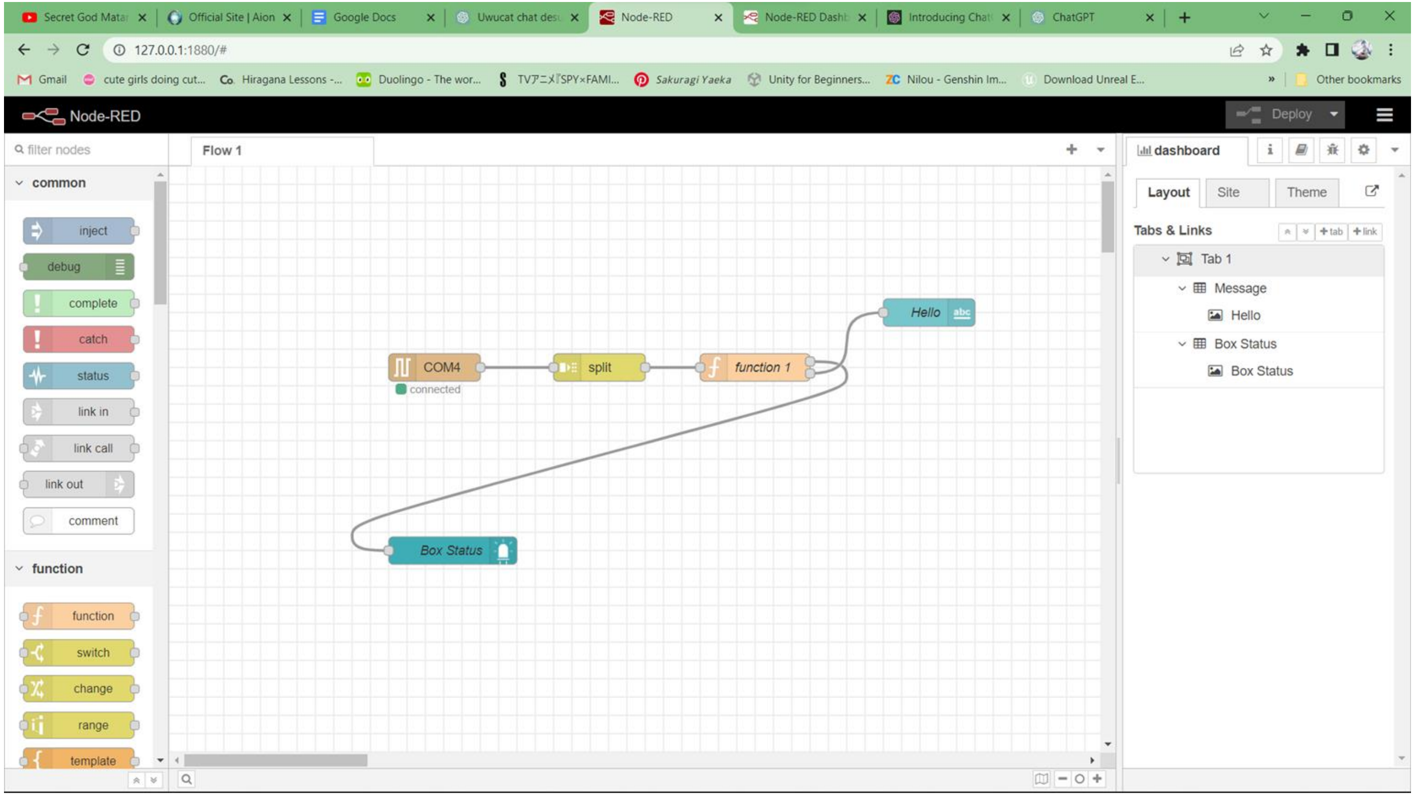
```
}
else if (forceValue >= threshold && !isBoxPrese

  Serial.print("Box ");
  Serial.print(i);
  Serial.println(" Restocked");
  isBoxPresent[i] = true;
}

//
if (!isBoxPresent[i] && millis() - timer[i] >= r
  Serial.print("Put Box ");
  Serial.print(i);
  Serial.println(" Back");
  timer[i] = millis();
}
```

```
Restock Box 0
Box 0 Restocked
```

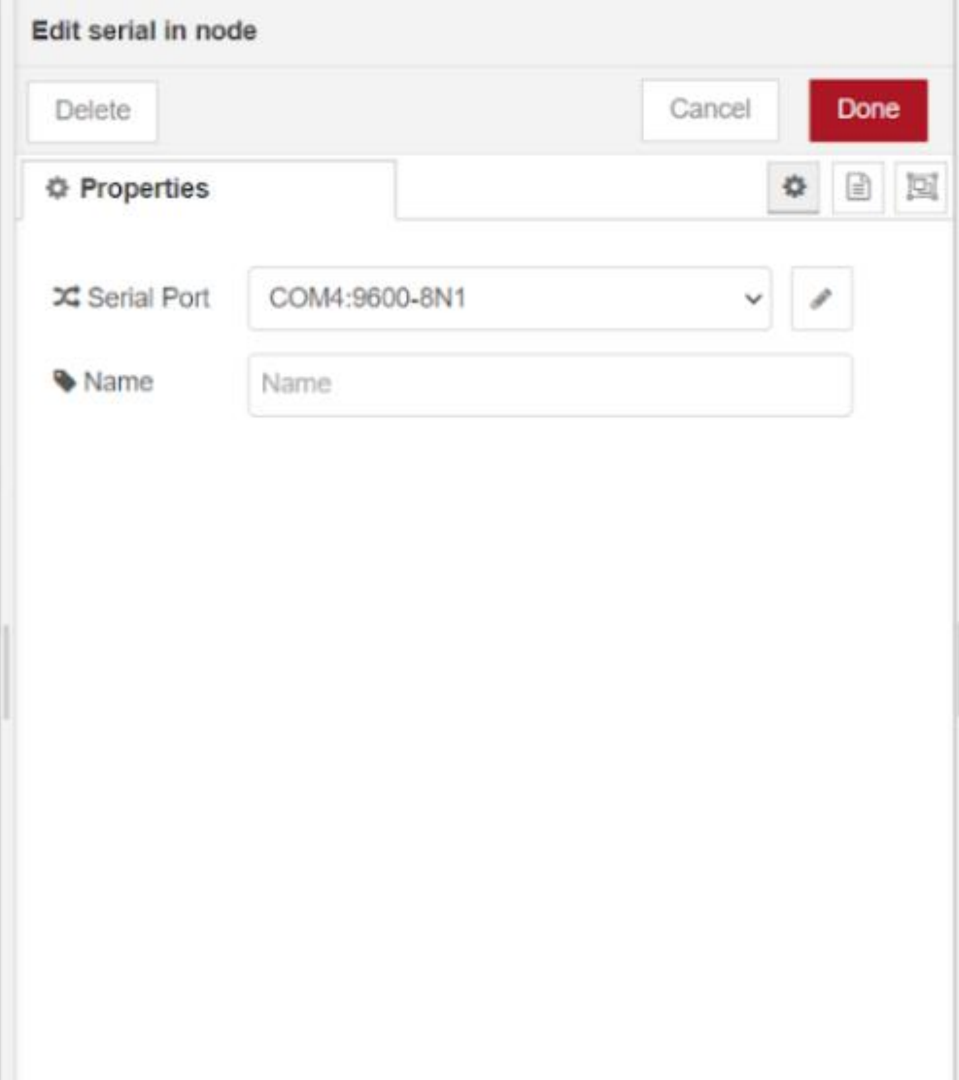
EXECUTION

node-red

Block/Circuit Diagram

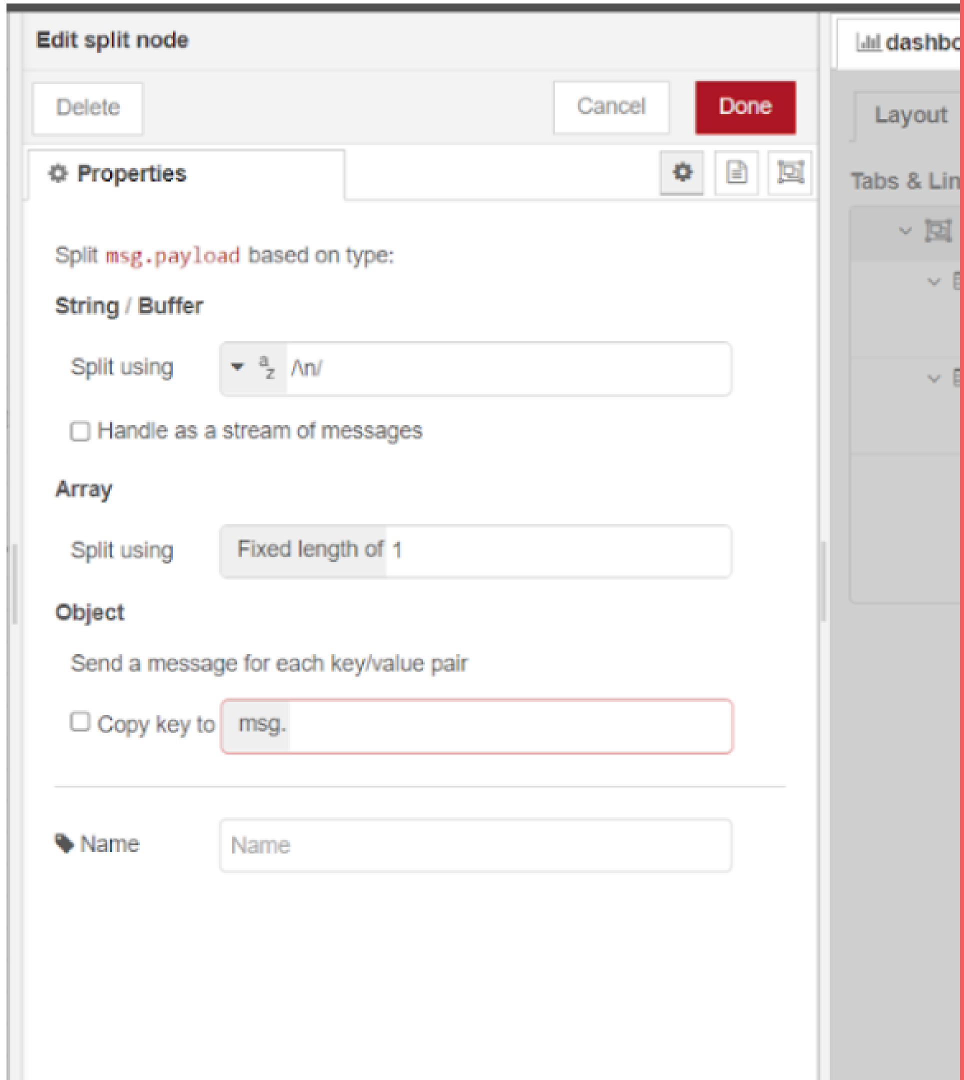


Serial Node



Serial Port COM4:9600-8N1

Name Name



Edit split node

Split `msg.payload` based on type:

String / Buffer

Split using ▾ ᵃz /\n/

□ Handle as a stream of messages

Array

Split using Fixed length of 1

Object

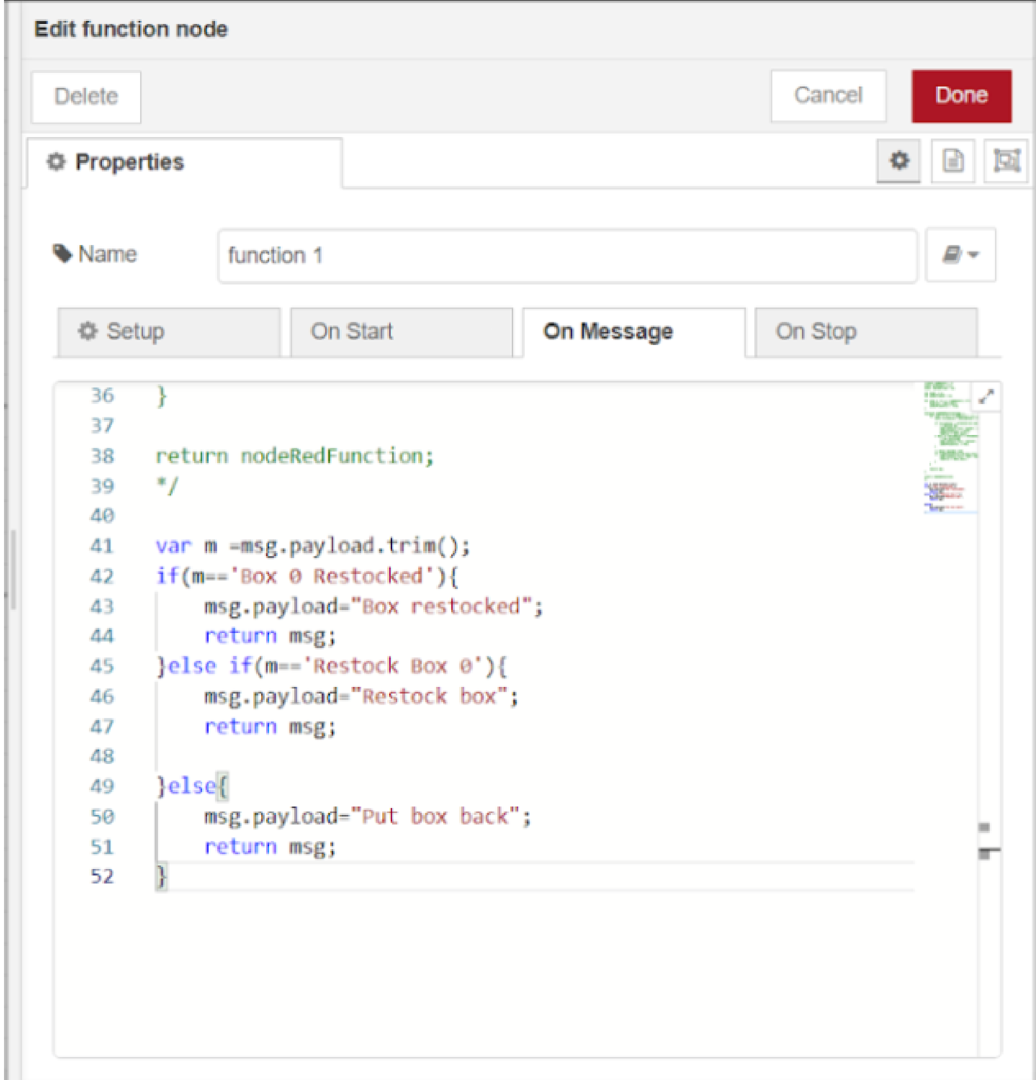
Send a message for each key/value pair

□ Copy key to msg.

Name Name

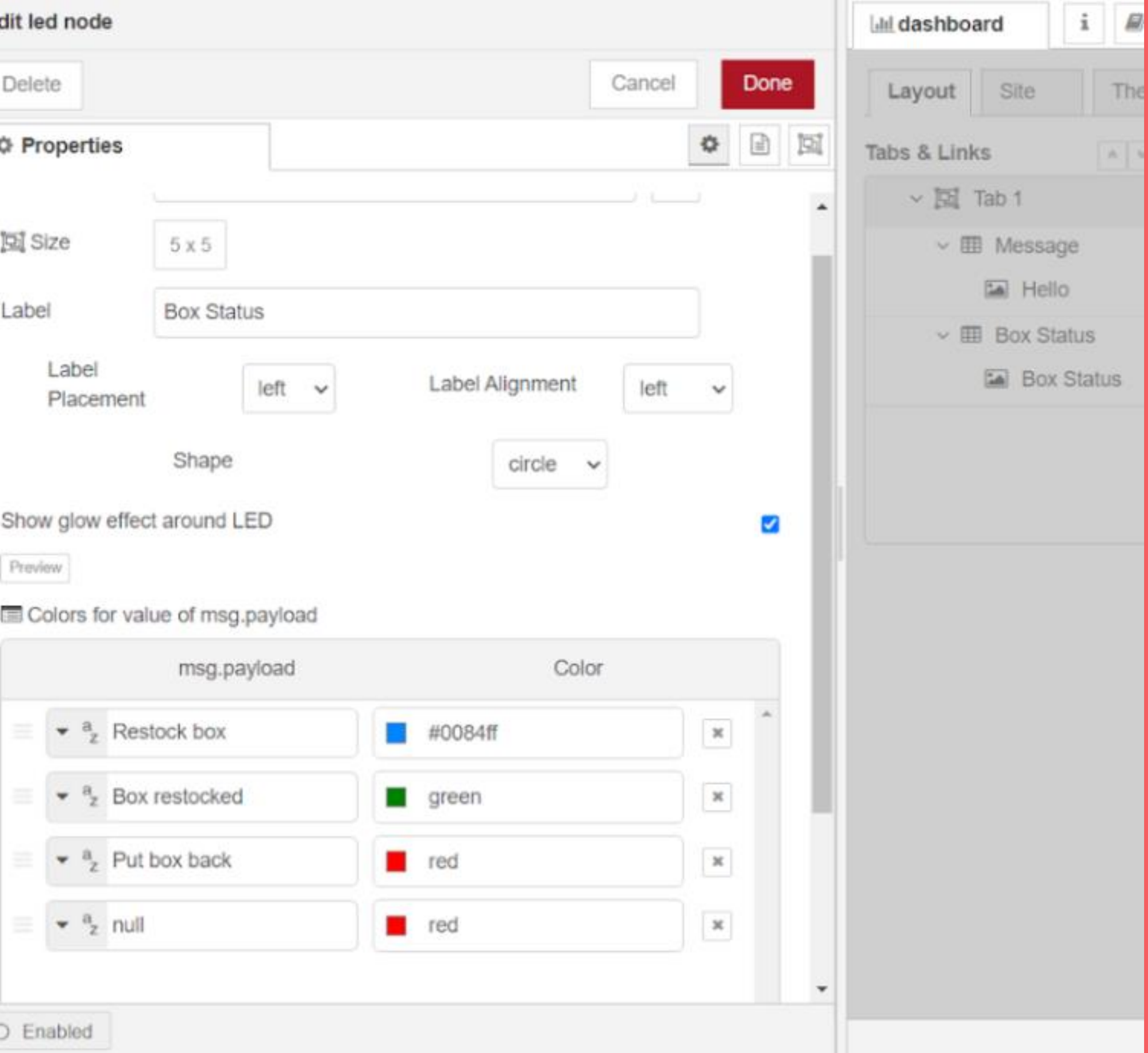
Split Node

Function Node



Name: function 1

```
}

return nodeRedFunction;
*/

var m =msg.payload.trim();
if(m=='Box 0 Restocked'){
    msg.payload="Box restocked";
    return msg;
}else if(m=='Restock Box 0'){
    msg.payload="Restock box";
    return msg;

}else{
    msg.payload="Put box back";
    return msg;
}
```



Size 5 x 5

Label Box Status

Label Placement left

Label Alignment left

Shape circle

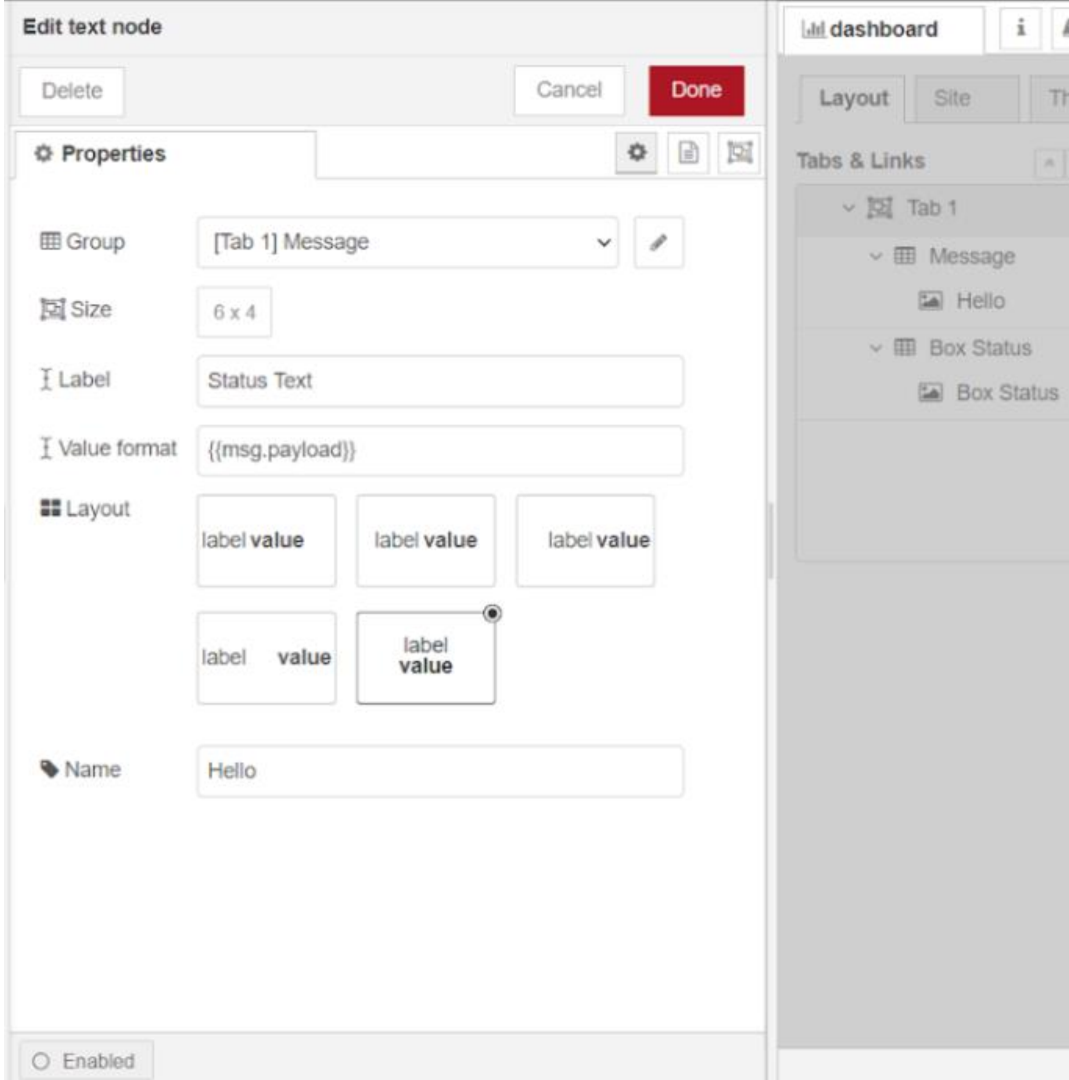
Show glow effect around LED ☑

Colors for value of msg.payload

msg.payload	Color
Restock box	🟦 #0084ff
Box restocked	🟩 green
Put box back	🟥 red
null	🟥 red

LED Node

Text Node



Edit text node

Group: [Tab 1] Message

Size: 6 x 4

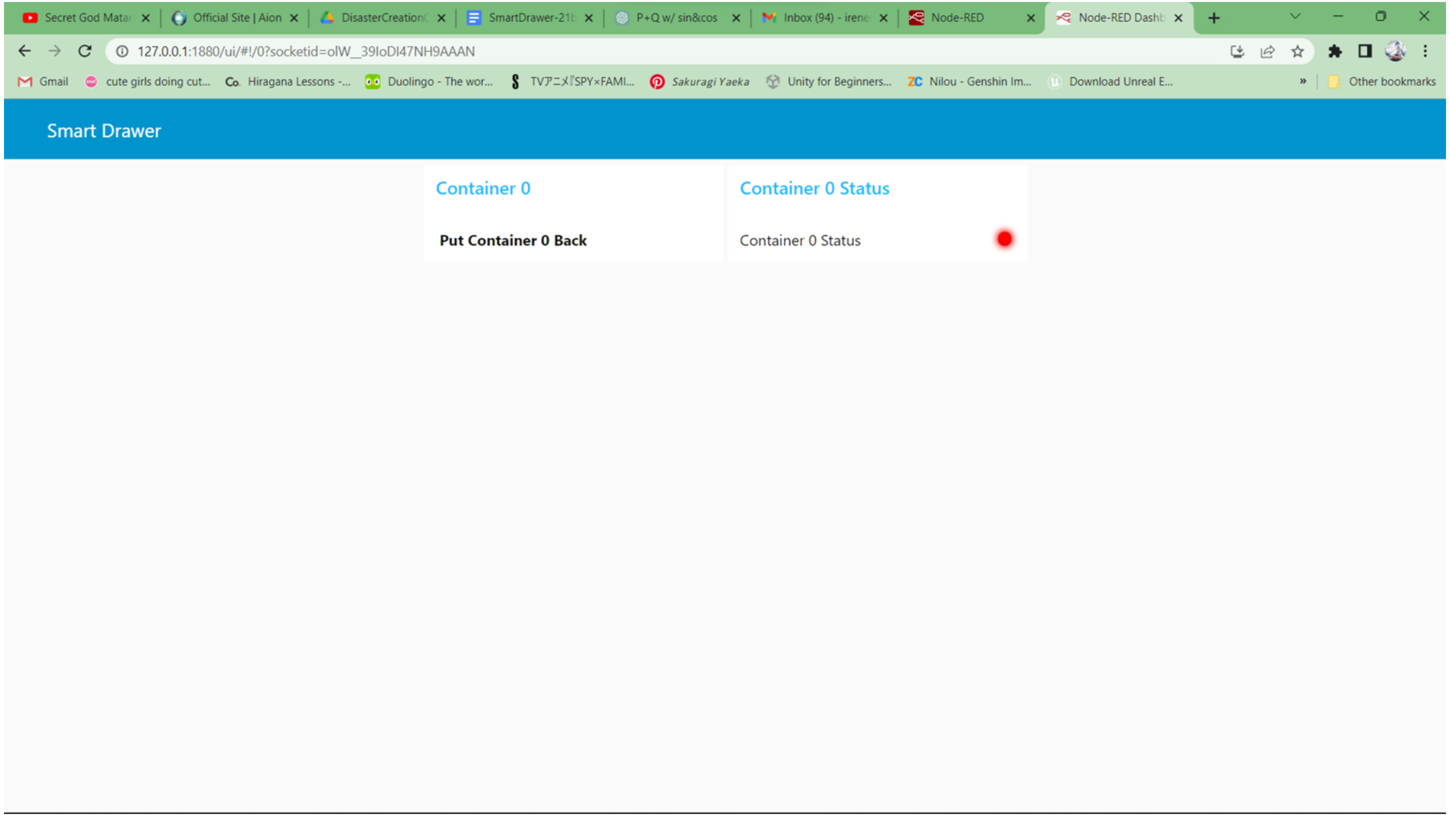
Label: Status Text

Value format: {{msg.payload}}

Layout: label value

Name: Hello

Case 1



Smart Drawer

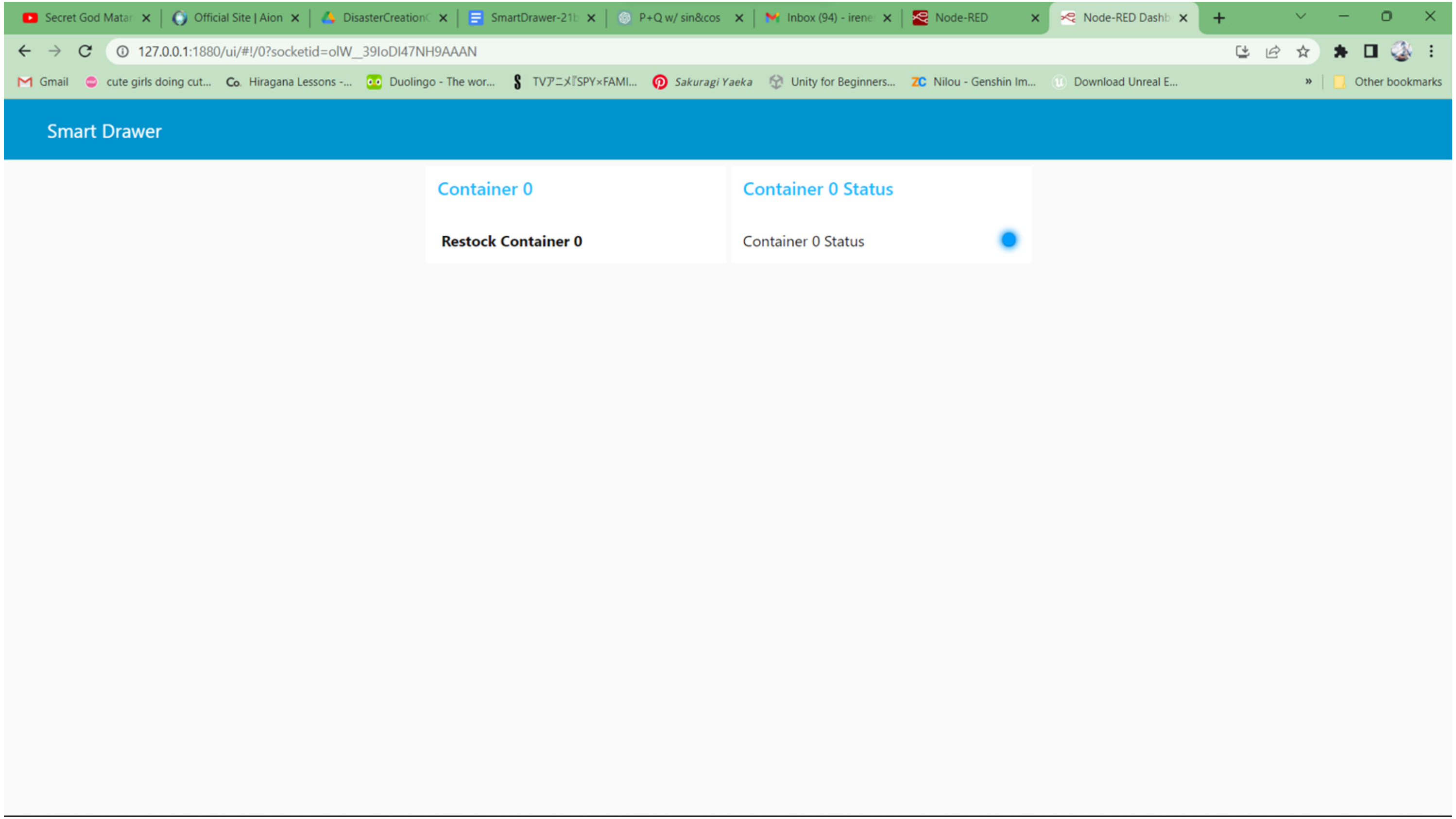
Container 0

Put Container 0 Back

Container 0 Status

Container 0 Status ●

Case 2



Smart Drawer

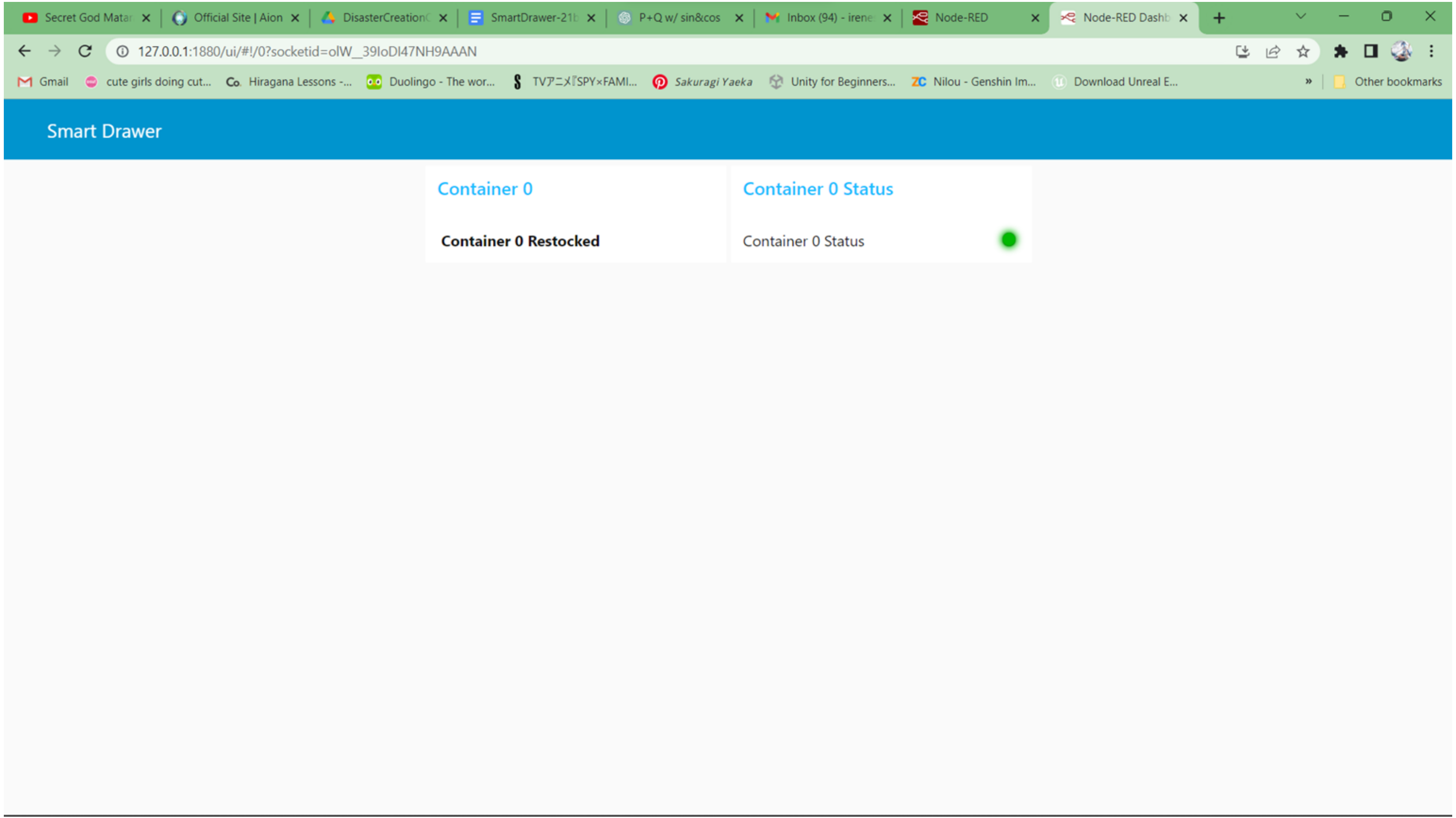
Container 0

Restock Container 0

Container 0 Status

Container 0 Status ●

Case 3



Smart Drawer

Container 0

Container 0 Restocked

Container 0 Status

Container 0 Status ●

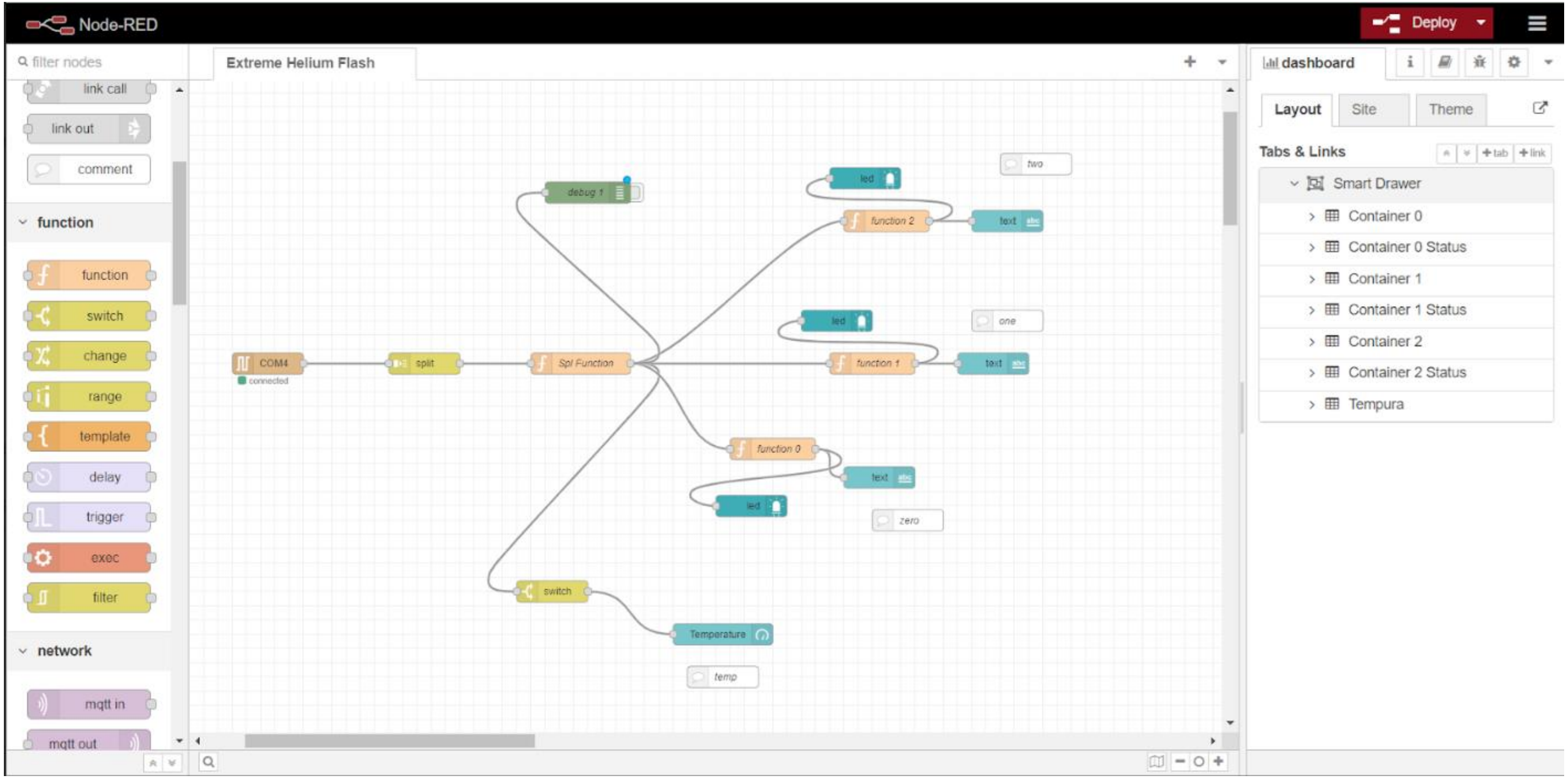
FINAL PROJECT

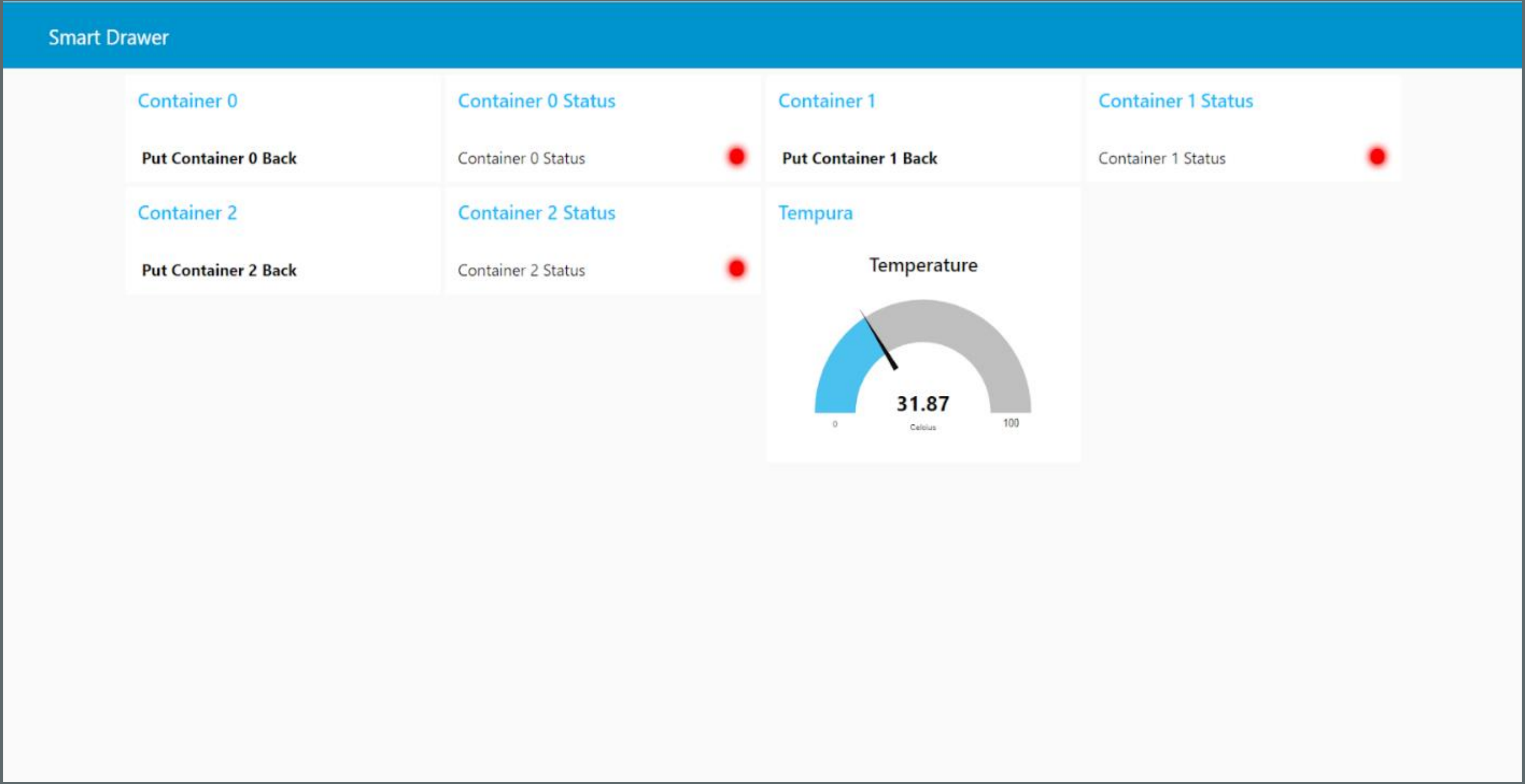
Final Project

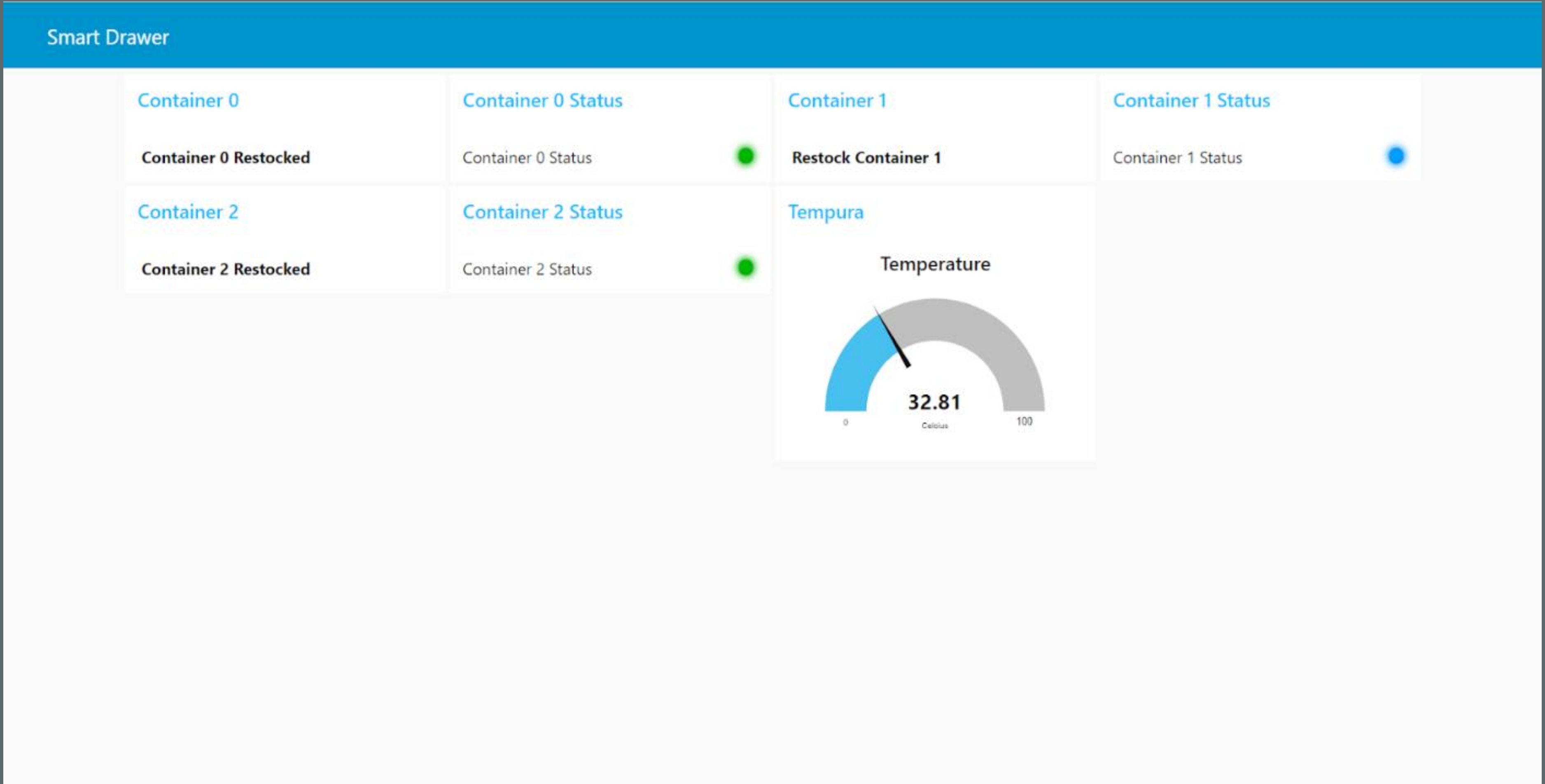
Apparatus used: 3 force sensors, 3 10k resistors and one Arduino Uno board

There are three compartments, in each compartment one force sensor is placed in. Along with the additional compartments/sensors, we have added a temperature sensor to sense the temperature of the environment of the Smart Drawer.

Node-Red


Node-RED
Deploy
filter nodes
link call
link out
comment
function
function
switch
change
range
template
delay
trigger
exec
filter
network
mqtt in
mqtt out
Extreme Helium Flash
COM4
connected
split
Spl Function
debug 1
led
function 2
text
two
led
one
function 1
text
function 0
text
led
zero
switch
Temperature
temp
dashboard
Layout
Site
Theme
Tabs & Links
+tab
+link
Smart Drawer
Container 0
Container 0 Status
Container 1
Container 1 Status
Container 2
Container 2 Status
Tempura


Smart Drawer
Container 0
Put Container 0 Back
Container 0 Status
Container 0 Status ●
Container 1
Put Container 1 Back
Container 1 Status
Container 1 Status ●
Container 2
Put Container 2 Back
Container 2 Status
Container 2 Status ●
Tempura
Temperature
31.87
0
Celsius
100



Smart Drawer

Container 0

Container 0 Restocked

Container 0 Status

Container 0 Status ●

Container 1

Restock Container 1

Container 1 Status

Container 1 Status ●

Container 2

Container 2 Restocked

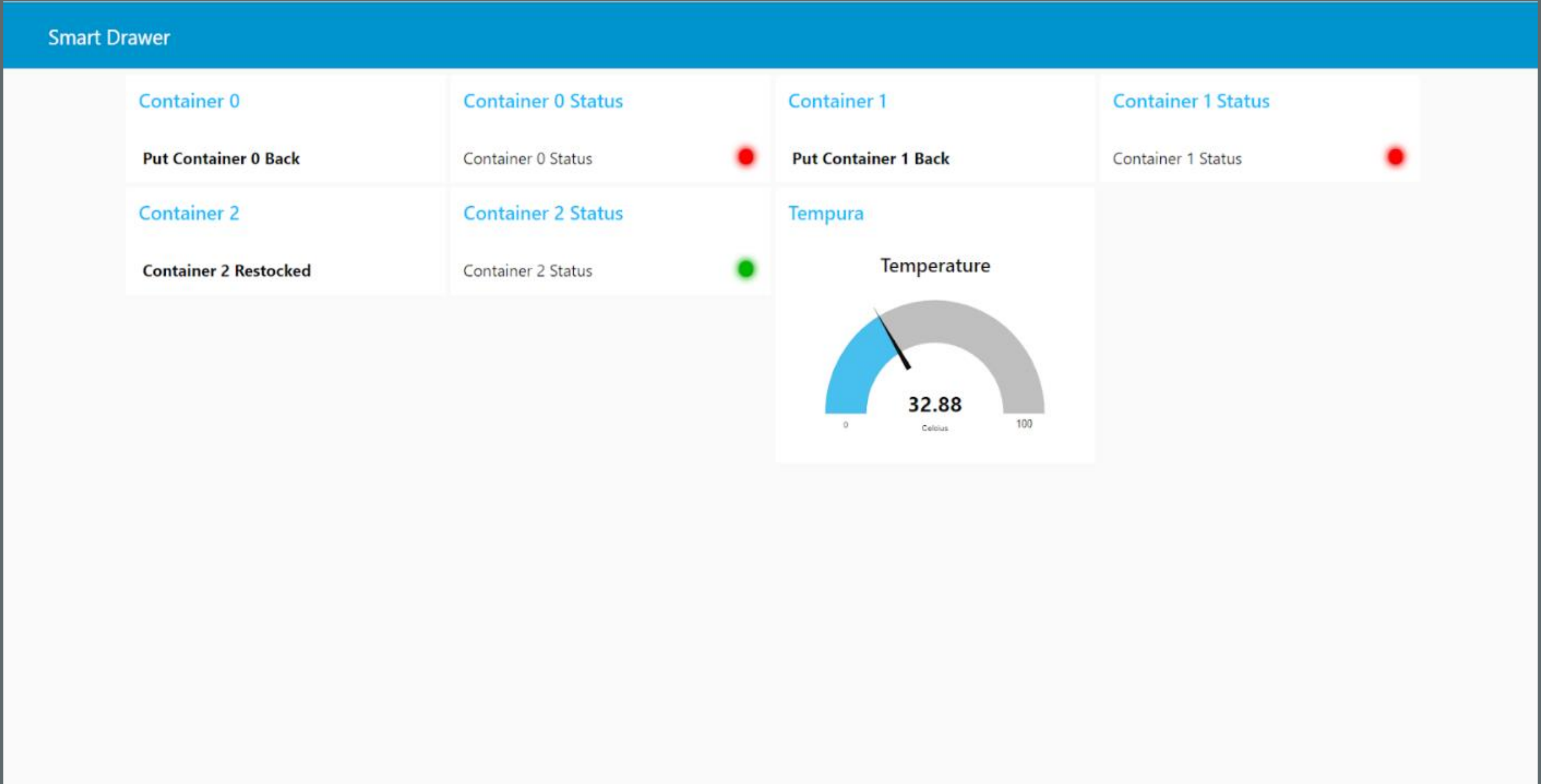
Container 2 Status

Container 2 Status ●

Tempura

Temperature

32.81



Smart Drawer

Container 0

Put Container 0 Back

Container 0 Status

Container 0 Status ●

Container 1

Put Container 1 Back

Container 1 Status

Container 1 Status ●

Container 2

Container 2 Restocked

Container 2 Status

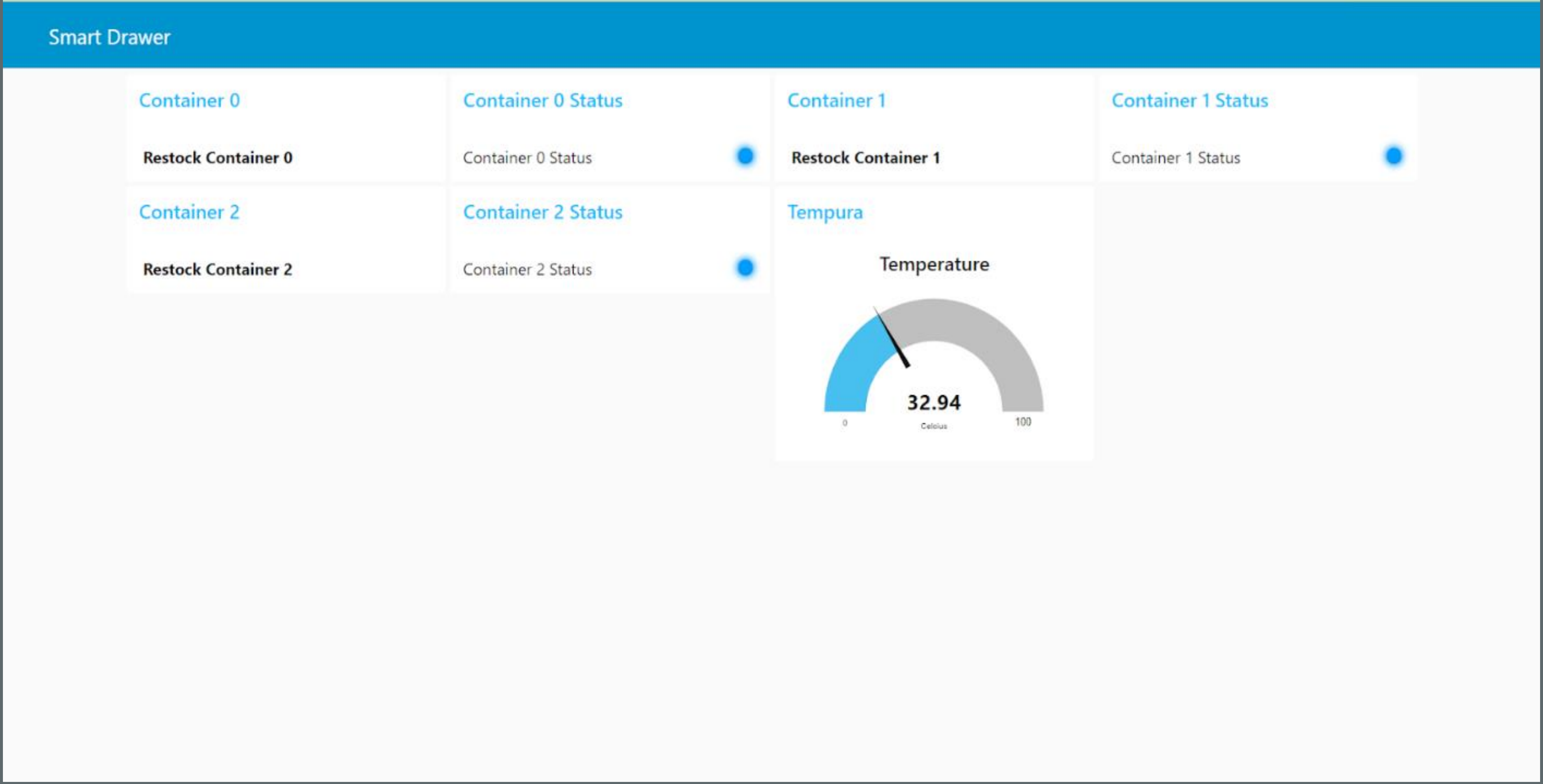
Container 2 Status ●

Tempura

Temperature

32.88

Celcius



Smart Drawer

Container 0

Restock Container 0

Container 0 Status

Container 0 Status ●

Container 1

Restock Container 1

Container 1 Status

Container 1 Status ●

Container 2

Restock Container 2

Container 2 Status

Container 2 Status ●

Tempura

Temperature

Discussion

By incorporating force sensors in each of its three compartments, the Smart Drawer can accurately detect the weight of containers placed inside. When a container reaches a weight above the predefined threshold value, the system notifies the user that the compartment is restocked, indicated by a green LED.

Conversely, if the weight of a container falls below the threshold value or if it is removed from the compartment, the force sensor detects this change and triggers a message to the user, prompting them to put the object back. In this scenario, the LED turns blue, indicating the need for restocking. If the container remains unattended for a certain time delay, the system intensifies the message to the user, urging them to restock. The LED color changes to red to signify the criticality of the situation.

Additionally, the Smart Drawer also features a temperature recording function, allowing the user to monitor the environmental conditions in which the drawer is placed. This information can be valuable for storing temperature-sensitive items or ensuring appropriate storage conditions.

Conclusion

In conclusion, the Smart Drawer project offers a convenient and efficient solution for managing inventory and ensuring timely restocking. The Smart Drawer project combines the use of force sensors, LED indicators, and temperature monitoring to provide an efficient and user-friendly system for inventory management and organization. It helps users stay informed about the stock levels, promotes timely restocking, and ensures the preservation of temperature-sensitive items. This Smart Drawer also accounts for the time it would take for the user to use the container and keep it back. This time delay can be changed according to the preference of the user. It can be personalised for different users and situations.

Thank you