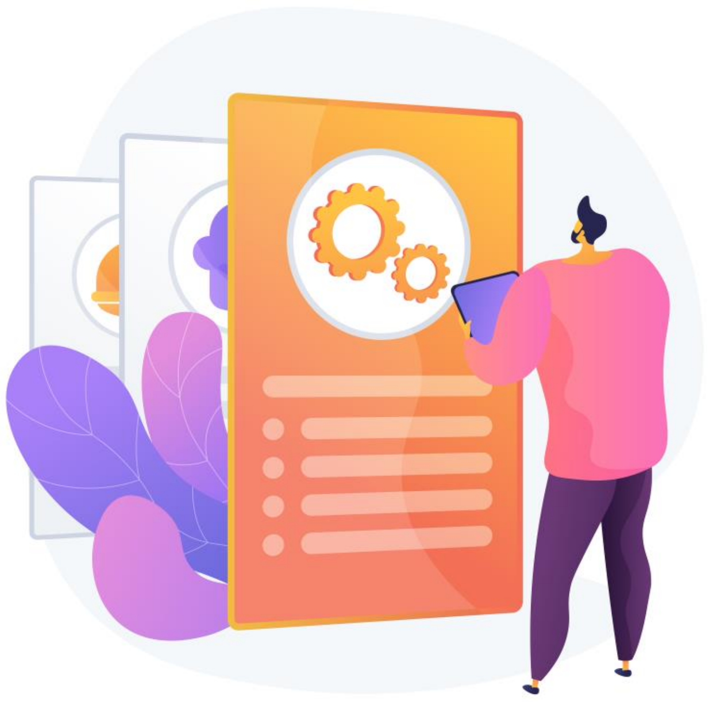


INTERACTIVE TIMELINE-MINDMAP

[TECHNICAL DOCUMENTATION]

FOR

Liz Motors

BY
Rahul Gupta

Table of Content

Setup

Web Technologies

1. HTML, CSS, and Javascript

Libraries

1. **jsMind**

   a) jsMind is mind map library built by Javascript, based on HTML5 canvas and SVG.
   b) jsMind is released under the BSD license and it is available to embed in any project that abides by its license.
   c) jsMind Content Delivery Network – Unpkg
   d) jsMind Version – 0.8.3

2. **jQuery**

   a) We are using jQuery to manage custom events and data handling, which is not available in the jsMind Library.
   b) jQuery Content Delivery Network - Google
   c) jQuery version - 3.7.1 / (August 28, 2023)

3. **Font Awesome**

   a) It is used to add custom icons to our mindmap, which is not available in the jsMind Library
   b) Font Awesome Content Delivery Network – Cloudfare
   c) Font Awesome version – 4.7.0

Compatibility

Operating Systems

1. Windows and MacOS
2. Android (version 11 to 14)
3. iOS (version 15 to 17)

Screen Resolutions

1. The mindmap is responsive and interactive on all screen sizes.
2. For touch screens of small sizes, it uses pinching out, drag, and tap features for accessibility.

GitHub Setup

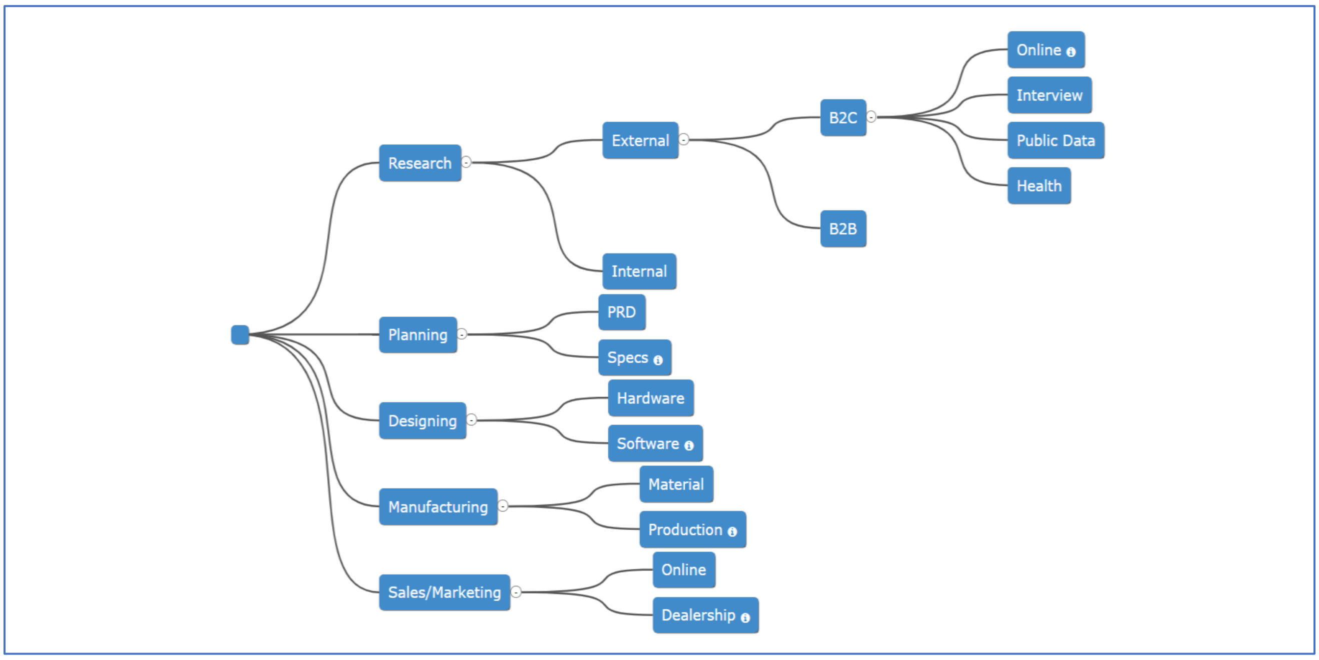
1. The repository is currently set to public - https://github.com/Rahul-dev2194/mindmap-timeline.git

A Look Inside

Mindmap Structure

1. jsMind offers three available formats to structure your mindmap –
   a) node_tree
   b) node_array
   c) freemind

2. We are using node_tree.



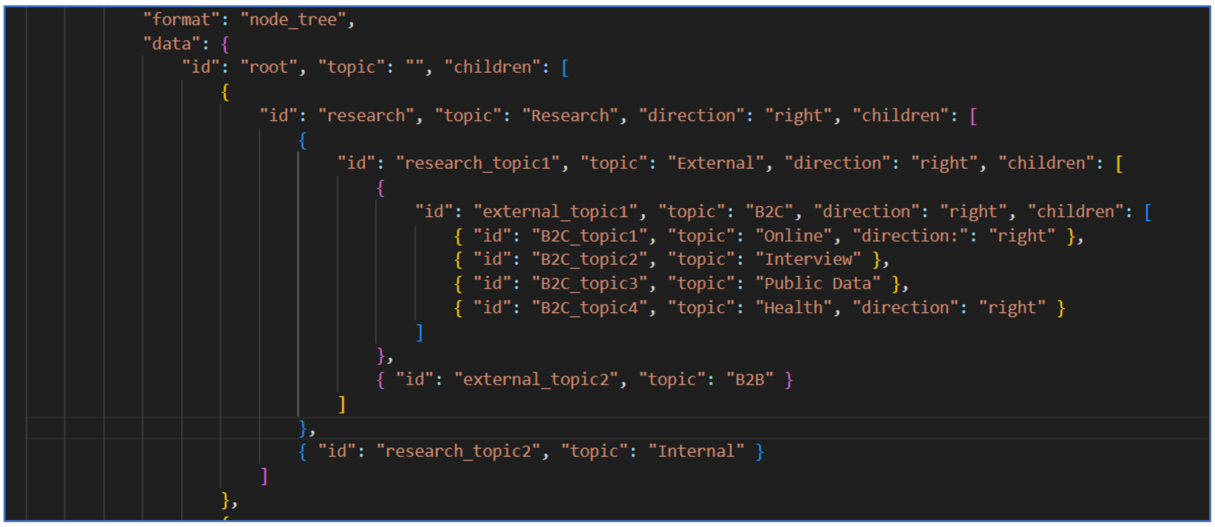
3. Please note we do have the feasibility to change the format of the mindmap whenever required.

How It Works

Node Creation and Rendering

1. Each segment in the mindmap is called a node and is rendered in the DOM as <jmnode> custom element.

2. Data inside each node is added through backend in the form of an object.

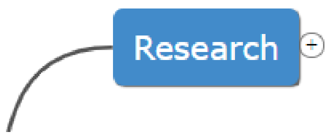


```
"format": "node_tree",
"data": {
    "id": "root", "topic": "", "children": [
        {
            "id": "research", "topic": "Research", "direction": "right", "children": [
                {
                    "id": "research_topic1", "topic": "External", "direction": "right", "children": [
                        {
                            "id": "external_topic1", "topic": "B2C", "direction": "right", "children": [
                                { "id": "B2C_topic1", "topic": "Online", "direction:": "right" },
                                { "id": "B2C_topic2", "topic": "Interview" },
                                { "id": "B2C_topic3", "topic": "Public Data" },
                                { "id": "B2C_topic4", "topic": "Health", "direction": "right" }
                            ]
                        },
                        { "id": "external_topic2", "topic": "B2B" }
                    ]
                },
                { "id": "research_topic2", "topic": "Internal" }
            ]
        },
```

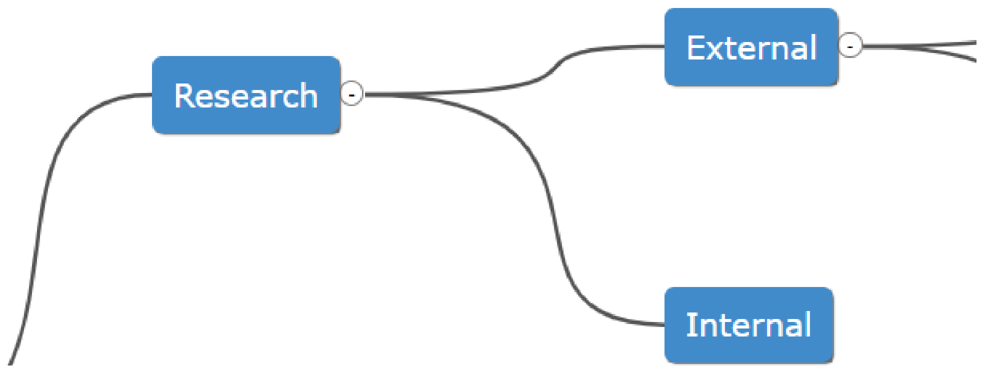
3. Currently, each node has been assigned the following properties –
   a) id
   b) topic
   c) direction
   d) children (optional)

4. Possible relations
   a) There can be only one root node.
   b) A parent node can have multiple children nodes and therefore, a child node can have multiple sibling nodes.
   c) No two parents can share the same child node.

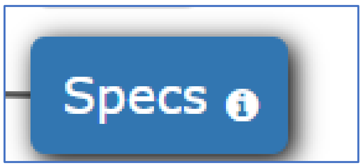
5. Each node with children nodes have two features
   a) A plus sign signifying expand to show children nodes



b) A minus sign signifying collapse to hide children nodes

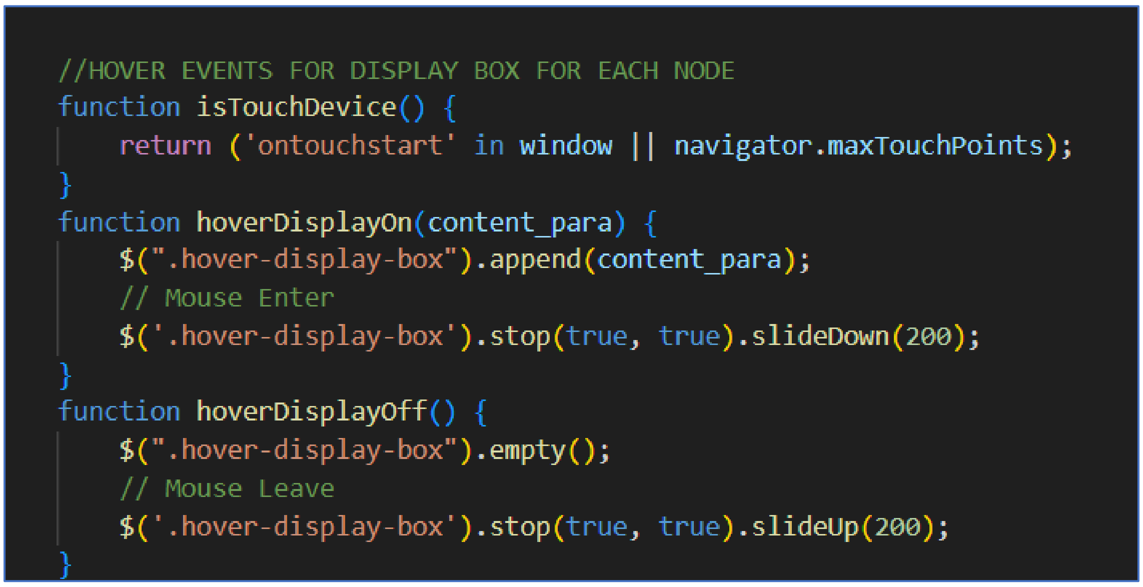


6. Nodes that have display box carry an information symbol to indicate the user to hover over/click the node.



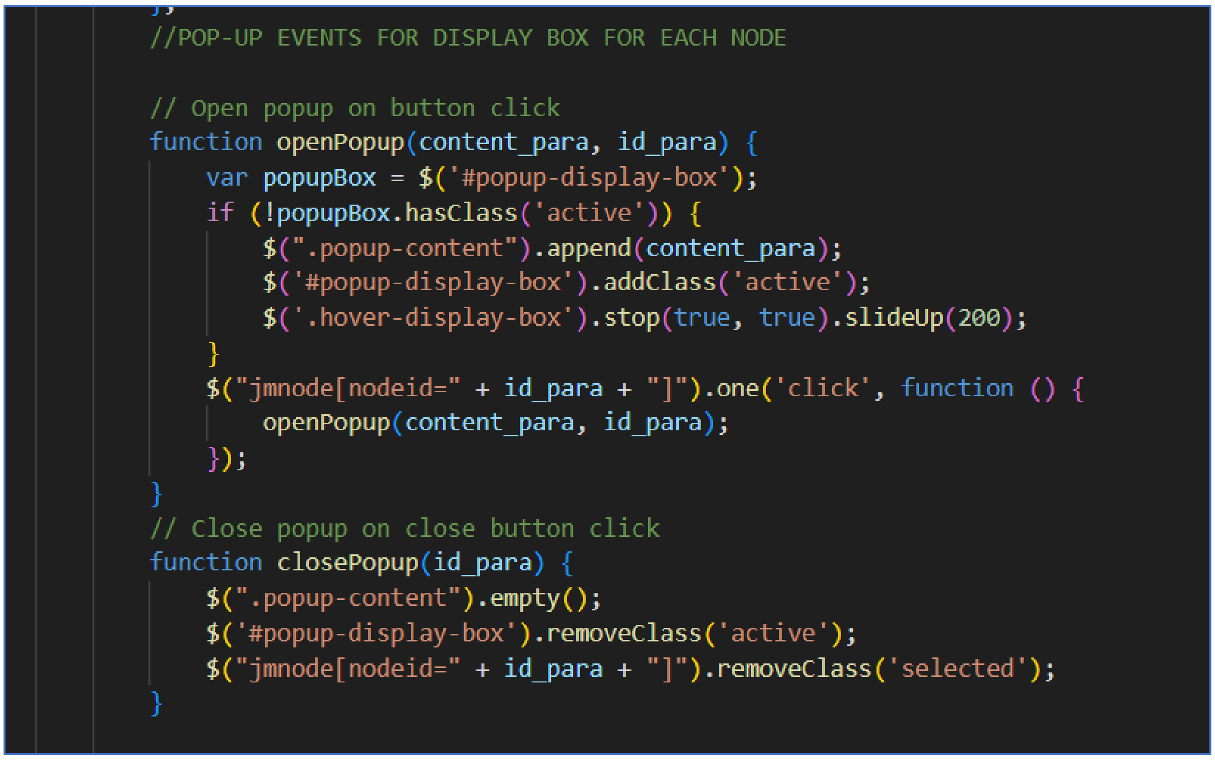
Display Box and Event Handling

1. We are using the following functions to manage hover and pop-up display box
   a) Hover functions – hoverDisplayOn() and hoverDisplayOff()



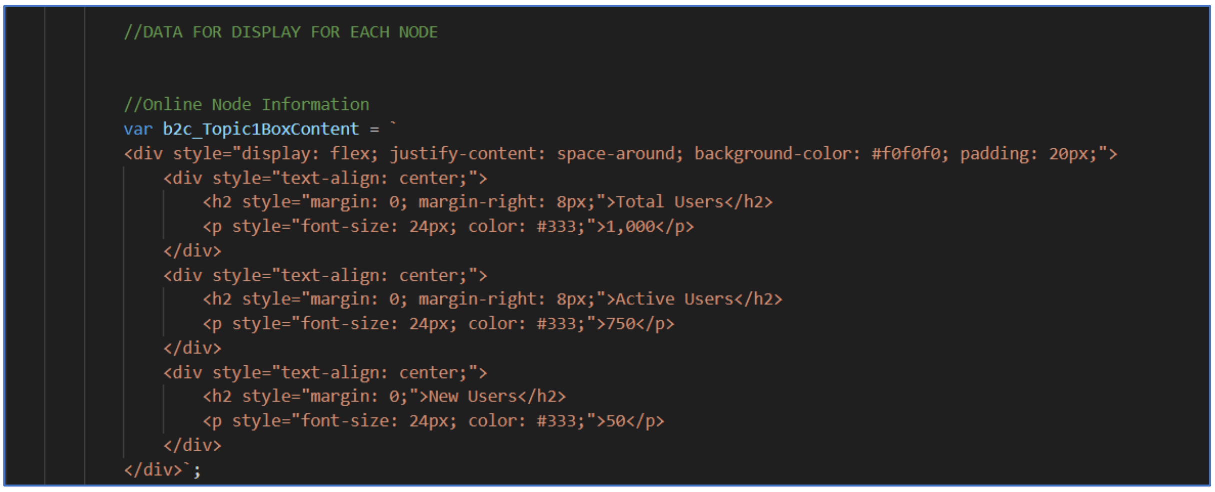
```
//HOVER EVENTS FOR DISPLAY BOX FOR EACH NODE
function isTouchDevice() {
    return ('ontouchstart' in window || navigator.maxTouchPoints);
}
function hoverDisplayOn(content_para) {
    $(".hover-display-box").append(content_para);
    // Mouse Enter
    $('.hover-display-box').stop(true, true).slideDown(200);
}
function hoverDisplayOff() {
    $(".hover-display-box").empty();
    // Mouse Leave
    $('.hover-display-box').stop(true, true).slideUp(200);
}
```

b) dPop-up Functions – openPopup() and closePopup()



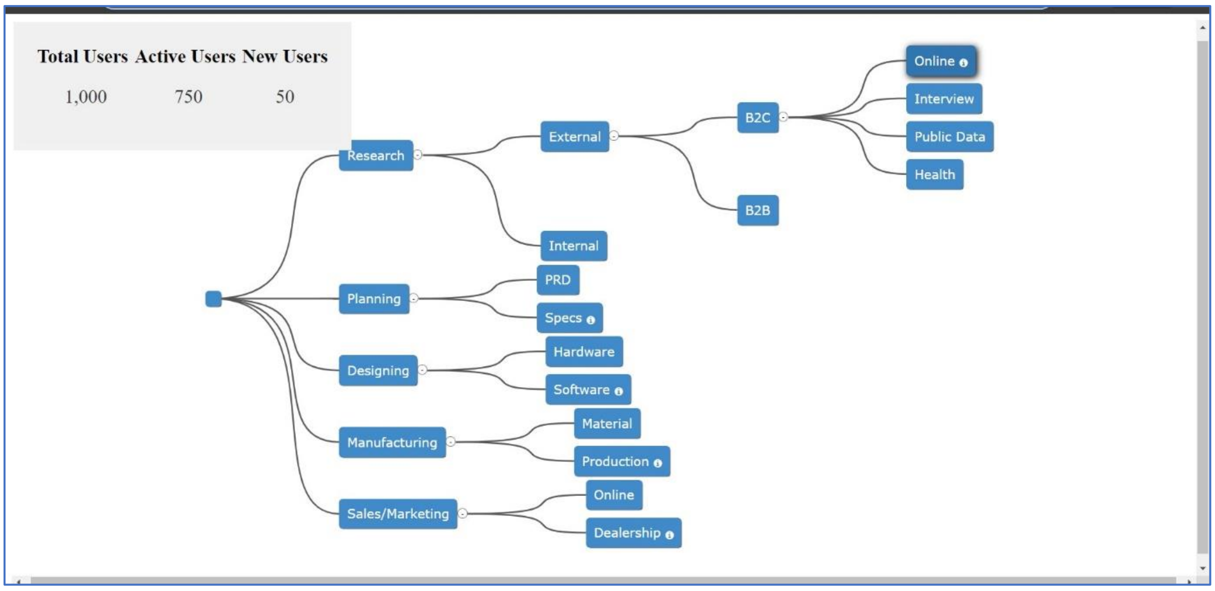
```
//POP-UP EVENTS FOR DISPLAY BOX FOR EACH NODE

// Open popup on button click
function openPopup(content_para, id_para) {
    var popupBox = $('#popup-display-box');
    if (!popupBox.hasClass('active')) {
        $(".popup-content").append(content_para);
        $('#popup-display-box').addClass('active');
        $('.hover-display-box').stop(true, true).slideUp(200);
    }
    $("jmnode[nodeid=" + id_para + "]").one('click', function () {
        openPopup(content_para, id_para);
    });
}
// Close popup on close button click
function closePopup(id_para) {
    $(".popup-content").empty();
    $('#popup-display-box').removeClass('active');
    $("jmnode[nodeid=" + id_para + "]").removeClass('selected');
}
```

2. The data inside display box is stored inside a JS variable and is fetched dynamically based on which node is clicked.



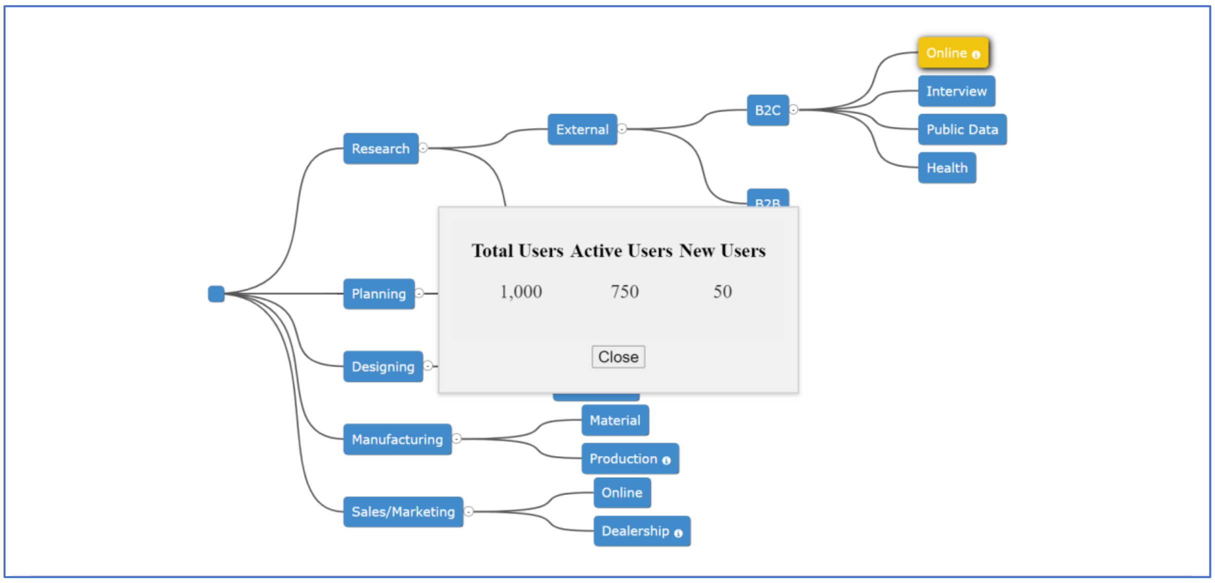
```
//DATA FOR DISPLAY FOR EACH NODE


//Online Node Information
var b2c_Topic1BoxContent = `
<div style="display: flex; justify-content: space-around; background-color: #f0f0f0; padding: 20px;">
    <div style="text-align: center;">
        <h2 style="margin: 0; margin-right: 8px;">Total Users</h2>
        <p style="font-size: 24px; color: #333;">1,000</p>
    </div>
    <div style="text-align: center;">
        <h2 style="margin: 0; margin-right: 8px;">Active Users</h2>
        <p style="font-size: 24px; color: #333;">750</p>
    </div>
    <div style="text-align: center;">
        <h2 style="margin: 0;">New Users</h2>
        <p style="font-size: 24px; color: #333;">50</p>
    </div>
</div>`;
```

3. Display Box on Hover



Total Users	Active Users	New Users
1,000	750	50

4. Display Box on Click



Total Users	Active Users	New Users
1,000	750	50

Close

Demo Video

1. 2024 Lizmotors Internship - Mindmap [Part 1]
   https://www.loom.com/share/518793d934654ab28dbea810b50fa7dc?sid=9efac21e-92b3-4811-989a-a88c80823aa7

2. 2024 Lizmotors Internship - Mindmap [Part 2]
   https://www.loom.com/share/d41d6f9af3704fb38da5387b2f97691e?sid=8a96aef2-927b-46f4-b44c-1411a37b1d5a

Additional Available Features

1. The following are the pre-built features that are available to use in jsMind library –
   a) Changing the mindmap structure in any of the available formats
   b) Changing the color theme of the mindmap
   c) Enable front-end building/editing of the mindmap done through jsMind node operations method
   d) Miscellaneous Design Changes

For a thorough review, you can check jsMind Documentation here - https://hizzgdev.github.io/jsmind/docs/en/