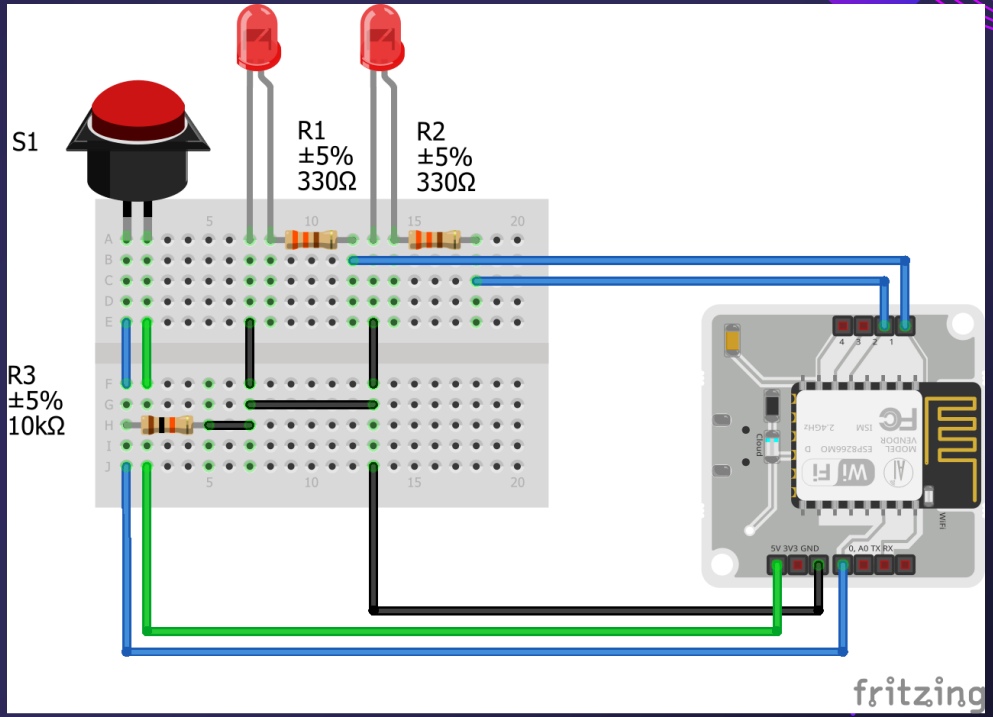
# Download any text editor (e.g. Sublime or Notepad ++) And be Ready with the below Circuit

If you have only one 330Ω resistor with you, no issue. Connect one LED.

Make sure to connect the signal along with a 10K resistor to the ground within the board. To reduce wiring errors.



# Custom Dashboard for Bolt IoT Products

PRESENTED BY AKSHAYAN SINHA

BOLT

# About Me

Education – 3rd Year student pursuing B. Tech in ECE from Lovely Professional University.

Languages – C/C++, Python, Kotlin, Javascript, PHP, Lua, MALTAB, BASH, HTML & CSS.

Skills – FrontEnd Web Dev, IoT, Automation, Robotics, Cloud Computing, XR (AR+VR+MR), Post-Production of Videos (Edit, VFX)

akshayan.sinha@gmail.com | /akshayansinha

# What You'll Learn Today

1. Basics of HTML, CSS and JAVASCRIPT
2. Syntax of a Button (PRACTICAL)
   - Use HTML to create
   - CSS to style and position
   - Javascript to program it
3. How Bolt module functions with API
4. Using Web Storage function
5. Analog Input -> Custom Gauge Chart
6. Importance of UI-UX when making a project
7. Overview of Floor Monitoring Dashboard
8. Python script to control Device without *boltiot* package

# Three Layers of Web Development

01

## Content

HTML is the basic element of webpage. The content in the form of text that is displayed, and its structure comes here.

02

## Presentation

CSS takes care of the styling of the page. It could be for a text, or animation with any object.

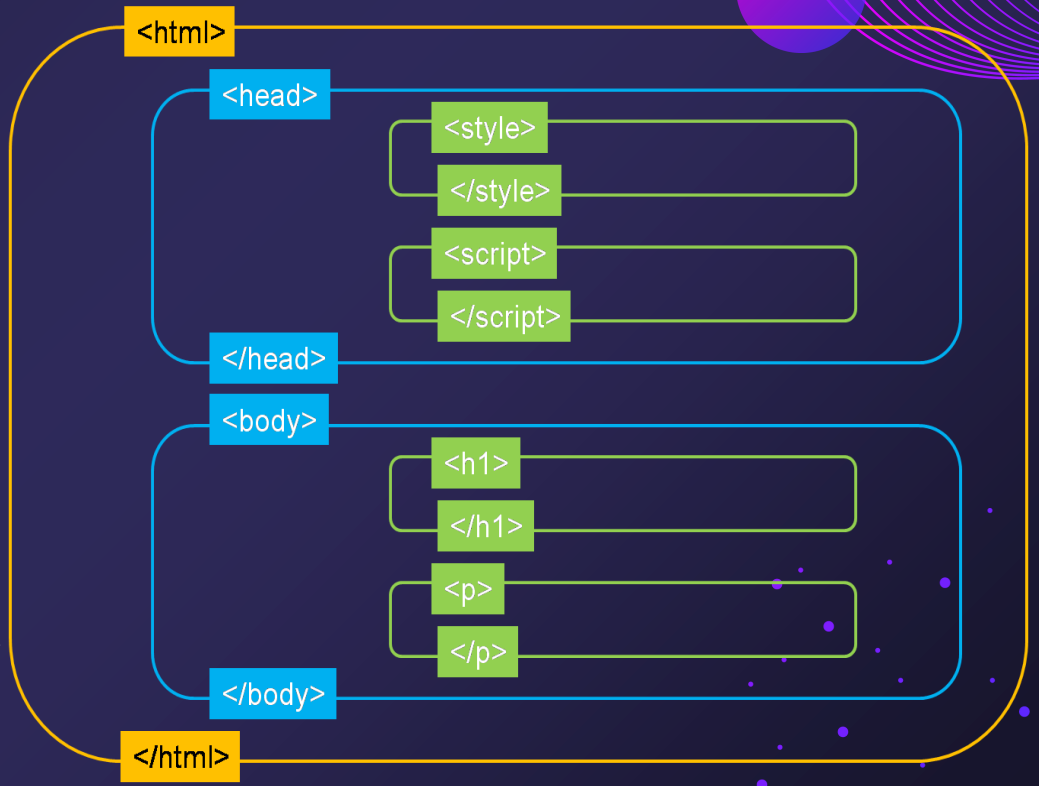
03

## Behaviour

JAVASCRIPT is the element that makes the page responsive. You can program and write functions to control the HTML as well as CSS

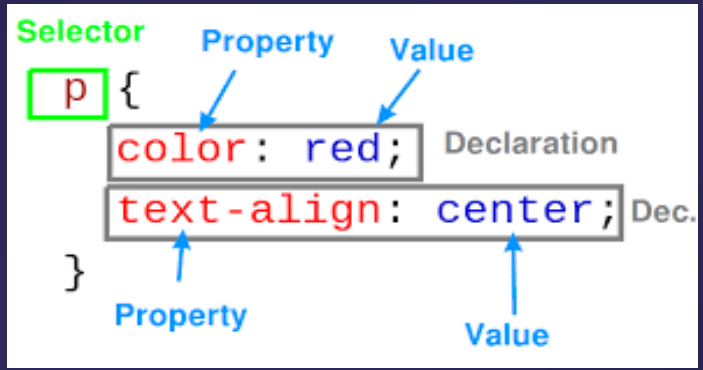
# Sample of .HTML file

```
<!DOCTYPE html>
<html>
  <head>

    //Title of the Page
    <title>This is the Title</title>

    // Connecting design of the page (CSS)
    <link rel="stylesheet" href="Location of the file.css">

    // Connecting functions that the HTML will do
    <script type="text/javascript" src = "Location of the file.js">
    </script>

  </head>

  <body>
    <center>
      <div id ='CONTENT' onClick = "change1()">
        <h1 class="BOX" onClick = "change2()"> INSIDE A BOX</h1>
      </div>
    </center>
  </body>
</html>
```

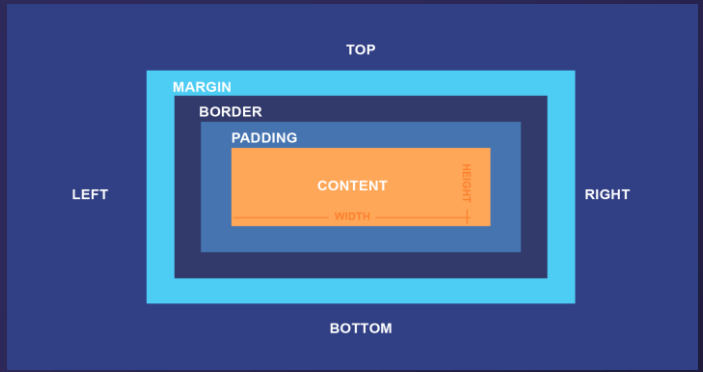


# Sample of .CSS file

```
body{
  background-color:  #183059;
}

h1{
 color: #AF5B5B;
}

#CONTENT{
   margin-top: 50px;
   padding:100px;
   background-color: #276FBF;
}

.BOX{
   width:500px;
   height:80px;
   line-height:80px;
   background-color: #F6F4F3;
}
```



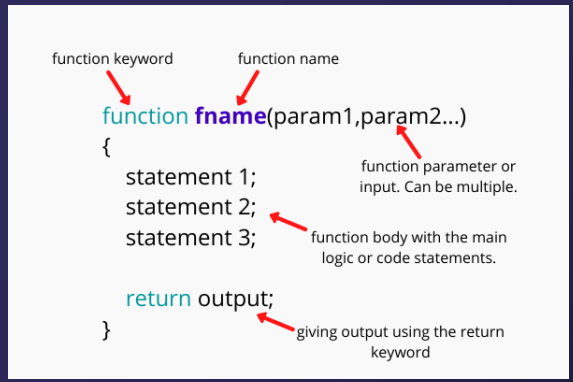

SYNTAX




BOX MODEL

# Sample of JAVASCRIPT (.JS) file

```
var flag1;
var flag2;

function change1(){
 if (flag1 =="0"){
  document.getElementById("CONTENT").style.backgroundColor = "pink";
  flag1="1";
 }
 else{
  document.getElementById("CONTENT").style.backgroundColor = "#276FBF";
  flag1="0";
 }
}

function change2(){
 if (flag2 =="0"){
  document.getElementById("CONTENT").getElementsByClassName("BOX")[0].style.backgroundColor = "lightgreen";
  flag2="1";
 }
 else{
  document.getElementById("CONTENT").getElementsByClassName("BOX")[0].style.backgroundColor = "white";
  flag2="0";
 }
}
```



# Test the Sample on CodePen

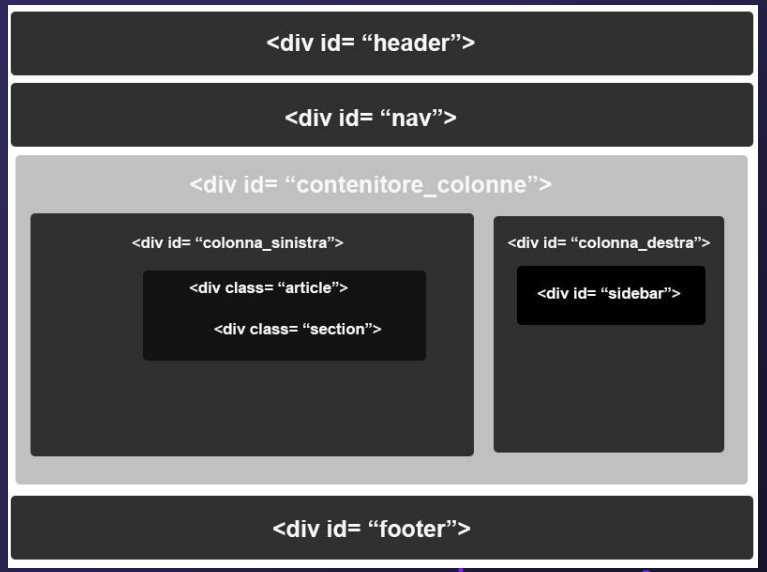
- Visit - https://codepen.io/hippyaki/pen/abWBWbM
- Shortened URL - cutt.ly/tmAdx9u

Or

- Say OK Google > "Scan QR Code" > Scan the QR →
- Learning platforms for Web Dev – w3schools, freecodecamp, github.com/topics
- For Doubts – stackoverflow (Just google the error you see on the console)
- Be in touch with me for Free Udemy Courses.

# What's happening?

- The elements were defined as 'ID' attribute. An ID is like an unique identity for an element in the html document. Each ID we create, is unique.
- An ID when created, creates a unique block for itself and can only apply to at most one element, while "class" selector can apply to multiple elements.
- 'ID' attributes can be styled using CSS. And the content(HTML) it contains inside it, is also unique.
- Example - `<h1 id="Header">Hello World</h1>`
- We can have multiple classes while we apply same stylings. And everywhere in the page with that class will avail it.
- But we can have only one styling for each ID.



- With DOM (Document Object Model), JavaScript can access and change all the elements of an HTML document. Including styling for CSS.
- `document.getElementById("CONTENT").style.backgroundColor = "#276FBF";`
- The above example changes background color of the element with ID = “CONTENT”. Even the text within the tag can be changed.

**↓ CHALLENGE ↓**

- Use `document.getElementById(<look in html file>).innerHTML = “ABCD”;` to change “INSIDE A BOX” -> “ABCD” - hint change ‘class’ to ‘id’ first

# Choose Any of Two (POLL)

OR

## Send the files

Pro - Easy and Time efficient.
Con – Few participant may download and ignore

1

## Type as you Watch

Pro – Better understanding
Con – Time Consuming and Errors for beginners

2

# Download the Script – from GitHub

Link - https://github.com/hippyaki/BoltIoT-Workshop-Scripts

Shortened URL - cutt.ly/AmAdhpy

Read the Instructions –

1 -> See for a 

button on top right of the repository. Click on that.

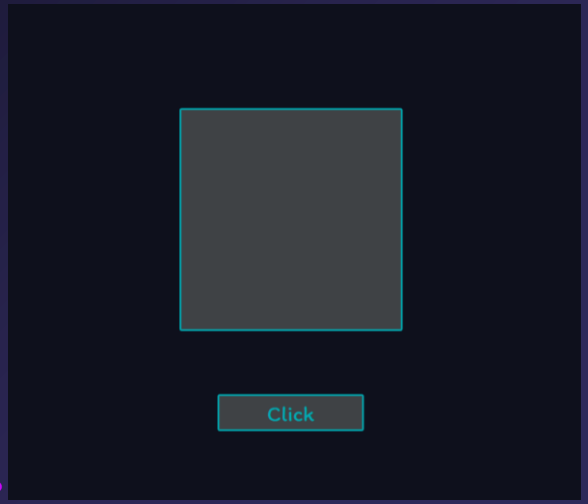
2 -> Download ZIP file

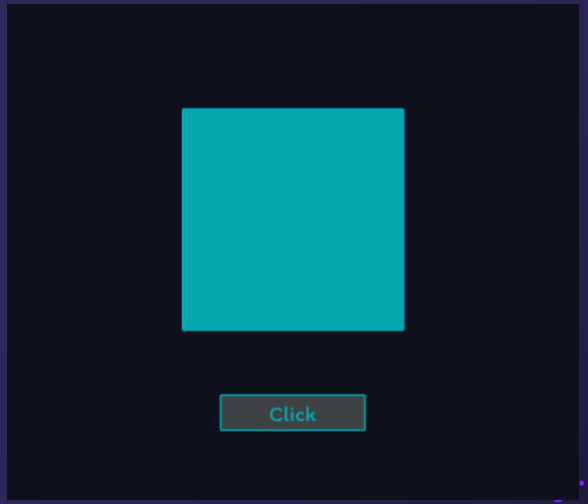
3 -> Go to Downloads Folder > Extract the folder in any location

4 -> Every folder (except Basic Button) along with index.html, contains a JS file – control.js.

5 -> Open control.js in an Editor > change api_key and device_id with your credentials.

# Button Interaction (Custom)



# Basic Button – Folder

- Two Elements – A “WINDOW”  and “BUTTON”.
- Since, we already changed styling(CSS) and content(HTML) in previous slides.
- Here, button will have an onclick function on it. When triggered, the background color of ID=“WINDOW” will change.
- For HTML, name of file = index.html

```
<body>
    <center>
        <div id="WINDOW"></div>
        <br>  <br>
        <button id="BUTTON" onclick="OnClick()">Click</button>
    </center>
    <script src="control.js"></script>
</body>
```

# For CSS, name of file = main.css

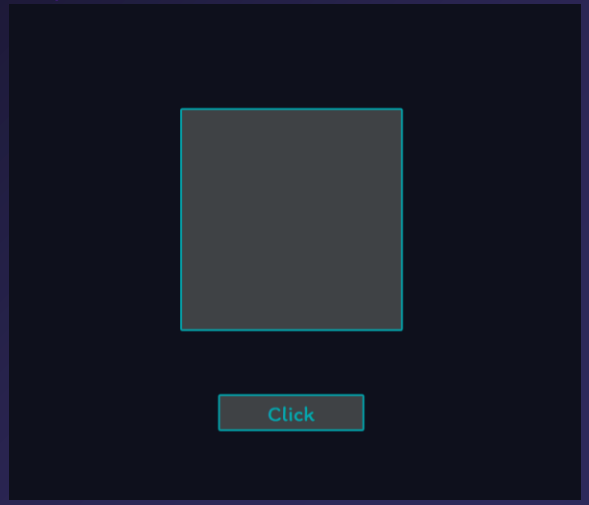
```
body{
    margin-top: 150px;
    background: #0E101C;
}
#WINDOW {
    height:  300px;
    width: 300px;
    line-height: 270%;

    background-color: #3f4245;
    border: 3px solid #00ADB5;
    border-radius: 3px;
    transition: 2s ease;
    color: #00ADB5;
    font-family:Itim,handwriting;
}
#WINDOW.on{
    transition: background-color 2s ease-out;
    background-color: #00ADB5;
    color: #5a5d61;
}
```

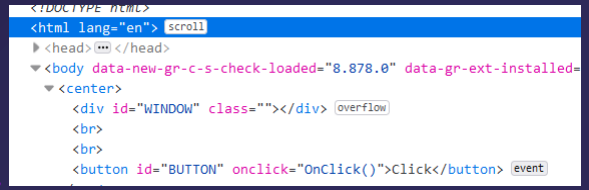
```
#BUTTON {
    height:  50px;
    width: 200px;
    line-height: 150%;

    margin-top: 50px;
    background-color: #3f4245;
    border: 3px solid #00ADB5;
    border-radius: 3px;
    transition: 0.3s ease;
    color: #00ADB5;
    font-size: 30px;
    font-family:Itim,handwriting;
}
#BUTTON:hover{
    transform: scale(1.05);
    transition: 0.3s ease;
    background-color:#505357;

}
```

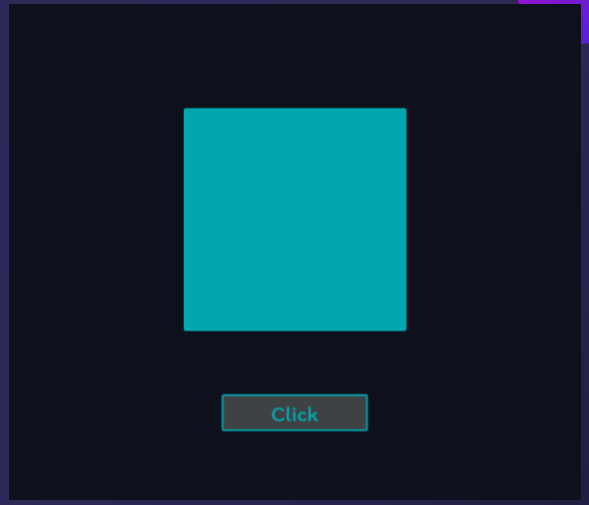
# For Javascript, name=control.js

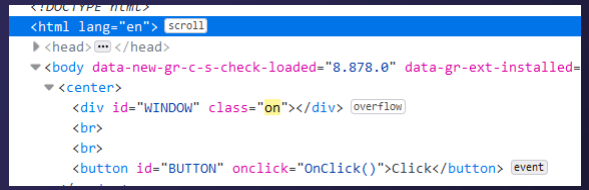
```
function OnClick(){
    var L1 = document.getElementById('WINDOW');
    if (L1.classList.contains("on")){
        document.getElementById("WINDOW").classList.remove("on");
        localStorage.removeItem("WINDOW");
    }
    else {
        document.getElementById("WINDOW").classList.add("on");
        window.localStorage.setItem("WINDOW", "on");
    }
}
function ReadButton(){
    if (window.localStorage.getItem('WINDOW') == 'on'){
        document.getElementById("WINDOW").classList.add("on");
    }
}

ReadButton();
```

# Open in Browser > Right click > Inspect

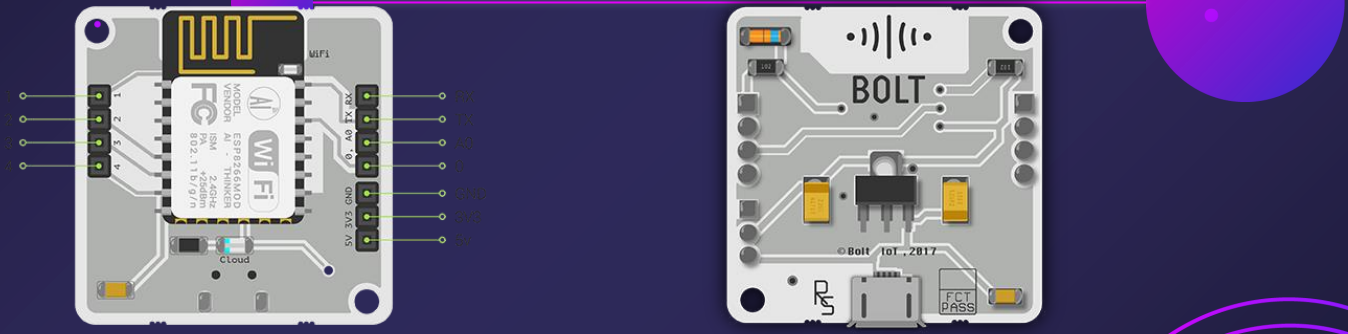


BOLT

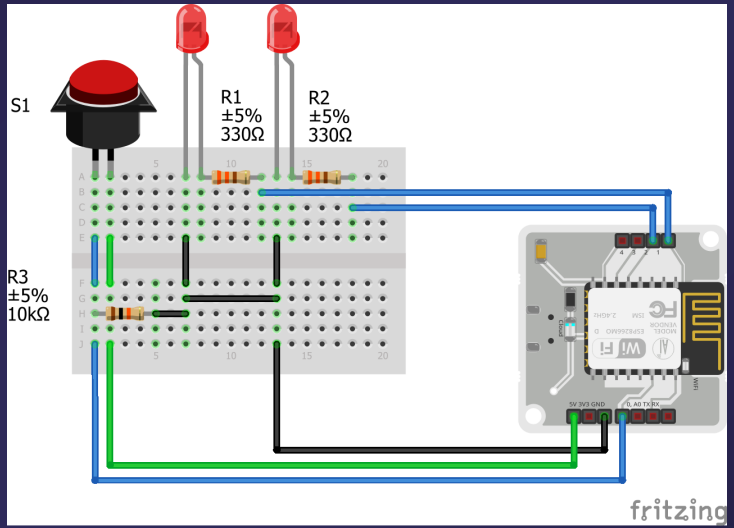



# Let‘s Connect Bolt Now !

# Are you ready with the circuit?

YES

1

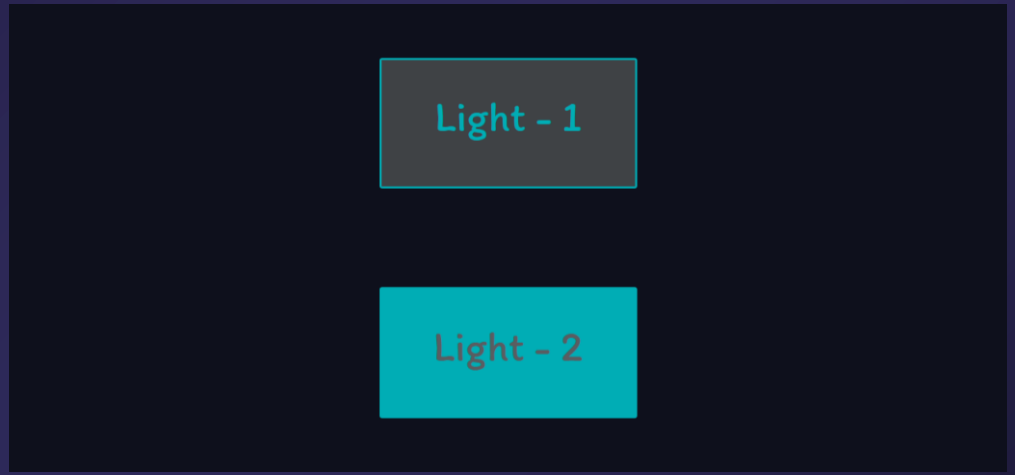



NO

2

# Write Function – Folder

- Two ID attributes – “L1” and “L2”  (Which refer as the LED of the circuit)
- Both elements work as buttons as well as windows.
- Status of the PINS visible in the button itself.



# For HTML, name of file = index.html

```
<body>
    <center>
        <div id="L1" onclick="L1()">Light - 1</div>
        <br>
        <br>
        <div id="L2" onclick="L2()">Light - 2</div>
    </center>
    <script src="control.js"></script>
</body>
```

# For CSS, name of file = main.css

```
body{
    margin-top: 150px;
    background: #0E101C;
    font-size: 50px;
}
#L1 {
    height:  150px;
    width: 300px;
    background-color: #3f4245;
    border: 3px solid #00ADB5;
    border-radius: 3px;
    transition: 0.3s ease;
    line-height: 270%;
    color: #00ADB5;
    font-family:Itim,handwriting;
}
#L1:hover{
    transform: scale(1.05);
    transition: 0.3s ease;
    background-color:#505357;
}
#L1.on{
    background-color: #00ADB5;
    color: #5a5d61;
}
#L2 {
    height:  150px;
    width: 300px;
    background-color: #3f4245;
    border: 3px solid #00ADB5;
    border-radius: 3px;
    transition: 0.3s ease;
    line-height: 270%;
    color: #00ADB5;
    font-family:Itim,handwriting;
}
#L2:hover{
    transform: scale(1.05);
    transition: 0.3s ease;
    background-color:#505357;
}

#L2.on{
    background-color: #00ADB5;
    color: #5a5d61;

}
```

# For Javascript, name = control.js

```
var debug = 0

var api_key = [redacted]
var d_name = [redacted]
var base_url = "https://cloud.boltiot.com/remote/"



function digitalWrite(pin,val){
    var xmlhttp = new XMLHttpRequest();
    xmlhttp.onreadystatechange = function() {
        if (xmlhttp.readyState == 4 && xmlhttp.status == 200 && debug == 1) {
            //document.getElementById("javascript_response").innerHTML = "Javascript Response : "+xmlhttp.responseText;
            alert(xmlhttp.responseText);
            var obj = JSON.parse(xmlhttp.responseText);
            if(obj.success=="1"){

            }
        }
    };
    xmlhttp.open("GET", base_url+api_key+"/digitalWrite?pin="+pin+"&state="+val+"&deviceName="+d_name,true);
    xmlhttp.send();
}
```

```
function ReadAll(){
    if (window.localStorage.getItem('L1') == 'on'){
        document.getElementById("L1").classList.add("on");
    }
    if (window.localStorage.getItem('L2') == 'on'){
        document.getElementById("L2").classList.add("on");
    }
}


ReadAll();
```

```
function L1(){

    var L1 = document.getElementById('L1');

    if (L1.classList.contains("on")){

        digitalWrite(1,'LOW');
        document.getElementById("L1").classList.remove("on");
        localStorage.removeItem("L1");
    }
    else {
        digitalWrite(1,'HIGH');
        document.getElementById("L1").classList.add("on");
        window.localStorage.setItem("L1", "on");
    }
}

function L2(){

    var L2 = document.getElementById('L2');

    if (L2.classList.contains("on")){
        digitalWrite(2,'LOW');
        document.getElementById("L2").classList.remove("on");
        localStorage.removeItem("L2");
    }
    else {
        digitalWrite(2,'HIGH');
        document.getElementById("L2").classList.add("on");
        window.localStorage.setItem("L2", "on");
    }
}
```

# Doubt with this Function?

**“GET” is used to request data from a specified resource.**

```
var debug = 0

var api_key = "fd669c56-5e8f-4b69-9bed-29319e69df7b";
var d_name = "BOLT8021328";
var base_url = "https://cloud.boltiot.com/remote/"



function digitalWrite(pin,val){
    var xmlhttp = new XMLHttpRequest();
    xmlhttp.onreadystatechange = function() {
        if (xmlhttp.readyState == 4 && xmlhttp.status == 200 && debug == 1) {
            //document.getElementById("javascript_response").innerHTML = "Javascript Response : "+xmlhttp.responseText;
            alert(xmlhttp.responseText);
            var obj = JSON.parse(xmlhttp.responseText);
            if(obj.success=="1"){

            }
        }
    };
    xmlhttp.open("GET", base_url+api_key+"/digitalWrite?pin="+pin+"&state="+val+"&deviceName="+d_name,true);
    xmlhttp.send();
}
```

https://cloud.boltiot.com/remote/{API-KEY}/digitalWrite?pin=0&state=LOW&deviceName={DEVICE-ID}

base_url

# What does this API URL return?

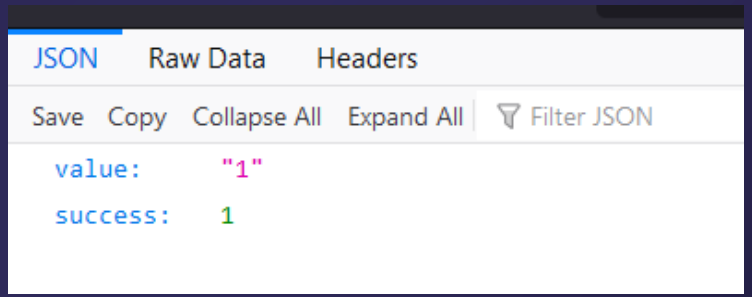
https://cloud.boltiot.com/remote/{API-KEY}/digitalWrite?pin=0&state=LOW&deviceName={DEVICE-ID}

base_url

What it returns, is a JSON data from bolt cloud.

RAW DATA - {"value": “1", "success": 1}

JSON Format -

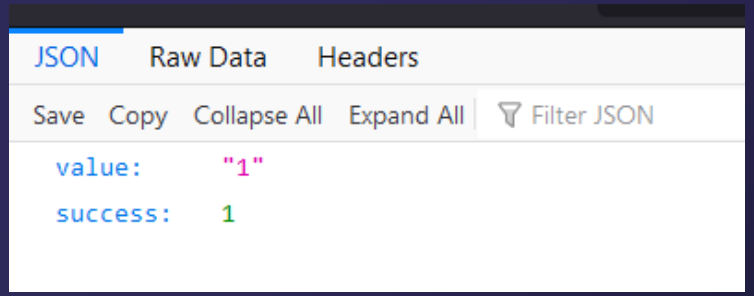


# What does digitalRead return?

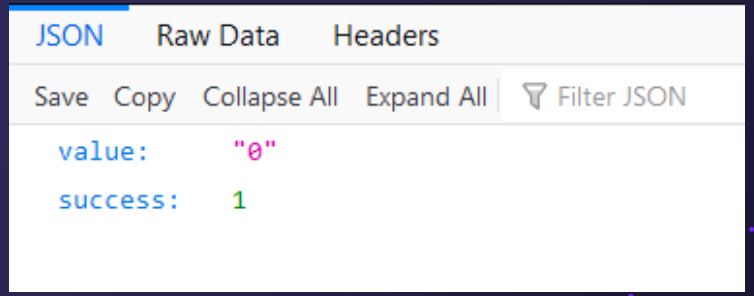
https://cloud.boltiot.com/remote/{API-KEY}/digitalRead?pin=0&deviceName={DEVICE-ID}

base_url

When PIN reads a signal HIGH –
(Current is following through the pin)

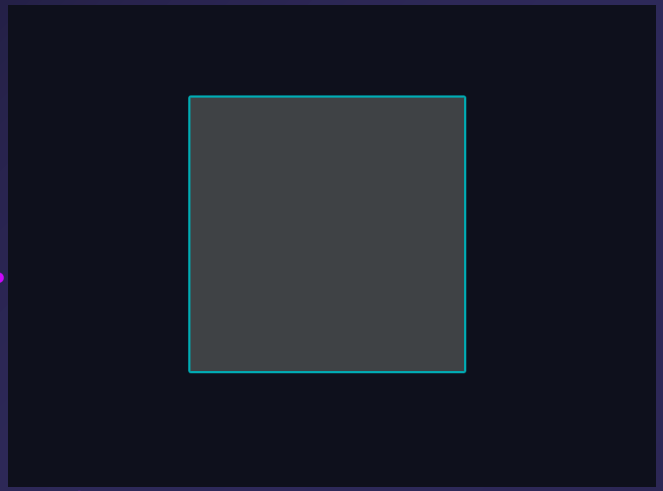


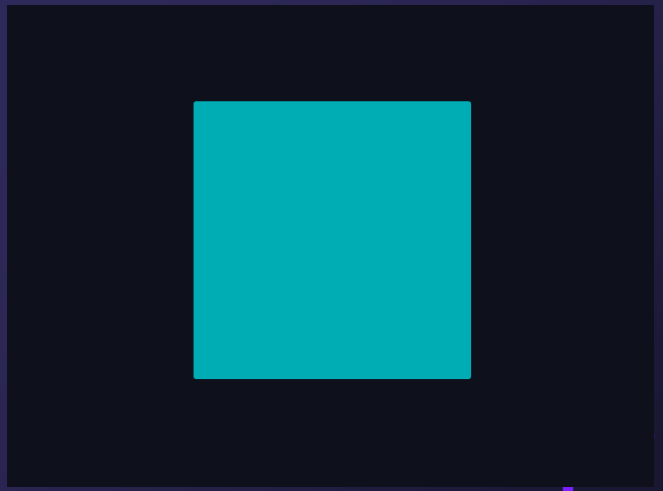

When PIN reads a signal LOW –
(Current is NOT following through the pin)



# Read Function – Folder (w Pro)

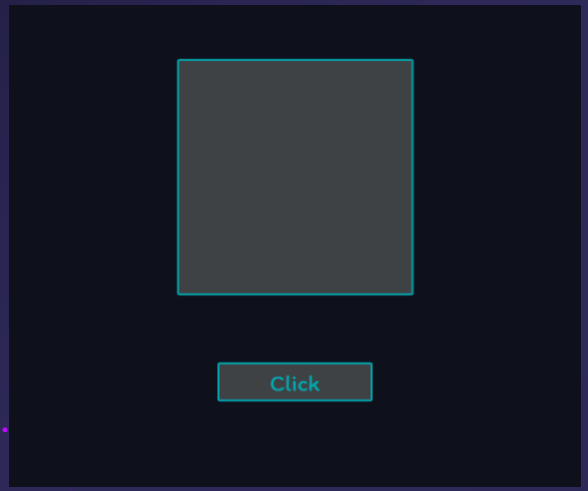
- Only one ID attribute “WINDOW’ to display status of pin. (Sends API req. every 0.5s)
- We used digitalRead API to check if the pin is HIGH or LOW
- Status of pin displayed on the “WINDOW” ID.

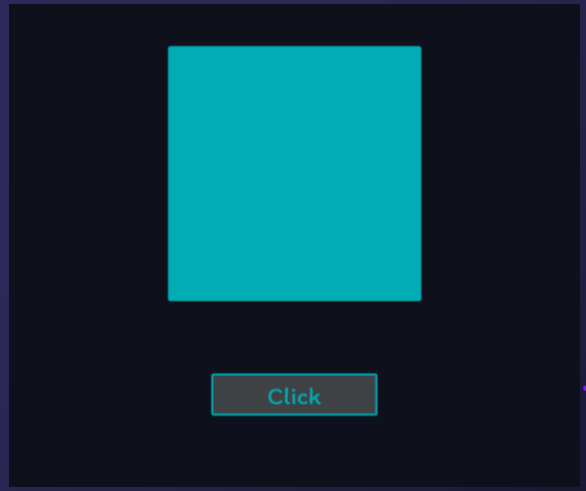




# Read Function – Folder (without Pro)

- Two ID attribute “WINDOW” to display status of pin and “BUTTON” to send the API request
- We used digitalRead to check if the pin is HIGH or LOW
- Status of pin displayed on the “WINDOW” ID.



# For HTML, name of file = index.html

- HTML syntax for PRO users

```
<div id="WINDOW"></div>
```

- HTML syntax for non-PRO users

```
<div id="WINDOW"></div>
        <br>
        <br>
<button id="BUTTON" onclick="OnClick()">Click</button>
```

# For JavaScript, name = control.js

```
function digitalRead(pin) {
    var xmlhttp = new XMLHttpRequest();
    xmlhttp.onreadystatechange = function() {

        //document.getElementById(element_id).innerHTML = xmlhttp.responseText;
        if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
          //  alert(xmlhttp.responseText);
            //document.getElementById("javascript_response").innerHTML = "Javascript Response : "+xmlhttp.responseText;
            var obj = JSON.parse(xmlhttp.responseText);
            if(obj.success=="1"){
                if(obj.value=="1"){
                    document.getElementById("WINDOW").classList.add("on");
                    window.sessionStorage.setItem("WINDOW", "on");
                }
                else if (obj.value=="0"){
                    document.getElementById("WINDOW").classList.remove("on");
                    sessionStorage.removeItem("WINDOW");
                }
            }
            else{
                    alert("Error = "+obj.value);
            }
        }
    };
    xmlhttp.open("GET",base_url+api_key+"/digitalRead?pin="+pin+"&deviceName="+d_name,true);
    xmlhttp.send();
}

function ReadAll(){
    if (window.sessionStorage.getItem('WINDOW') == 'on'){
        document.getElementById("WINDOW").classList.add("on");
    }
}
function OnClick(){
    digitalRead(0);
}
ReadAll();
```

# Steps to follow (for Non Pro users)

- Complete the circuit for the digitalRead using a SWITCH that comes with the KIT.
- Keep pressing the Switch, which sets the digitalRead value -> HIGH (“1”).
- While the switch is pressed, on the Webpage, click the BUTTON.
- The BUTTON will send an API request.
- If the switch was PRESSED i.e. digitalRead value -> HIGH (“1”), then your WINDOW color will change to BLUE.
- Release the switch i.e. digitalRead value -> LOW (“0”), click the BUTTON again.
- Color of the WINDOW will return to GREY.

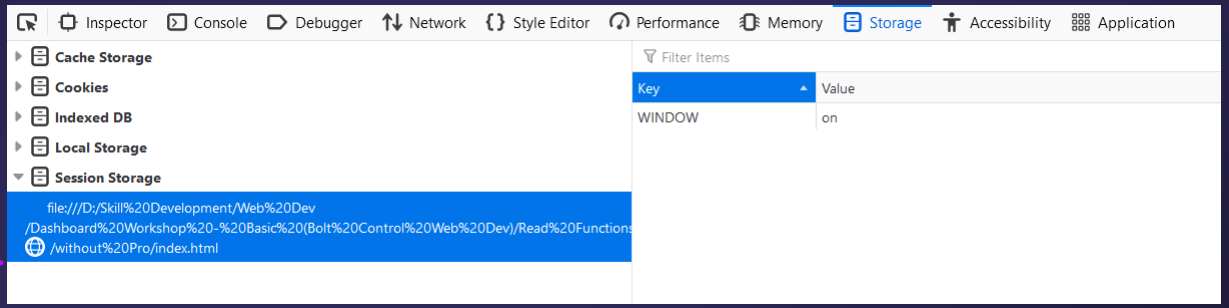
# Web Storage API

The Web Storage API is a simple syntax for storing and retrieving data in the browser.

For Mozilla - Right Click anywhere in browser > Inspect > Storage > Session / Local
For Chrome - Right Click anywhere in browser > Inspect > Application > Storage > Session / Local

- sessionStorage – It stores data for one session. The data is deleted when the browser/tab is closed. It can be refreshed, but not closed

- localStorage - The localStorage object provides access to a local storage for a particular Web Site. The data will not be deleted when the browser is closed.



# For more such functions -

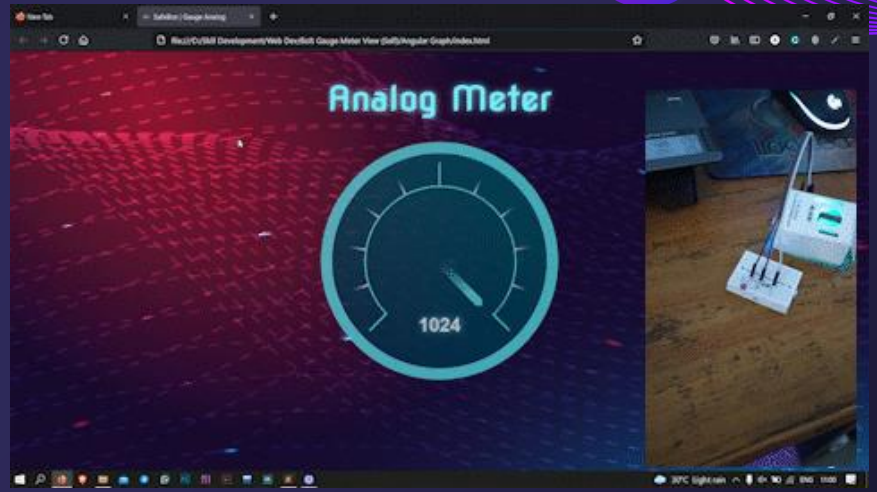
- Visit https://docs.boltiot.com/docs/introduction

  The above link will have the API URLs you can use to control the module

- Visit https://cloud.boltiot.com/static/js/boltCommands.js

  The above link will have ALL the existing functions similar to what we saw ALREADY pre-defined.

# Gauge meter using analogRead

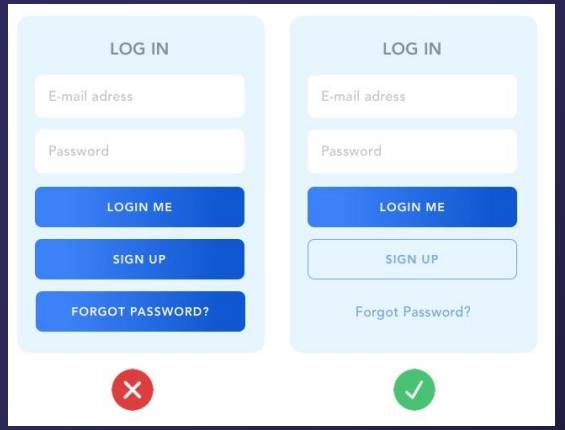

https://github.com/hippyaki/Bolt-IoT-custom-Graph-samples

# Importance of UI – UX

- Internet of Things (IoT) products in the commercial, industrial, or consumer space, has well thought out and nicely implemented hardware as essential parts of an IoT solution. But with optimized user interface (UI), it results in a very satisfying user experience leveraging IoT technology.
- Sensors, connectivity and product intelligence don't just create new data streams, they unlock a universe of possibilities and potential perils for user experiences and the organizations that design IoT user interfaces.
- It's bad enough when a tech-savvy user struggles to get the IoT solution up and running. It's unimaginable what the experience is like in the hands of a customer who isn't particularly tech- or IT-savvy. I suspect products often get thrown away in frustration.

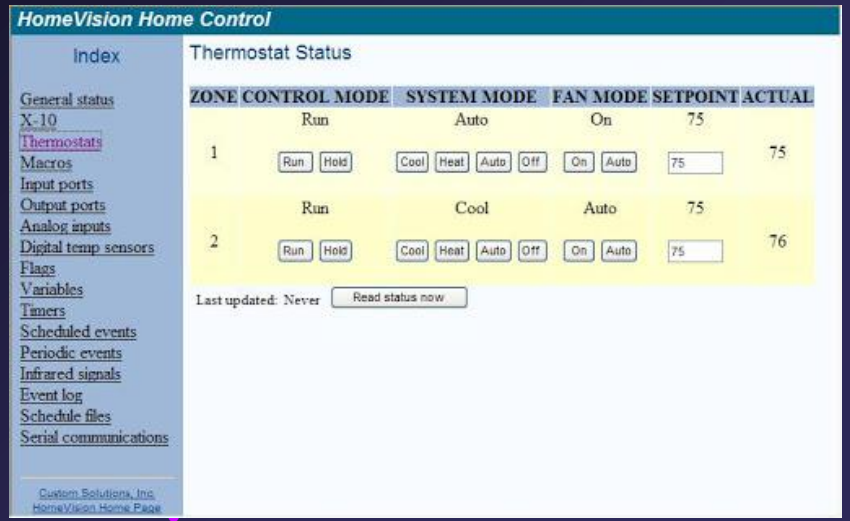
# Comparison of UI &UX


UX

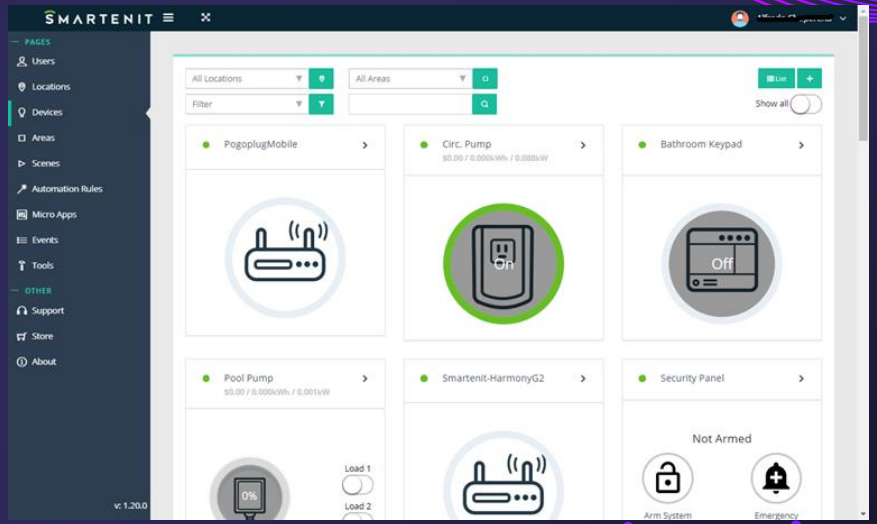



UI

# Which one do you prefer?



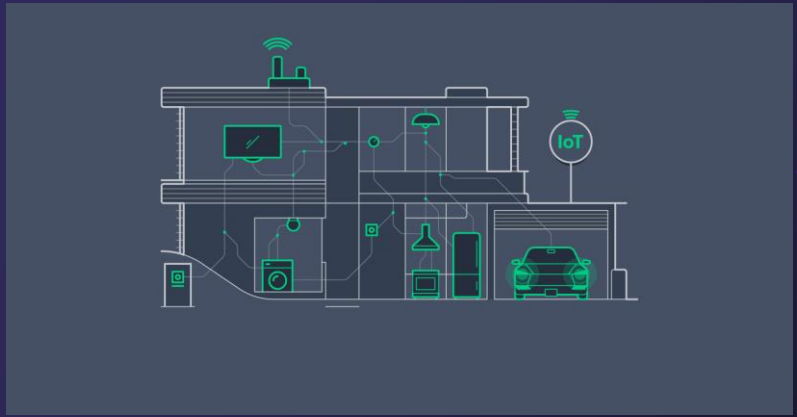

1



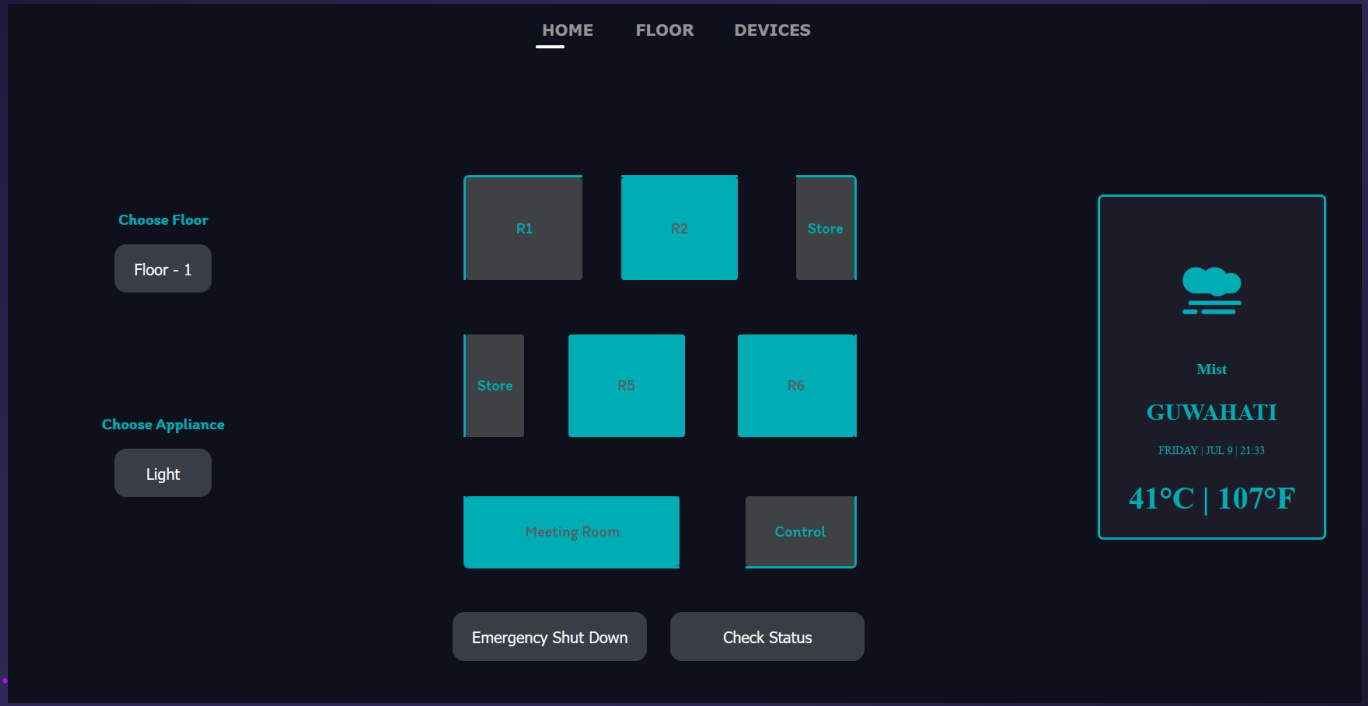

2

# Floor Monitoring Visual Samples



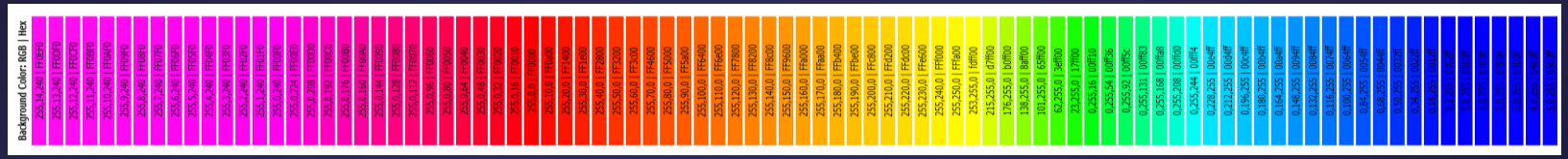



# Practical Demonstration



# Get the Dashboard and Customize it !

- HTML – 125 lines | JavaScript – 563 Lines | CSS – 760 lines
- But you get it all for FREE, with a catch.
- Use the syntax I have provided, build a color changing WINDOW, which changes color with temperature. Using 5 colors – Violet > Red > Yellow > Green > Blue




- Hint - Use analogRead, with LM35 sensor. Write the (if – else if) condition inside analogRead function's obj.success condition.
- analogRead returns values from 0-1024, which is the obj.value.

- Submit the Assignment on - https://forms.gle/eGpC3hS1nVZ6vPf28

# GIVEAWAY – Python Script

There's more !!

I have more on my GitHub for you !

Python package 'boltiot' does not have digitalMultiWrite yet. So I made a custom program with which you can use the digitalMultiWrite function, using API only.

How to Get it?

Follow me on GitHub -> Take a Screenshot -> Attach the screenshot during Assignment submission

(You can attach only the screenshot incase you don't wish to do the assignment)

# Thank You !

Have any Doubt? Or Wish to Explore and Learn more?

Contact me at -

akshayan.sinha@gmail.com | LinkedIn – Akshayan Sinha

akshayansinha.netlify.app