

National Institute Of Technology, Raipur



Name: Himanshi Gupta

Roll No:19223037

Minor Project Report

Dashboard Analysis

Submitted to:

Dr. Manju Pandey

Abstract: Dashboard for Data Analytics which displays the data in the UI in forms of various graphical representation. This helps to do analyse the data, do comparisons and have features of export particular data representation. This is implemented in HTML, CSS, JavaScript.

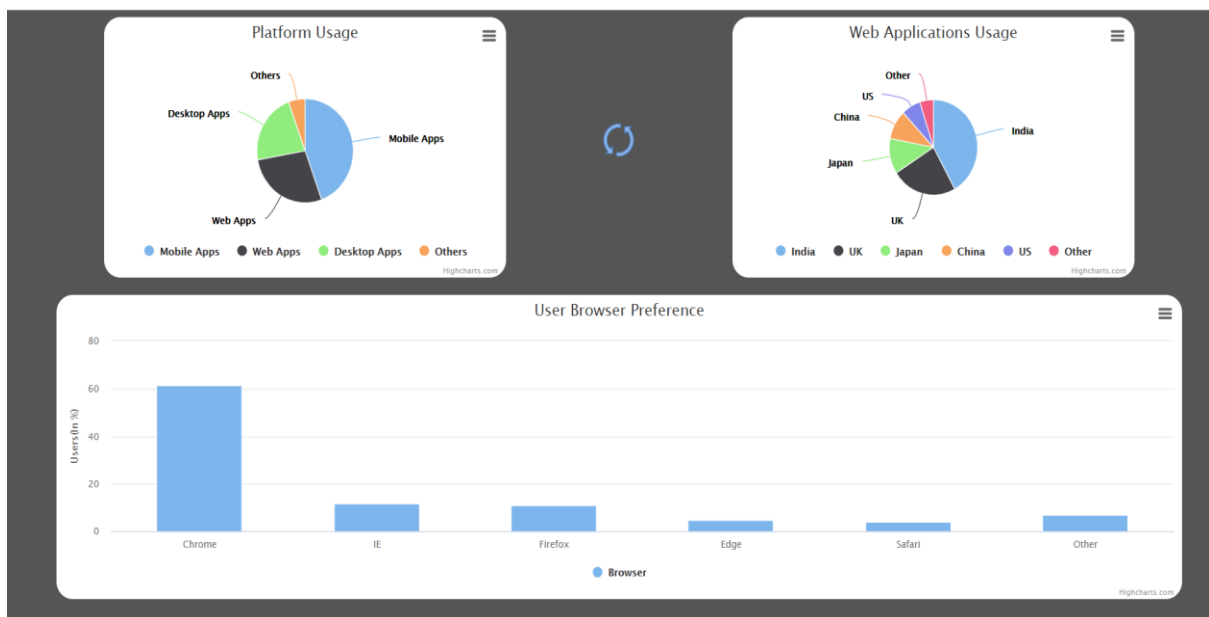
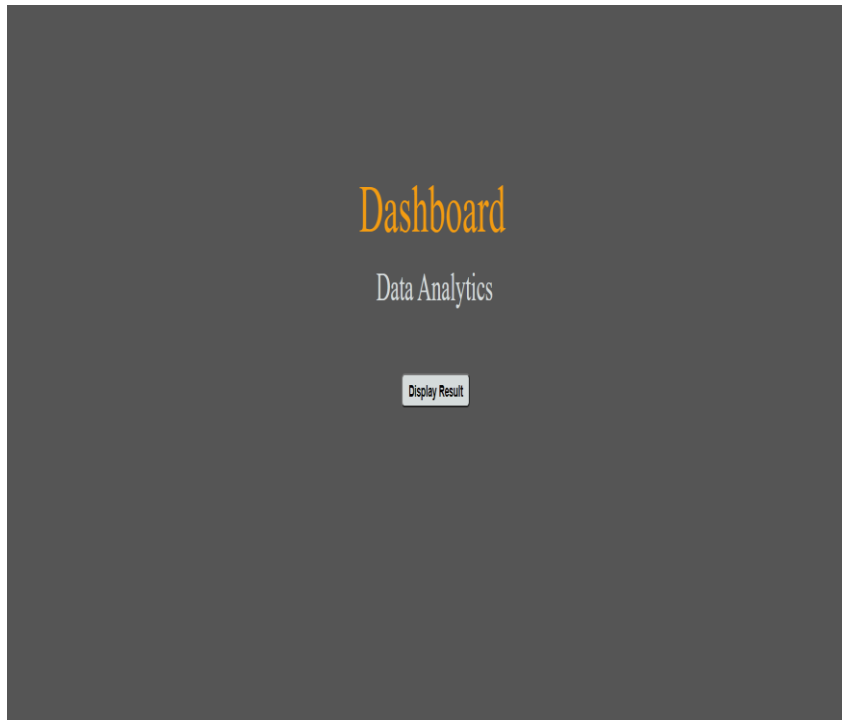
Introduction: A data dashboard is an information management tool used to track, analyze, and display [key performance indicators](#), metrics, and data points. You can use a dashboard to monitor the overall health of your business, department, or a specific process.

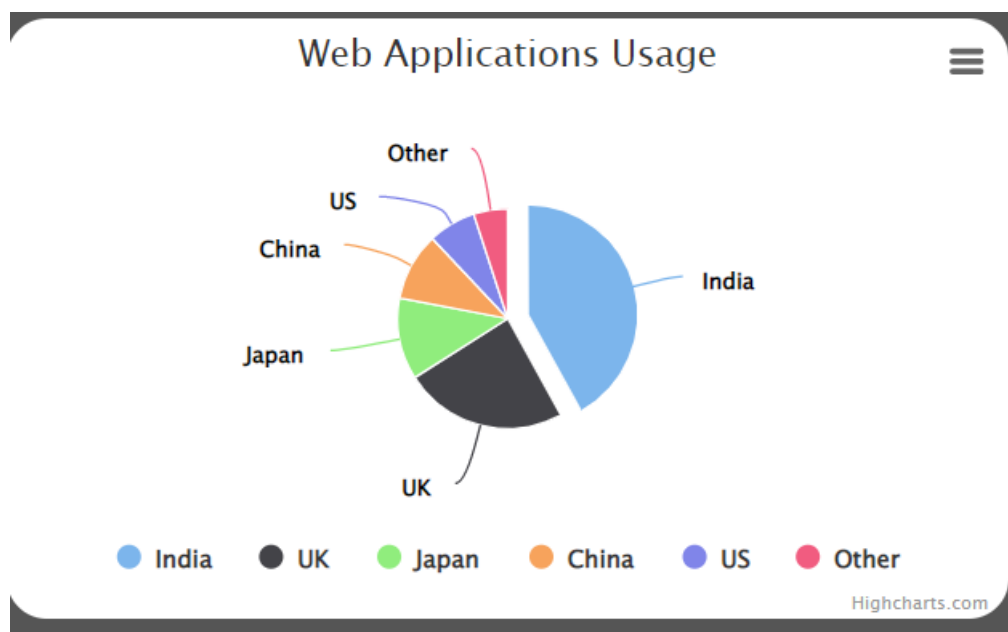
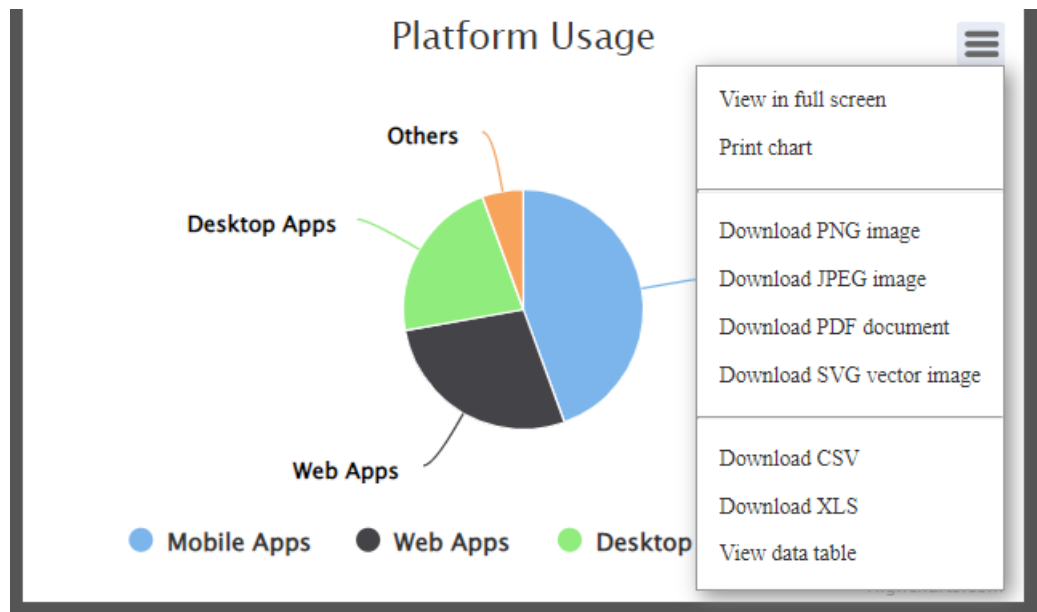
- Dashboards aggregate data from multiple data sources. a dashboard isn't just about saving time, but seeing all of your data together in one place. the connection between your data isn't another api call—it's you and the expertise you bring to the metrics that are the power behind your business.
- Dashboards are interactive—you can segment, filter, and visualize your data.
- Dashboards display live, dynamically updated data (as your data source updates, your dashboard does too).

IMPLEMENTATION: So, what I do is for now I put all the data in separately in one "JSON" file. Then we have some index page which is our landing page so from there we call every function by using some event and we use some chart library as well and its configurations we will add. One important thing here is instead of putting data in one json file we directly put data in some database like MySQL etc. so directly data is stored there and we fetch data from there using some jQuery and finally we display the result according to that using some pictorial representation.

Highlights: By this, what I want to show this is prove to be good as it possess some real time applications such as we see whenever there is poll in voting (election), in companies, in campus placements, and too see the maximum usage of something like web technologies like that so in such scenario it will be very useful.

Results: There are some screenshots:



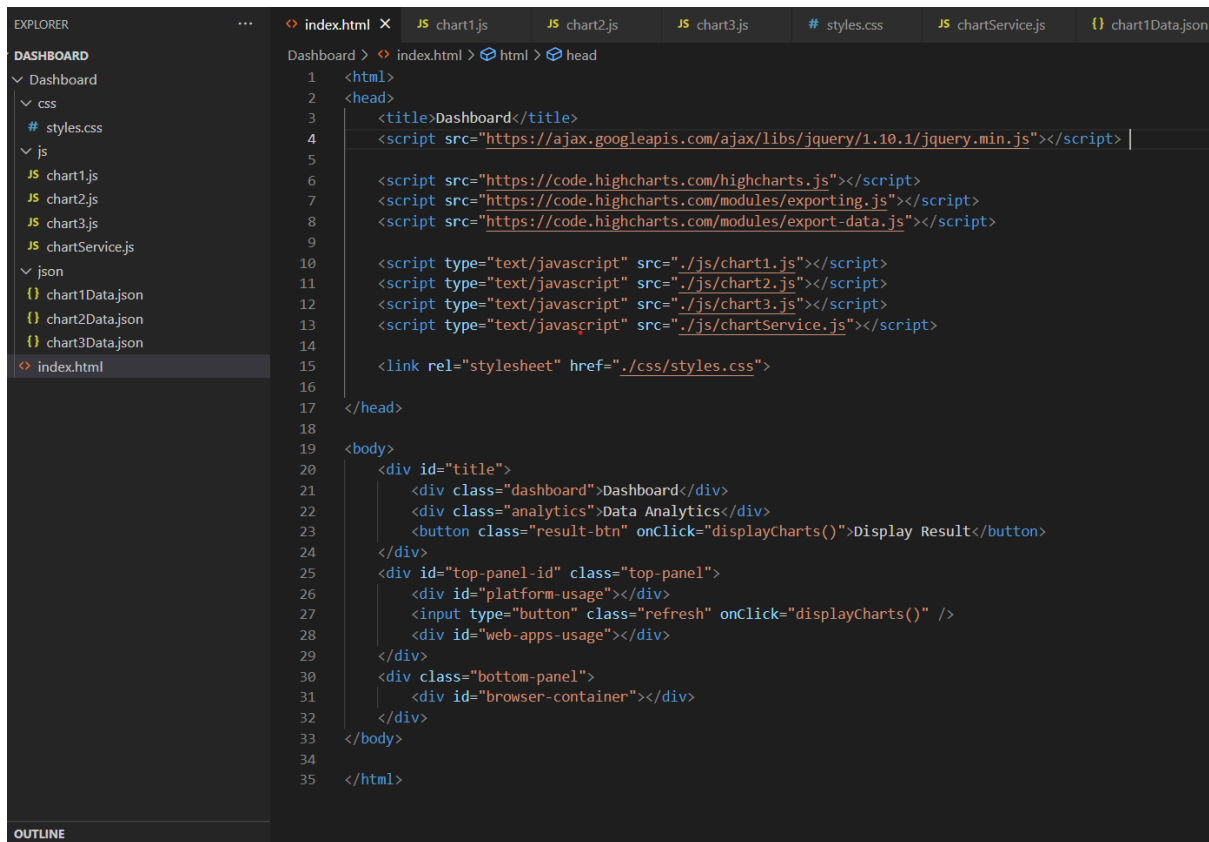


Here I use some tooltip configuration and show legends property, and for dynamically update the data I use refresh button so that we get the updated results, it not just saves our time it also provides results in a seconds and it will very much clear so that at one glance it is visible in a single screen.

So ,we have 3 JSON file in which data are stored in object form and json which stored and transmit the data in between two computers, that is why I used JSON here.

Here is some coding part screenshot:

Landing page



```
1 <html>
2 <head>
3   <title>Dashboard</title>
4   <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.10.1/jquery.min.js"></script>
5
6   <script src="https://code.highcharts.com/highcharts.js"></script>
7   <script src="https://code.highcharts.com/modules/exporting.js"></script>
8   <script src="https://code.highcharts.com/modules/export-data.js"></script>
9
10  <script type="text/javascript" src="./js/chart1.js"></script>
11  <script type="text/javascript" src="./js/chart2.js"></script>
12  <script type="text/javascript" src="./js/chart3.js"></script>
13  <script type="text/javascript" src="./js/chartService.js"></script>
14
15  <link rel="stylesheet" href="./css/styles.css">
16
17 </head>
18
19 <body>
20   <div id="title">
21     <div class="dashboard">Dashboard</div>
22     <div class="analytics">Data Analytics</div>
23     <button class="result-btn" onClick="displayCharts()">Display Result</button>
24   </div>
25   <div id="top-panel-id" class="top-panel">
26     <div id="platform-usage"></div>
27     <input type="button" class="refresh" onClick="displayCharts()" />
28     <div id="web-apps-usage"></div>
29   </div>
30   <div class="bottom-panel">
31     <div id="browser-container"></div>
32   </div>
33 </body>
34
35 </html>
```

All json data for different chart

- For chart1 : Platform Usage

The screenshot shows the VS Code interface. On the left, the Explorer sidebar displays a project structure under 'DASHBOARD' with folders 'css' and 'js', and a 'json' folder containing 'chart1Data.json', 'chart2Data.json', and 'chart3Data.json'. The 'json' folder is expanded, and 'chart1Data.json' is selected. The main editor shows the content of 'chart1Data.json' with line numbers 1 through 18. The JSON data is as follows:

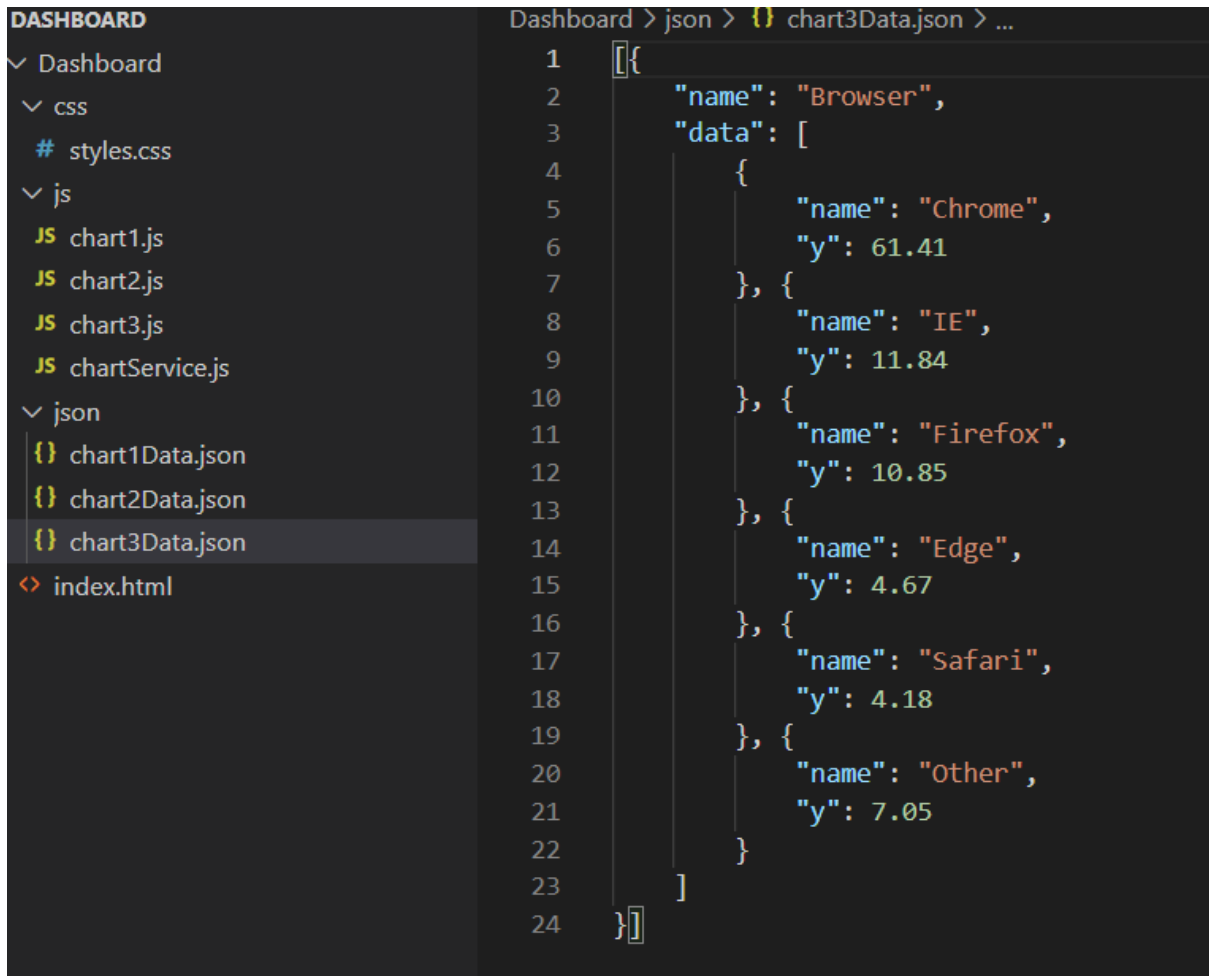
```
1 [
2   {
3     "name": "Mobile Apps",
4     "y": 40
5   },
6   {
7     "name": "Web Apps",
8     "y": 25
9   },
10  {
11    "name": "Desktop Apps",
12    "y": 20
13  },
14  {
15    "name": "Others",
16    "y": 5
17  }
18 ]
```

For chart2 json data

The screenshot shows the VS Code interface. On the left, the Explorer sidebar displays the same project structure as the first image, but 'chart2Data.json' is now selected. The main editor shows the content of 'chart2Data.json' with line numbers 1 through 21. The JSON data is as follows:

```
1 [
2   {
3     "name": "India",
4     "y": 42
5   }, {
6     "name": "UK",
7     "y": 24
8   }, {
9     "name": "Japan",
10    "y": 12
11  }, {
12    "name": "China",
13    "y": 10
14  }, {
15    "name": "US",
16    "y": 7
17  }, {
18    "name": "Other",
19    "y": 5
20  }
21 ]
```

For chart 3 json data



```
DASHBOARD
Dashboard > json > {} chart3Data.json > ...

1  [{
2      "name": "Browser",
3      "data": [
4          {
5              "name": "Chrome",
6              "y": 61.41
7          }, {
8              "name": "IE",
9              "y": 11.84
10         }, {
11             "name": "Firefox",
12             "y": 10.85
13         }, {
14             "name": "Edge",
15             "y": 4.67
16         }, {
17             "name": "Safari",
18             "y": 4.18
19         }, {
20             "name": "Other",
21             "y": 7.05
22         }
23     ]
24 }]
```

For make it fancy and perfect border and container its size I use CSS so below is CSS code.

index.html <body> JS chart2.js JS chart3.js # styles.css JS chartService.js JS chart1.js

Dashboard Selector Specificity: (0, 0, 1) dy

```
1 body {
2     background-color: #555;
3 }
4
5 .top-panel {
6     display: none;
7     flex-direction: row;
8 }
9
10 /* .bottom-panel {
11     display: none;
12     flex-direction: row;
13 } */
14
15 .refresh {
16     height: 42px;
17     width: 42px;
18     background: url('https://img.icons8.com/office/40/000000/refresh--v1.png') no-repeat;
19     margin-top: 8%;
20     border: none;
21 }
22
23 #platform-usage, #web-apps-usage {
24     min-width: 500px;
25     height: 300px;
26     max-width: 600px;
27     margin: 0 auto;
28 }
29
30 #browser-container {
31     min-width: 500px;
32     height: 350px;
33     max-width: 1400px;
34     margin: 0 auto;
35     display: none;
36 }
37
38 .highcharts-container {
```



```
Dashboa
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
```

<element class="bottom-panel">

<element id="browser-container">

Selector Specificity: (1, 1, 0)

```
.bottom-panel #browser-container {
  margin-top: 20px;
}

#title {
  max-width: 20%;
  margin: auto;
  margin-top: 10%;
}

#title .analytics {
  margin-top: 5%;
}

#title .analytics {
  font-size: 35px;
  margin-left: 13%;
  color: ■ #D5DBDB;
}

#title .dashboard {
  font-size: 60px;
  margin-left: 3%;
  color: ■ #F39C12;
}

.result-btn {
  height: 30px;
  border-radius: 5px;
  font-weight: bold;
  background-color: ■ #D5DBDB;
  font-size: 14px;
  padding: 6px 10px;
  margin: 19% 0 0 28%;
```

High chart library configuration is added and corresponding function is now call from there.

```
Dashboard > js > JS chartService.js > hideTitleBlock
1  function hideTitleBlock() {
2      var titleBlock = document.getElementById("title");
3      titleBlock.style.display = "none";
4
5      var panel1 = document.getElementById("top-panel-id");
6      panel1.style.display = "flex";
7
8      var panel2 = document.getElementById("browser-container");
9      panel2.style.display = "flex";
10 }
11
12 function displayCharts() {
13     hideTitleBlock();
14     displayPieChart();
15     displayHighChart();
16     displayBottomChart();
17 }
```

It will go to that function line by line

```
ashboard > js > JS chart1.js > displayPieChart > $.getJSON("./json/chart1Data.json") callback > plotOp
1  v function displayPieChart() {
2
3  v   $.getJSON("./json/chart1Data.json", function (data) {
4  v       Highcharts.chart('platform-usage', {
5  v           chart: {
6  v               type: 'pie'
7  v           },
8  v           title: {
9  v               text: "Platform Usage"
10 v           },
11 v           tooltip: {
12 v               pointFormat: '{series.name}: <b>{point.percentage:.1f}</b>'
13 v           },
14 v           plotOptions: {
15 v               pie: {
16 v                   allowPointSelect: true,
17 v                   cursor: 'pointer',
18 v                   showInLegend: true
19 v               }
20 v           },
21 v           series: [{
22 v               name: 'Platform',
23 v               colorByPoint: true,
24 v               data: data
25 v           }]
26 v       });
27 v   });
28 }
29
```

Dashboard > js > JS chart2.js > displayHighChart

```
1 function displayHighChart() {
2     $.getJSON("../json/chart2Data.json", function (data) {
3         Highcharts.chart('web-apps-usage', {
4             chart: {
5                 type: 'pie'
6             },
7             title: {
8                 text: "Web Applications Usage"
9             },
10            tooltip: {
11                pointFormat: '{series.name}: <b>{point.percentage:.1f}</b>'
12            },
13            plotOptions: {
14                pie: {
15                    allowPointSelect: true,
16                    cursor: 'pointer',
17                    showInLegend: true
18                }
19            },
20            series: [{
21                name: 'Region',
22                colorByPoint: true,
23                data: data
24            }]
25        });
26    });
27 }
28
```

```

function displayBottomChart() {
    $.getJSON("../json/chart3Data.json", function (data) {
        Highcharts.chart('browser-container', {
            chart: {
                type: 'column'
            },
            title: {
                text: "User Browser Preference"
            },
            tooltip: {
                pointFormat: '{series.name}: <b>{point.percentage:.1f}</b>'
            },
            yAxis: {
                min: 0,
                title: {
                    text: 'Users(In %)'
                }
            },
            xAxis: {
                categories: [
                    'Chrome',
                    'IE',
                    'Firefox',
                    'Edge',
                    'Safari',
                    'Other'
                ]
            },
            series: data
        });
    });
}

```

Conclusion: Dashboards serve different purposes for different roles. A data dashboard is an analytics tool that helps capture, track, and consume data in a way that is meaningful to your business. ... a visual display of the most important information needed to achieve one or more objectives; consolidated and arranged on a single screen so the information can be monitored at a glance.