

## **Practical No. 1: Setup the environment for all the data visualization tools.**

### **1. Tableau Software:**

Tableau Software is a company entirely dedicated to the data analysis and visualization. Through its product Tableau Desktop and Tableau Server it offers users a powerful tool for data analysis and enrichment. In addition, its powerful graphics engine can generate stunning visualizations on large volumes of data. Tableau Software is widely used in the communication sector (New York Times, O'Reilly Media), pharmaceuticals (Bayer CropScience) and education (mainly American universities).

### **2. Tabulae:**

Tabulae is a web platform based on W3C standards, for data exploitation and visual analysis. It is a flexible tool that stands out for its ease of use and ability to fully customize the presentation of information through dashboards and interactive reports. Tabulae provides dynamic mechanisms of exploitation, enriching the users' experience and their ability to interpret the data. Without programming knowledge, you can transform any data set into an interactive web application accessible from any device.

### **3. QlikView y QlikSense:**

QlikTech offers two main products QlikView and QlikSense. Its flagship product, QlikView is a business intelligence software that stands out for its ease of use and visuality allowing business users to make decisions based on data. QlikView enables the consolidation of data from multiple sources in a single application, which facilitates the exploration of associations between data. On the other hand QlikSense is a lighter and simpler version more oriented to visualization than analytics. Both solutions allow the construction of dashboards and reports easily. The website has a varied catalog of demos, which allows us to quickly discover functionality and visual appearance.

### **4. Carto:**

Carto is a cloud service that provides GIS capabilities and tools to build maps accessible through a web browser. Carto is a service widely used for creating thematic

maps because of its ease of use and visually attractive results, so we can find maps created with this tool in many news digital newspapers, blogs, etc.

#### 5. **Socrata:**

Socrata is a company that offers to non-technical users who want to share information the ability to easily create visualizations of their data. Founded in 2007 it is specialized in supporting the public sector offering a comprehensive service ranging from support for data collection to its conversion into applications. It is currently present in a number of open data portals such as the data transparency portal of Gijón.

#### 6. **CKAN (Visualization Modules)**

The widely known platform for publishing data catalogs, CKAN, offers a set of visualization modules that can generate different graphical representations. The Table view allows us to have a tabular version, in table format, of the data of a particular dataset, offering listing tools and faceted view. It also has a module for data visualization in graph form, enabling the comparison of variables through a series within the same graph. Similarly, if the data contains geo-referenced data the module maps can be used to create interactive visualizations.

### **EVALUATION:**

Involvement (4)	Understanding / Problem solving (3)	Timely Completion (3)	<b>Total (10)</b>

**Signature with date:**

## PRACTICAL-2

### 2(a). Develop the Different basic Graphical Shapes using HTML5 CANVAS

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="utf-8">

  <title>Filling Color inside a Rectangle on the Canvas</title>

  <style>

    canvas

    {

      border: 1px solid #000;

    }

  </style>

  <script>

    window.onload = function () {

      var canvas = document.getElementById("myCanvas");

      var context = canvas.getContext("2d");

      context.rect(50, 50, 200, 100);

      context.fillStyle = "#FB8B89";

      context.fill();

      context.lineWidth = 5;

      context.strokeStyle = "black";

      context.stroke();

    };

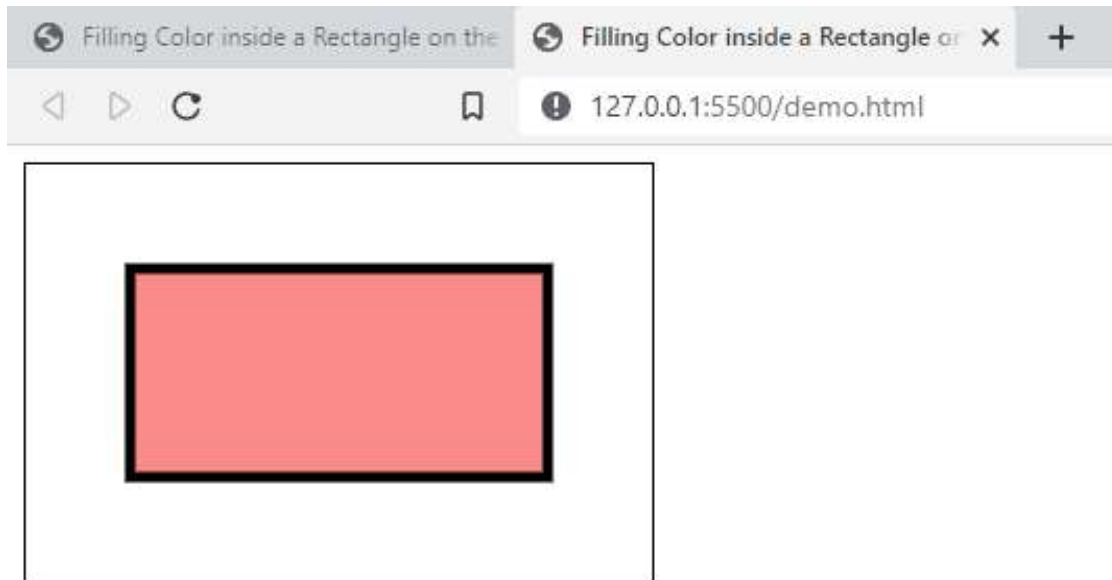
  </script>

</head>

<body>
```

```
<canvas id="myCanvas" width="300" height="200"></canvas>  
</body>  
</html>
```

**Output:**



## 2(B).Develop different advanced graphical shapes us using HTML5 canvas

```
<!DOCTYPE html>

<html>

<head>

  <title>HTML5 Canvas - Sine wave</title>

  <style>

    body {

      background: #151515;

    }

    div {

      z-index: 2;

      position: absolute;

    }

    input[type=range] {

      width: 200px;

      height: 15px;

      -webkit-appearance: none;

      background: #111;

      outline: none;

      margin: 5px;

      position: relative;

      border-radius: 15px;

      overflow: hidden;

      box-shadow: inset 0 0 5px rgba(0, 0, 0, 1);

    }

    input[type=range]::-moz-range-thumb,
```

```

input[type=range]::-webkit-slider-thumb {
    -webkit-appearance: none;
    width: 15px;
    height: 15px;
    border-radius: 50%;
    background: #00fd0a;
    cursor: pointer;
    border: 4px solid #333;
    box-shadow: -407px 0 0 400px #00fd0a;
}
</style>
</head>

<body>
    <div>
        <input title="height" type="range" id="height" oninput="ranges()"
onchange="ranges()">

        <input title="wavelength" type="range" id="wavelength" oninput="ranges()"
onchange="ranges()">

        <input title="amplitude" type="range" id="amplitude" oninput="ranges()"
onchange="ranges()">

        <input title="frequency" type="range" id="freq" oninput="ranges()"
onchange="ranges()"></br>

        <input title="H" type="range" id="h" oninput="color()" onchange="color()">
        <input title="S" type="range" id="s" oninput="color()" onchange="color()">
        <input title="L" type="range" id="l" oninput="color()" onchange="color()">
    </div>

    <canvas style="background-color:white"></canvas>

```

</body>

<script>

```
var inheight = document.getElementById('height');
var inwavelen = document.getElementById('wavelength');
var inamplitude = document.getElementById('amplitude');
var infreq = document.getElementById('freq');
var inh = document.getElementById('h');
var ins = document.getElementById('s');
var inl = document.getElementById('l');
var canvas = document.querySelector('canvas');
var c = canvas.getContext("2d");
var y, wavelength, amplitude, freq;
var inc = 0.01;//change to 0.01 to make //animation
var H, W, h, s, l;
// inc=parseFloat(freq);
function animate() {
    requestAnimationFrame(animate);
    c.fillStyle = "rgba(0,0,0,0.01)";
    c.fillRect(0, 0, W, H); //animation
    c.beginPath();
    c.moveTo(0, y);
    for (let i = 0; i < canvas.width; ++i) {
        c.lineTo(i, y + Math.sin(i * wavelength + inc) * amplitude);
    }
    c.stroke();
    c.strokeStyle = `hsl(${h},${s}%,${l}%)`;
    inc += parseFloat(freq); //animation
```

```

}
animate();
function ranges() {
    y = inheight.value / 2;
    wavelength = inwavelen.value;
    ampliude = inamplitude.value;
    freq = infreq.value;
}
function color() {
    h = inh.value;
    s = ins.value;
    l = inl.value;
    c.strokeStyle = `hsl(${r},${g}%,${b}%)`;
}
canvas.width = window.innerWidth;
canvas.height = window.innerHeight;
H = canvas.height;
W = canvas.width;
//set min/max height of wave
inheight.min = 0; inheight.max = H; inheight.value = H;
//set min/max/step wavelength of wave
inwavelen.step = 0.001;
inwavelen.min = -0.01;
inwavelen.max = 0.01;
inwavelen.value = 0.01;
//set amplitude of wave
inamplitude.min = -300;

```



```
inamplitude.max = 300;

inamplitude.value = 100;

//set frequency

infreq.step = 0.01;

infreq.min = -0.1;

infreq.max = 0.1;

infreq.value = 0.01;

//set color values

inh.max = 255;

inh.min = 0;

inh.value = 0;

ins.max = 100;

ins.min = 0;

ins.value = 50;

inl.max = 100;

inl.min = 0;

inl.value = 50;

y = inheight.value / 2;

wavelength = inwavelen.value;

ampliude = inamplitude.value;

freq = infreq.value;

inc = parseFloat(freq);

h = inh.value;

s = ins.value;

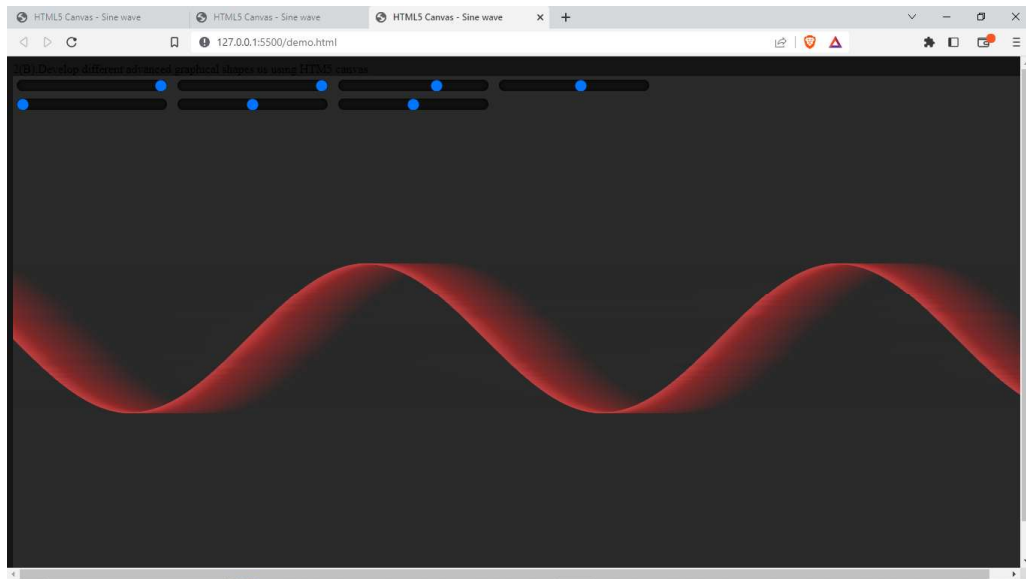
l = inl.value;

c.strokeStyle = `hsl(${s},${s}%,${l}%)`;

window.onresize = () => {
```

```
canvas.width = window.innerWidth;  
canvas.height = window.innerHeight;  
  
H = canvas.height;  
W = canvas.width;  
  
}</script>  
</html>
```

## Output:



## 2(C) : Develop the Different basic Graphical Shapes using HTML5 SVG

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="utf-8">

  <title>Create a Circle with HTML5 SVG</title>

  <style>

    svg {

      border: 1px solid black;

    }

  </style>

</head>

<body>

  <svg width="300" height="200">

    <circle cx="150" cy="100" r="70" style="fill:lime;

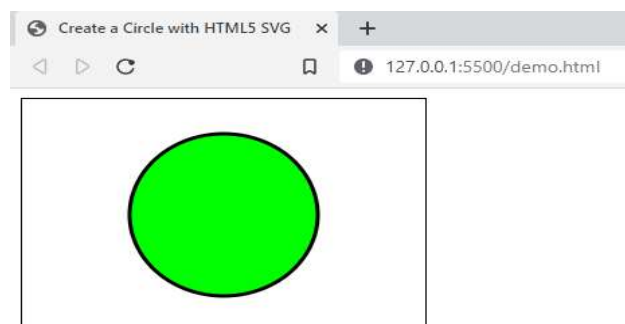
stroke:black; stroke-width:3;" />

  </svg>

</body>

</html>
```

### Output:



## 2(D) : . Develop the Different Advanced Graphical Shapes using HTML5 SVG.

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="utf-8">

  <title>Create a Circle with HTML5 SVG</title>

  <style>

    svg {

      border: 1px solid black;

    }

  </style>

</head>

<body>

  <svg width="400" height="400" style="border: 1px solid #ccc">

    <polygon points="200,0 0,100 50,115" style="fill:orange;stroke:yellow;stroke
width:1; "></polygon>

    <polygon points="200,0 50,115 150,135"
style="fill:blue;stroke:yellow;strokewidth:1;opacity:0.5 "></polygon>

    <polygon points="200,0 150,135 245,135" style="fill:black;stroke:yellow;stroke-
width:1;opacity:0.5 "></polygon>

    <polygon points="200,0 245,135 350,128" style="fill:green;stroke:yellow;stroke-
width:1;opacity:0.5 "></polygon>

    <polygon points="200,0 350,128 400,110"
style="fill:red;stroke:yellow;strokewidth:1;opacity:0.5 "></polygon>

    <line x1="200" y1="0" x2="200" y2="300"
style="fill:red;stroke:black;strokewidth:5;"></line>

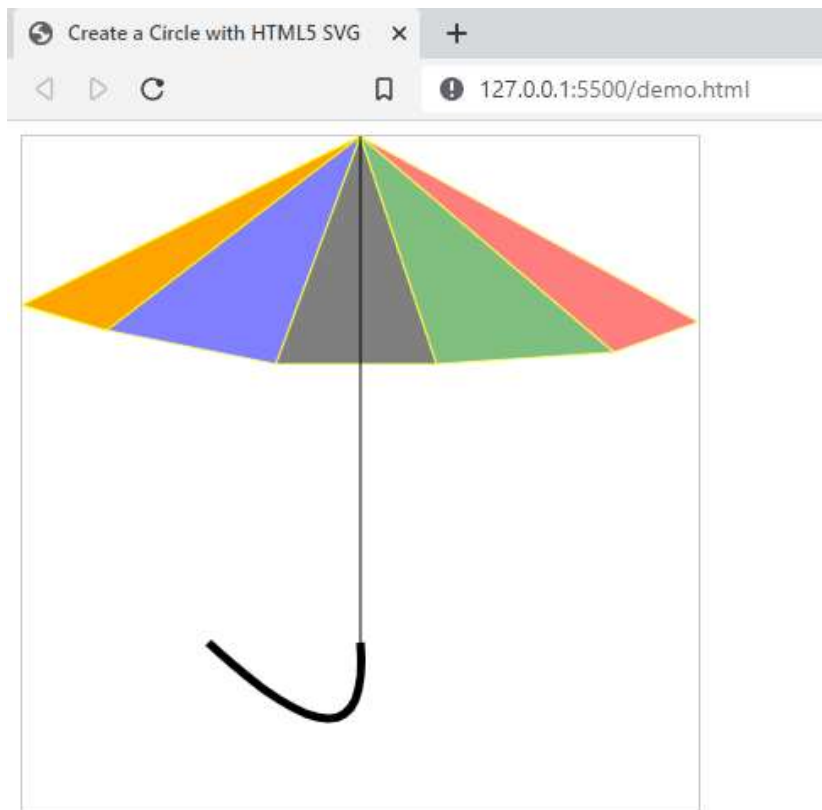
    <path d="M 200 300 q 7 90 -90 0" stroke="black" stroke-width="5" fill="none" />

  </svg>
```

</body>

</html>

**Output:**



## PRACTICAL: 3

### 3(a) : Develop the simple bar chart using TML5 CANVAS

```
<!DOCTYPE HTML>

<html>

<head>

<script>

    window.onload = function () {

        var chart = new CanvasJS.Chart("chartContainer", {

            animationEnabled: true,

            title: {

                text: "Fortune 500 Companies by Country"

            },

            axisX: {

                interval: 1

            },

            axisY2: {

                interlacedColor: "rgba(1,77,101,.2)",

                gridColor: "rgba(1,77,101,.1)",

                title: "Number of Companies"

            },

            data: [{

                type: "bar",

                name: "companies",

                axisYType: "secondary",

                color: "#014D65",

                dataPoints: [

                    { y: 3, label: "Sweden" },
```

```

        { y: 7, label: "Taiwan" },
        { y: 5, label: "Russia" },
        { y: 9, label: "Spain" },
        { y: 7, label: "Brazil" },
        { y: 7, label: "India" },
        { y: 9, label: "Italy" },
        { y: 8, label: "Australia" },
        { y: 11, label: "Canada" },
        { y: 15, label: "South Korea" },
        { y: 12, label: "Netherlands" },
        { y: 15, label: "Switzerland" },
        { y: 25, label: "Britain" },
        { y: 28, label: "Germany" },
        { y: 29, label: "France" },
        { y: 52, label: "Japan" },
        { y: 103, label: "China" },
        { y: 134, label: "US" }

    ]

    }

    });

    chart.render();

}

</script>

</head>

<body>

    <div id="chartContainer" style="height: 300px; width: 100%;"></div>

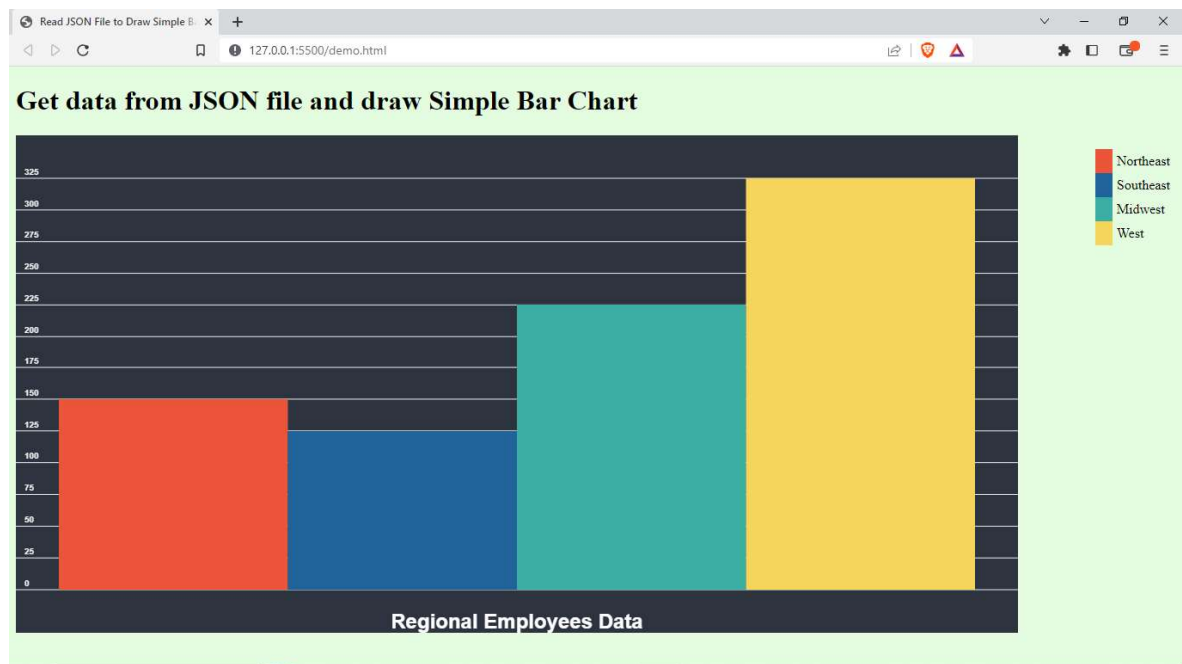
```

```
<script src="https://canvasjs.com/assets/script/canvasjs.min.js"></script>

</body>

</html>
```

## Output:





### 3(B) : Read the data .txt file and draw Data Table

```
<!DOCTYPE html>
```

```
<html lang="en-IN" dir="ltr">
```

```
<head>
```

```
<title>Read Text File to Data Table</title>
```

```
<meta charset="utf-8">
```

```
<meta name="author" content="SidPro"/>
```

```
<meta name="viewport" content="width=device-width,initial-scale=1.0"/>
```

```
<meta name="description" content="Data visualization using javaScript and HTML"/>
```

```
<script src="js/jquery-3.5.1.min.js"></script>
```

```
<style>
```

```
#mytable {
```

```
font-family: Arial, Helvetica, sans-serif;
```

```
border-collapse: collapse;
```

```
width: 100%;
```

```
}
```

```
#mytable td, #mytable th {
```

```
border: 1px solid #ddd;
```

```
padding: 8px;
```

```
}
```

```
#mytable tr:nth-child(even){background-color: #f2f2f2;}
```

```
#mytable tr:hover {background-color: #ddd;}
```

```
#mytable th {
```

```

padding-top: 12px;

padding-bottom: 12px;

text-align: left;

background-color: cornflowerblue;

color: white;

}

</style>

</head>

<body>

  <h1>Get data from text file</h1>

  <table id="mytable">

  </table>

</body>

<script--0987uyt04:50 PM 02-02-2023

var content = "";

$( document ).ready(function() {

//just change the name of .txt file to load file

$.get('data/data.txt', function(theData) {

theData = theData.replace(/\r/g,"");

theData = theData.replace(/\t/g,' ');

theData = theData.split('\n');

totalRows = theData.length;

theHead = theData[0].split(' ');

content += "<tr>";

theHead.forEach(TH);

```

```
        content += "</tr>";

        for(let i=1;i<totalRows;++i){

            theTD = theData[i].split(' ');

            content += "<tr>";

            theTD.forEach(TD);

            content += "</tr>";

        }

        $('#mytable').html(content);

    });

});

function TH(value){

    content += "<th>" + value + "</th>";

}

function TD(value){

    content += "<td>" + value + "</td>";

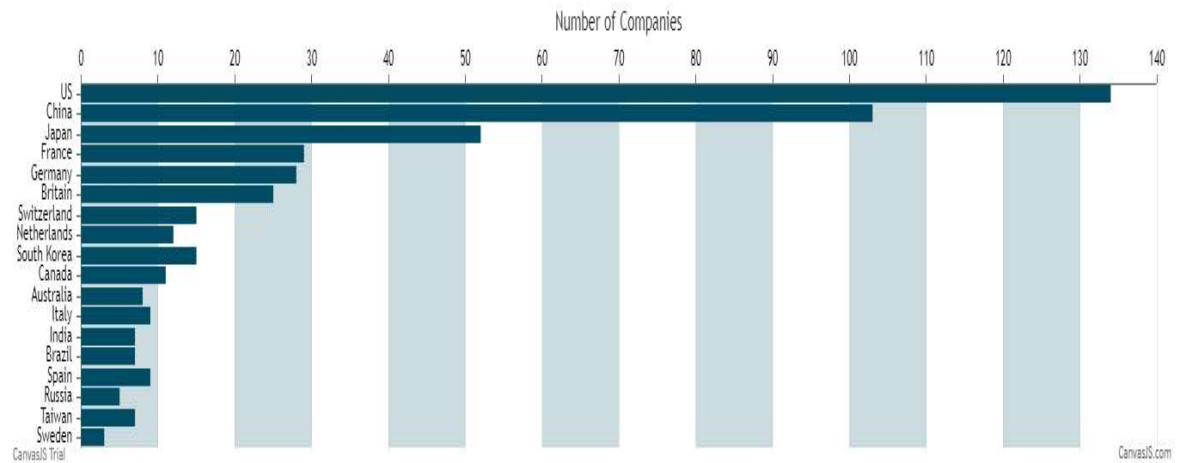
}

</script>

</html>
```

**Output:**

## Fortune 500 Companies by Country



### 3(C) : Read the data .txt file and draw Simple Bar ChartS

```
<!DOCTYPE html>

<html lang="en-IN" dir="ltr">

<head>

    <title>Read Text File to Draw Simple Bar Chart</title>

    <meta charset="utf-8">

    <meta name="author" content="SidPro" />

    <meta name="viewport" content="width=device-width,initial-scale=1.0" />

    <meta name="description" content="Data visualization using javaScript and HTML" />

    <script src="js/jquery-3.5.1.min.js"></script>

    <style>

        #mytable {

            font-family: Arial, Helvetica, sans-serif;

            border-collapse: collapse;

            width: 100%;

        }

        #mytable td,

        #mytable th {

            border: 1px solid #ddd;

            padding: 8px;

        }

        #mytable tr:nth-child(even) {

            background-color: #f2f2f2;

        }

        #mytable tr:hover {
```

```

        background-color: #ddd;
    }
    #mytable th {
        padding-top: 12px;
        padding-bottom: 12px;
        text-align: left;
        background-color: cornflowerblue;
        color: white;
    }
    legend {
        float: right;
    }
</style>
</head>
<body>
    <h1>Get data from text file and draw Simple Bar Chart</h1>
    <label style="font-size:18px" for="month"><b>Choose a Month: </b></label>
    <select id="month" onchange="draw()">
        <option value="1">January</option>
        <option value="2" selected>February</option>
        <option value="3">March</option>
        <option value="4">April</option>
        <option value="5">May</option>
        <option value="6">June</option>
        <option value="7">July</option>
        <option value="8">August</option>
        <option value="9">September</option>

```

```
<option value="10">October</option>
<option value="11">November</option>
<option value="12">December</option>
</select><br><br>
```

```
<canvas id="myCanvas" style="background: #2e3340;"></canvas>
```

```
<legend for="myCanvas"></legend>
```

```
</body>
```

```
<script>
```

```
var myCanvas = document.getElementById("myCanvas");
```

```
myCanvas.width = window.innerWidth - 200;
```

```
myCanvas.height = window.innerHeight - 120;
```

```
var ctx = myCanvas.getContext("2d");
```

```
function drawLine(ctx, startX, startY, endX, endY, color) {
```

```
    ctx.save();
```

```
    ctx.strokeStyle = color;
```

```
    ctx.beginPath();
```

```
    ctx.moveTo(startX, startY);
```

```
    ctx.lineTo(endX, endY);
```

```
    ctx.stroke();
```

```
    ctx.restore();
```

```
}
```

```
function drawBar(ctx, upperLeftCornerX, upperLeftCornerY, width, height, color) {
```

```
    ctx.save();
```

```
ctx.fillStyle = color;

ctx.fillRect(upperLeftCornerX, upperLeftCornerY, width, height);

ctx.restore();

}
```

```
var Barchart = function (options) {

  this.options = options;

  this.canvas = options.canvas;

  this.ctx = this.canvas.getContext("2d");

  this.colors = options.colors;

  this.draw = function () {

    var maxValue = 0;

    for (var categ in this.options.data) {

      maxValue = Math.max(maxValue, this.options.data[categ]);

    }

    var canvasActualHeight = this.canvas.height - this.options.padding * 2;

    var canvasActualWidth = this.canvas.width - this.options.padding * 2;

    //drawing the grid lines

    var gridValue = 0;

    while (gridValue <= maxValue) {

      var gridY = canvasActualHeight * (1 - gridValue / maxValue) +
this.options.padding;

      drawLine(

        this.ctx,
```



```

    0,

    gridY,

    this.canvas.width,

    gridY,

    this.options.gridColor
);

//writing grid markers

this.ctx.save();

this.ctx.fillStyle = this.options.gridColor;

this.ctx.textBaseline = "bottom";

this.ctx.font = "bold 10px Arial";

this.ctx.fillText(gridValue, 10, gridY - 2);

this.ctx.restore();

    gridValue += this.options.gridScale;
}

//drawing the bars

var barIndex = 0;

var numberOfBars = Object.keys(this.options.data).length;

var barSize = (canvasActualWidth) / numberOfBars;

for (categ in this.options.data) {

    var val = this.options.data[categ];

    var barHeight = Math.round(canvasActualHeight * val / maxValue);

    drawBar(

```

```

        this.ctx,

        this.options.padding + barIndex * barSize,

        this.canvas.height - barHeight - this.options.padding,

        barSize,

        barHeight,

        this.colors[barIndex % this.colors.length]

    );

    barIndex++;
}

//drawing series name
this.ctx.save();

this.ctx.textBaseline = "bottom";

this.ctx.textAlign = "center";

this.ctx.fillStyle = "#fff";

this.ctx.font = "bold 24px Arial";

this.ctx.fillText(this.options.seriesName, this.canvas.width / 2, this.canvas.height);

this.ctx.restore();

//draw legend
barIndex = 0;

var legend = document.querySelector("legend[for='myCanvas']");

var ul = document.createElement("ul");

legend.append(ul);

for (categ in this.options.data) {

    var li = document.createElement("li");

```

```

        li.style.listStyle = "none";

        li.style.borderLeft = "20px solid " + this.colors[barIndex % this.colors.length];

        li.style.padding = "5px";

        li.textContent = categ;

        ul.append(li);

        barIndex++;

    }

}

}

//just change month number

var month = parseInt(document.getElementById('month').value);

var myVinyls = {};

var data;

var months = document.getElementById('month');

$(document).ready(function () {

    //just change the name of .txt file to load file

    $.get('data.txt', function (theData) {

        theData = theData.replace(/\r/g, "");

        theData = theData.replace(/\t/g, ' ');

        theData = theData.split('\n');

        data = theData;

        totalRows = theData.length;

        theHead = theData[0].split(' ');

        theRow = theData[month].split(' ');

        for (let i = 1; i < theHead.length - 2; ++i)

            myVinyls[theHead[i]] = theRow[i];
    });

```

```

var myBarchart = new Barchart({
    canvas: myCanvas,
    seriesName: "Sales Data of " + months.options[months.selectedIndex].text,
    padding: 40,
    gridScale: 500,
    gridColor: "#eeeeee",
    data: myVinyls,
    colors: ["#a55ca5", "#67b6c7", "#bccd7a", "#eb9743", "#32a852", "#a432a8"]
});

myBarchart.draw();
});
});

function draw() {
    theData = data;
    month = parseInt(document.getElementById('month').value);
    theHead = theData[0].split(' ');
    theRow = theData[month].split(' ');
    for (let i = 1; i < theHead.length - 2; ++i)
        myVinyls[theHead[i]] = theRow[i];

    var legend = document.querySelector("legend[for='myCanvas']");
    legend.remove();

    var x = document.createElement("LEGEND");
    x.setAttribute("for", "myCanvas");

```

```

document.body.appendChild(x);

ctx.clearRect(0, 0, myCanvas.width, myCanvas.height);

var myBarchart = new Barchart({
    canvas: myCanvas,
    seriesName: "Sales Data of " + months.options[months.selectedIndex].text,
    padding: 40,
    gridScale: 500,
    gridColor: "#eeeeee",
    data: myVinyls,
    colors: ["#a55ca5", "#67b6c7", "#bccd7a", "#eb9743", "#32a852", "#a432a8"]
});

myBarchart.draw();
}
</script>
</html>

```

### **data.txt**

	month	facecream	facewash	toothpaste	bathingssoap	shampoo	moisturizer	total_units	total_profit
1	2500	1500	5200	9200	1200	1500	21100	211000	
2	2630	1200	5100	6100	2100	1200	18330	183300	
3	2140	1340	4550	9550	3550	1340	22470	224700	
4	3400	1130	5870	8870	1870	1130	22270	222700	

5 3600 1740 4560 7760 1560 1740 20960 209600  
6 2760 1555 4890 7490 1890 1555 20140 201400  
7 2980 1120 4780 8980 1780 1120 29550 295500  
8 3700 1400 5860 9960 2860 1400 36140 361400  
9 3540 1780 6100 8100 2100 1780 23400 234000  
10 1990 1890 8300 10300 2300 1890 26670 266700  
11 2340 2100 7300 13300 2400 2100 41280 412800  
12 2900 1760 7400 14400 1800 1760 30020 300200`/

**Output:**

Read Text File to Data Table

127.0.0.1:5500/demo.html

Get data from text file

month_number	facecream	facewash	toothpaste	bathingsoap	shampoo	moisturizer	total_units	total_profit
1	2500	1500	5200	9200	1200	1500	21100	211000
2	2630	1200	5100	6100	2100	1200	18330	183300
3	2140	1340	4550	9550	3550	1340	22470	224700
4	3400	1130	5870	8870	1870	1130	22270	222700
5	3600	1740	4560	7760	1560	1740	20960	209600
6	2760	1555	4890	7490	1890	1555	20140	201400
7	2980	1120	4780	8980	1780	1120	29550	295500
8	3700	1400	5860	9960	2860	1400	36140	361400
9	3540	1780	6100	8100	2100	1780	23400	234000
10	1990	1890	8300	10300	2300	1890	26670	266700
11	2340	2100	7300	13300	2400	2100	41280	412800
12	2900	1760	7400	14400	1800	1760	30020	300200`/

### 3(D) : Read the data .csv file and draw Data Table

```
<!DOCTYPE html>

<html lang="en-IN" dir="ltr">

<head>

  <title>Read CSV File to Data Table</title>

  <meta charset="utf-8">

  <meta name="author" content="SidPro" />

  <meta name="viewport" content="width=device-width,initial-scale=1.0" />

  <meta name="description" content="Data visualization using javaScript and HTML" />

  <script src="js/jquery-3.5.1.min.js"></script>

  <script src="js/jquery.csv.js"></script>

  <style>

    #mytable {

      font-family: Arial, Helvetica, sans-serif;

      border-collapse: collapse;

      width: 100%;

    }

    #mytable td,

    #mytable th {

      border: 1px solid #ddd;

      padding: 8px;

    }

    #mytable tr:nth-child(even) {

      background-color: #f2f2f2;

    }

  </style>

</head>

<body>

  <table id="mytable">

    <tr>

      <th>Name</th>

      <th>Age</th>

      <th>Gender</th>

      <th>Address</th>

    </tr>

    <tr>

      <td>John</td>

      <td>25</td>

      <td>Male</td>

      <td>123 Main St</td>

    </tr>

    <tr>

      <td>Jane</td>

      <td>30</td>

      <td>Female</td>

      <td>456 Elm St</td>

    </tr>

    <tr>

      <td>Mike</td>

      <td>35</td>

      <td>Male</td>

      <td>789 Oak St</td>

    </tr>

    <tr>

      <td>Emily</td>

      <td>28</td>

      <td>Female</td>

      <td>101 Pine St</td>

    </tr>

    <tr>

      <td>David</td>

      <td>40</td>

      <td>Male</td>

      <td>202 Cedar St</td>

    </tr>

    <tr>

      <td>Olivia</td>

      <td>22</td>

      <td>Female</td>

      <td>303 Birch St</td>

    </tr>

    <tr>

      <td>Noah</td>

      <td>38</td>

      <td>Male</td>

      <td>404 Maple St</td>

    </tr>

    <tr>

      <td>Sophia</td>

      <td>27</td>

      <td>Female</td>

      <td>505 Walnut St</td>

    </tr>

    <tr>

      <td>Liam</td>

      <td>32</td>

      <td>Male</td>

      <td>606 Cherry St</td>

    </tr>

    <tr>

      <td>Isabella</td>

      <td>29</td>

      <td>Female</td>

      <td>707 Peach St</td>

    </tr>

    <tr>

      <td>Ethan</td>

      <td>33</td>

      <td>Male</td>

      <td>808 Apple St</td>

    </tr>

    <tr>

      <td>Mia</td>

      <td>26</td>

      <td>Female</td>

      <td>909 Orange St</td>

    </tr>

    <tr>

      <td>Alexander</td>

      <td>37</td>

      <td>Male</td>

      <td>1010 Grape St</td>

    </tr>

    <tr>

      <td>Charlotte</td>

      <td>24</td>

      <td>Female</td>

      <td>1111 Lemon St</td>

    </tr>

    <tr>

      <td>William</td>

      <td>31</td>

      <td>Male</td>

      <td>1212 Lime St</td>

    </tr>

    <tr>

      <td>Amelia</td>

      <td>23</td>

      <td>Female</td>

      <td>1313 Coffee St</td>

    </tr>

    <tr>

      <td>Benjamin</td>

      <td>36</td>

      <td>Male</td>

      <td>1414 Tea St</td>

    </tr>

    <tr>

      <td>Evelyn</td>

      <td>21</td>

      <td>Female</td>

      <td>1515 Mint St</td>

    </tr>

    <tr>

      <td>Lucas</td>

      <td>34</td>

      <td>Male</td>

      <td>1616 Basil St</td>

    </tr>

    <tr>

      <td>Hannah</td>

      <td>20</td>

      <td>Female</td>

      <td>1717 Dill St</td>

    </tr>

    <tr>

      <td>Gabriel</td>

      <td>39</td>

      <td>Male</td>

      <td>1818 Parsley St</td>

    </tr>

    <tr>

      <td>Chloe</td>

      <td>25</td>

      <td>Female</td>

      <td>1919 Thyme St</td>

    </tr>

    <tr>

      <td>Isaac</td>

      <td>32</td>

      <td>Male</td>

      <td>2020 Rosemary St</td>

    </tr>

    <tr>

      <td>Grace</td>

      <td>27</td>

      <td>Female</td>

      <td>2121 Sage St</td>

    </tr>

    <tr>

      <td>Nathan</td>

      <td>30</td>

      <td>Male</td>

      <td>2222 Oregano St</td>

    </tr>

    <tr>

      <td>Lily</td>

      <td>24</td>

      <td>Female</td>

      <td>2323 Fennel St</td>

    </tr>

    <tr>

      <td>Oscar</td>

      <td>35</td>

      <td>Male</td>

      <td>2424 Chives St</td>

    </tr>

    <tr>

      <td>Victoria</td>

      <td>28</td>

      <td>Female</td>

      <td>2525 Onion St</td>

    </tr>

    <tr>

      <td>Caleb</td>

      <td>31</td>

      <td>Male</td>

      <td>2626 Garlic St</td>

    </tr>

    <tr>

      <td>Sofia</td>

      <td>22</td>

      <td>Female</td>

      <td>2727 Shallots St</td>

    </tr>

    <tr>

      <td>Elijah</td>

      <td>36</td>

      <td>Male</td>

      <td>2828 Potatoes St</td>

    </tr>

    <tr>

      <td>Ava</td>

      <td>29</td>

      <td>Female</td>

      <td>2929 Carrots St</td>

    </tr>

    <tr>

      <td>Carter</td>

      <td>33</td>

      <td>Male</td>

      <td>3030 Spinach St</td>

    </tr>

    <tr>

      <td>Mila</td>

      <td>26</td>

      <td>Female</td>

      <td>3131 Broccoli St</td>

    </tr>

    <tr>

      <td>Wyatt</td>

      <td>38</td>

      <td>Male</td>

      <td>3232 Asparagus St</td>

    </tr>

    <tr>

      <td>Elena</td>

      <td>21</td>

      <td>Female</td>

      <td>3333 Cauliflower St</td>

    </tr>

    <tr>

      <td>Grayson</td>

      <td>34</td>

      <td>Male</td>

      <td>3434 Zucchini St</td>

    </tr>

    <tr>

      <td>Scarlett</td>

      <td>27</td>

      <td>Female</td>

      <td>3535 Eggplant St</td>

    </tr>

    <tr>

      <td>Logan</td>

      <td>30</td>

      <td>Male</td>

      <td>3636 Tomatoes St</td>

    </tr>

    <tr>

      <td>Zoey</td>

      <td>23</td>

      <td>Female</td>

      <td>3737 Peppers St</td>

    </tr>

    <tr>

      <td>Caden</td>

      <td>37</td>

      <td>Male</td>

      <td>3838 Onions St</td>

    </tr>

    <tr>

      <td>Penelope</td>

      <td>25</td>

      <td>Female</td>

      <td>3939 Potatoes St</td>

    </tr>

    <tr>

      <td>Ethan</td>

      <td>32</td>

      <td>Male</td>

      <td>4040 Carrots St</td>

    </tr>

    <tr>

      <td>Luna</td>

      <td>28</td>

      <td>Female</td>

      <td>4141 Spinach St</td>

    </tr>

    <tr>

      <td>Carter</td>

      <td>35</td>

      <td>Male</td>

      <td>4242 Broccoli St</td>

    </tr>

    <tr>

      <td>Aria</td>

      <td>22</td>

      <td>Female</td>

      <td>4343 Asparagus St</td>

    </tr>

    <tr>

      <td>Mason</td>

      <td>39</td>

      <td>Male</td>

      <td>4444 Cauliflower St</td>

    </tr>

    <tr>

      <td>Sofia</td>

      <td>26</td>

      <td>Female</td>

      <td>4545 Zucchini St</td>

    </tr>

    <tr>

      <td>Lucas</td>

      <td>31</td>

      <td>Male</td>

      <td>4646 Eggplant St</td>

    </tr>

    <tr>

      <td>Mia</td>

      <td>24</td>

      <td>Female</td>

      <td>4747 Tomatoes St</td>

    </tr>

    <tr>

      <td>Nathan</td>

      <td>36</td>

      <td>Male</td>

      <td>4848 Peppers St</td>

    </tr>

    <tr>

      <td>Charlotte</td>

      <td>29</td>

      <td>Female</td>

      <td>4949 Onions St</td>

    </tr>

    <tr>

      <td>Ethan</td>

      <td>33</td>

      <td>Male</td>

      <td>5050 Potatoes St</td>

    </tr>

    <tr>

      <td>Lily</td>

      <td>27</td>

      <td>Female</td>

      <td>5151 Carrots St</td>

    </tr>

    <tr>

      <td>Oscar</td>

      <td>30</td>

      <td>Male</td>

      <td>5252 Spinach St</td>

    </tr>

    <tr>

      <td>Isabella</td>

      <td>23</td>

      <td>Female</td>

      <td>5353 Broccoli St</td>

    </tr>

    <tr>

      <td>Caleb</td>

      <td>37</td>

      <td>Male</td>

      <td>5454 Asparagus St</td>

    </tr>

    <tr>

      <td>Victoria</td>

      <td>21</td>

      <td>Female</td>

      <td>5555 Cauliflower St</td>

    </tr>

    <tr>

      <td>Grayson</td>

      <td>34</td>

      <td>Male</td>

      <td>5656 Zucchini St</td>

    </tr>

    <tr>

      <td>Scarlett</td>

      <td>27</td>

      <td>Female</td>

      <td>5757 Eggplant St</td>

    </tr>

    <tr>

      <td>Logan</td>

      <td>30</td>

      <td>Male</td>

      <td>5858 Tomatoes St</td>

    </tr>

    <tr>

      <td>Zoey</td>

      <td>23</td>

      <td>Female</td>

      <td>5959 Peppers St</td>

    </tr>

    <tr>

      <td>Caden</td>

      <td>37</td>

      <td>Male</td>

      <td>6060 Onions St</td>

    </tr>

    <tr>

      <td>Penelope</td>

      <td>25</td>

      <td>Female</td>

      <td>6161 Potatoes St</td>

    </tr>

    <tr>

      <td>Ethan</td>

      <td>32</td>

      <td>Male</td>

      <td>6262 Carrots St</td>

    </tr>

    <tr>

      <td>Luna</td>

      <td>28</td>

      <td>Female</td>

      <td>6363 Spinach St</td>

    </tr>

    <tr>

      <td>Carter</td>

      <td>35</td>

      <td>Male</td>

      <td>6464 Broccoli St</td>

    </tr>

    <tr>

      <td>Aria</td>

      <td>22</td>

      <td>Female</td>

      <td>6565 Asparagus St</td>

    </tr>

    <tr>

      <td>Mason</td>

      <td>39</td>

      <td>Male</td>

      <td>6666 Cauliflower St</td>

    </tr>

    <tr>

      <td>Sofia</td>

      <td>26</td>

      <td>Female</td>

      <td>6767 Zucchini St</td>

    </tr>

    <tr>

      <td>Lucas</td>

      <td>31</td>

      <td>Male</td>

      <td>6868 Eggplant St</td>

    </tr>

    <tr>

      <td>Mia</td>

      <td>24</td>

      <td>Female</td>

      <td>6969 Tomatoes St</td>

    </tr>

    <tr>

      <td>Nathan</td>

      <td>36</td>

      <td>Male</td>

      <td>7070 Peppers St</td>

    </tr>

    <tr>

      <td>Charlotte</td>

      <td>29</td>

      <td>Female</td>

      <td>7171 Onions St</td>

    </tr>

    <tr>

      <td>Ethan</td>

      <td>33</td>

      <td>Male</td>

      <td>7272 Potatoes St</td>

    </tr>

    <tr>

      <td>Lily</td>

      <td>27</td>

      <td>Female</td>

      <td>7373 Carrots St</td>

    </tr>

    <tr>

      <td>Oscar</td>

      <td>30</td>

      <td>Male</td>

      <td>7474 Spinach St</td>

    </tr>

    <tr>

      <td>Isabella</td>

      <td>23</td>

      <td>Female</td>

      <td>7575 Broccoli St</td>

    </tr>

    <tr>

      <td>Caleb</td>

      <td>37</td>

      <td>Male</td>

      <td>7676 Asparagus St</td>

    </tr>

    <tr>

      <td>Victoria</td>

      <td>21</td>

      <td>Female</td>

      <td>7777 Cauliflower St</td>

    </tr>

    <tr>

      <td>Grayson</td>

      <td>34</td>

      <td>Male</td>

      <td>7878 Zucchini St</td>

    </tr>

    <tr>

      <td>Scarlett</td>

      <td>27</td>

      <td>Female</td>

      <td>7979 Eggplant St</td>

    </tr>

    <tr>

      <td>Logan</td>

      <td>30</td>

      <td>Male</td>

      <td>8080 Tomatoes St</td>

    </tr>

    <tr>

      <td>Zoey</td>

      <td>23</td>

      <td>Female</td>

      <td>8181 Peppers St</td>

    </tr>

    <tr>

      <td>Caden</td>

      <td>37</td>

      <td>Male</td>

      <td>8282 Onions St</td>

    </tr>

    <tr>

      <td>Penelope</td>

      <td>25</td>

      <td>Female</td>

      <td>8383 Potatoes St</td>

    </tr>

    <tr>

      <td>Ethan</td>

      <td>32</td>

      <td>Male</td>

      <td>8484 Carrots St</td>

    </tr>

    <tr>

      <td>Luna</td>

      <td>28</td>

      <td>Female</td>

      <td>8585 Spinach St</td>

    </tr>

    <tr>

      <td>Carter</td>

      <td>35</td>

      <td>Male</td>

      <td>8686 Broccoli St</td>

    </tr>

    <tr>

      <td>Aria</td>

      <td>22</td>

      <td>Female</td>

      <td>8787 Asparagus St</td>

    </tr>

    <tr>

      <td>Mason</td>

      <td>39</td>

      <td>Male</td>

      <td>8888 Cauliflower St</td>

    </tr>

    <tr>

      <td>Sofia</td>

      <td>26</td>

      <td>Female</td>

      <td>8989 Zucchini St</td>

    </tr>

    <tr>

      <td>Lucas</td>

      <td>31</td>

      <td>Male</td>

      <td>9090 Eggplant St</td>

    </tr>

    <tr>

      <td>Mia</td>

      <td>24</td>

      <td>Female</td>

      <td>9191 Tomatoes St</td>

    </tr>

    <tr>

      <td>Nathan</td>

      <td>36</td>

      <td>Male</td>

      <td>9292 Peppers St</td>

    </tr>

    <tr>

      <td>Charlotte</td>

      <td>29</td>

      <td>Female</td>

      <td>9393 Onions St</td>

    </tr>

    <tr>

      <td>Ethan</td>

      <td>33</td>

      <td>Male</td>

      <td>9494 Potatoes St</td>

    </tr>

    <tr>

      <td>Lily</td>

      <td>27</td>

      <td>Female</td>

      <td>9595 Carrots St</td>

    </tr>

    <tr>

      <td>Oscar</td>

      <td>30</td>

      <td>Male</td>

      <td>9696 Spinach St</td>

    </tr>

    <tr>

      <td>Isabella</td>

      <td>23</td>

      <td>Female</td>

      <td>9797 Broccoli St</td>

    </tr>

    <tr>

      <td>Caleb</td>

      <td>37</td>

      <td>Male</td>

      <td>9898 Asparagus St</td>

    </tr>

    <tr>

      <td>Victoria</td>

      <td>21</td>

      <td>Female</td>

      <td>9999 Cauliflower St</td>

    </tr>

  </table>

</body>

</html>
```

```
#mytable tr:hover {
    background-color: #ddd;
}

#mytable th {
    padding-top: 12px;
    padding-bottom: 12px;
    text-align: left;
    background-color: cornflowerblue;
    color: white;
}

</style>
</head>

<body>
    <h1>Get data from CSV file</h1>
    <table id="mytable">
    </table>
</body>
<script>
    var content = "";
    $(document).ready(function () {
        //just change the name of .csv file to load file
        $.get('data/data.csv', function (theData) {
            console.log(theData);
            theData = theData.replace(/"/g, "");
            console.log(theData);
        });
    });
</script>
```



```

    theData = theData.split(/\r?\n\r/);

    console.log(theData);

    totalRows = theData.length;


    theHead = theData[0].split(',');

    content += "<tr>";

    theHead.forEach(TH);

    content += "</tr>";


    for (let i = 1; i < totalRows; ++i) {

        theTD = theData[i].split(',');

        content += "<tr>";

        theTD.forEach(TD);

        content += "</tr>";

    }

    $('#mytable').html(content);


    });

});


function TH(value) {

    content += "<th>" + value + "</th>";

}

function TD(value) {

```

```

        content += "<td>" + value + "</td>";

    }

</script>

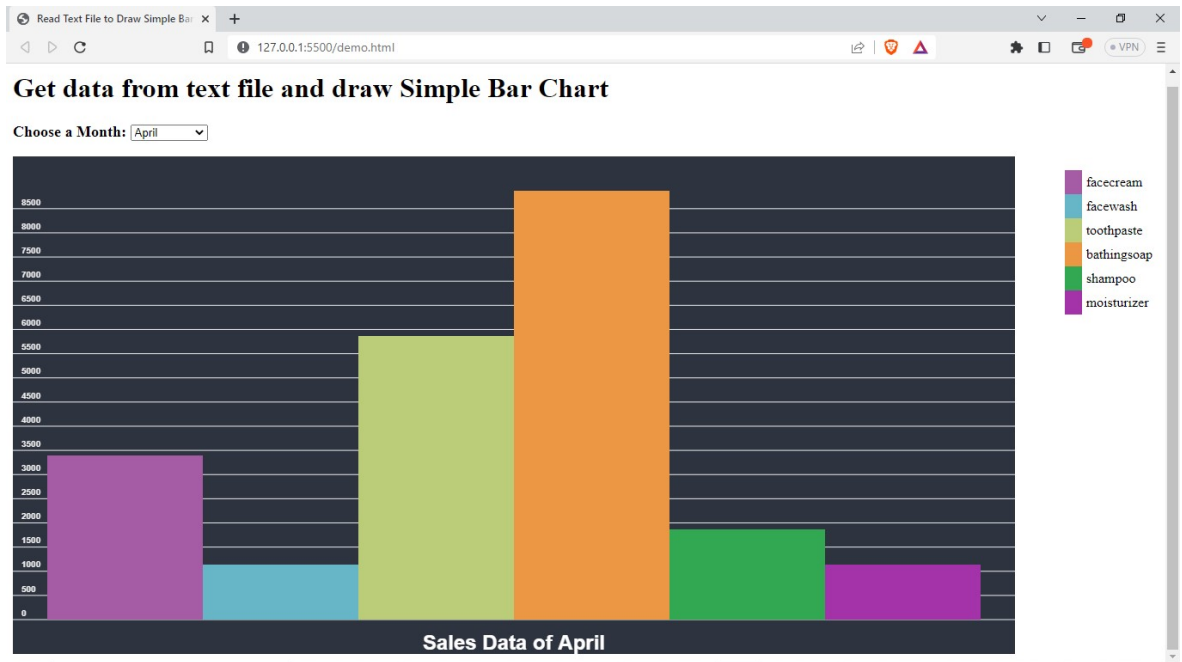
</html>

```

### **Data.txt**

month_number	facecream		facewash		toothpaste		bathingssoap		shampoo
	moisturizer		total_units		total_profit				
1	2500	1500	5200	9200	1200	1500	21100	211000	
2	2630	1200	5100	6100	2100	1200	18330	183300	
3	2140	1340	4550	9550	3550	1340	22470	224700	
4	3400	1130	5870	8870	1870	1130	22270	222700	
5	3600	1740	4560	7760	1560	1740	20960	209600	
6	2760	1555	4890	7490	1890	1555	20140	201400	
7	2980	1120	4780	8980	1780	1120	29550	295500	
8	3700	1400	5860	9960	2860	1400	36140	361400	
9	3540	1780	6100	8100	2100	1780	23400	234000	
10	1990	1890	8300	10300	2300	1890	26670	266700	
11	2340	2100	7300	13300	2400	2100	41280	412800	
12	2900	1760	7400	14400	1800	1760	30020	300200	

**Output:**



### 3(E) : Read the data .csv file and draw Column Bar Chart

```
<!DOCTYPE html>

<html lang="en-IN" dir="ltr">

<head>

  <title>Read CSV File to Draw Simple Column Bar Chart</title>

  <meta charset="utf-8">

  <meta name="author" content="SidPro" />

  <meta name="viewport" content="width=device-width,initial-scale=1.0" />

  <meta name="description" content="Data visualization using javaScript and HTML" />

  <script src="js/jquery-3.5.1.min.js"></script>

  <style>

    #mytable {

      font-family: Arial, Helvetica, sans-serif;

      border-collapse: collapse;

      width: 100%;

    }

    #mytable td,

    #mytable th {

      border: 1px solid #ddd;

      padding: 8px;

    }

    #mytable tr:nth-child(even) {

      background-color: #f2f2f2;

    }

  </style>

</head>

<body>

  <table id="mytable">

    <tr>

      <th>Name</th>

      <th>Age</th>

      <th>Gender</th>

      <th>City</th>

      <th>Country</th>

    </tr>

    <tr>

      <td>John</td>

      <td>25</td>

      <td>Male</td>

      <td>New York</td>

      <td>USA</td>

    </tr>

    <tr>

      <td>Jane</td>

      <td>30</td>

      <td>Female</td>

      <td>Los Angeles</td>

      <td>USA</td>

    </tr>

    <tr>

      <td>Mike</td>

      <td>22</td>

      <td>Male</td>

      <td>Chicago</td>

      <td>USA</td>

    </tr>

    <tr>

      <td>Emily</td>

      <td>28</td>

      <td>Female</td>

      <td>San Francisco</td>

      <td>USA</td>

    </tr>

    <tr>

      <td>David</td>

      <td>35</td>

      <td>Male</td>

      <td>London</td>

      <td>UK</td>

    </tr>

    <tr>

      <td>Sophia</td>

      <td>20</td>

      <td>Female</td>

      <td>Paris</td>

      <td>France</td>

    </tr>

    <tr>

      <td>Daniel</td>

      <td>27</td>

      <td>Male</td>

      <td>Berlin</td>

      <td>Germany</td>

    </tr>

    <tr>

      <td>Olivia</td>

      <td>32</td>

      <td>Female</td>

      <td>Tokyo</td>

      <td>Japan</td>

    </tr>

    <tr>

      <td>Liam</td>

      <td>24</td>

      <td>Male</td>

      <td>Sydney</td>

      <td>Australia</td>

    </tr>

    <tr>

      <td>Ava</td>

      <td>29</td>

      <td>Female</td>

      <td>Melbourne</td>

      <td>Australia</td>

    </tr>

    <tr>

      <td>Noah</td>

      <td>21</td>

      <td>Male</td>

      <td>Auckland</td>

      <td>New Zealand</td>

    </tr>

    <tr>

      <td>Isabella</td>

      <td>31</td>

      <td>Female</td>

      <td>Wellington</td>

      <td>New Zealand</td>

    </tr>

  </table>

  <div>

    <img alt="A column bar chart showing the age distribution of the data. The x-axis represents the age groups (20-25, 26-30, 31-35, 36-40, 41-45, 46-50, 51-55, 56-60, 61-65, 66-70, 71-75, 76-80, 81-85, 86-90, 91-95, 96-100) and the y-axis represents the count of individuals in each age group. The bars are colored in a light blue shade." data-bbox="147 116 822 868"/>

  </div>

</body>

</html>
```

```
#mytable tr:hover {  
    background-color: #ddd;  
}
```

```
#mytable th {  
    padding-top: 12px;  
    padding-bottom: 12px;  
    text-align: left;  
    background-color: cornflowerblue;  
    color: white;  
}
```

```
legend {  
    float: right;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>Get data from CSV file and draw Column Simple Bar Chart</h1>
```

```
<label style="font-size:18px" for="month"><b>Choose a Month: </b></label>
```

```
<select id="month" onchange="draw()">
```

```
<option value="1">January</option>
```

```
<option value="2" selected>February</option>
```

```
<option value="3">March</option>
```

```
<option value="4">April</option>
```

```
<option value="5">May</option>
```

```
<option value="6">June</option>
<option value="7">July</option>
<option value="8">August</option>
<option value="9">September</option>
<option value="10">October</option>
<option value="11">November</option>
<option value="12">December</option>
</select><br><br>
```

```
<canvas id="myCanvas" style="background: #86A09E;"></canvas>
<legend for="myCanvas"></legend>
```

```
</body>
```

```
<script>
```

```
var myCanvas = document.getElementById("myCanvas");
myCanvas.width = window.innerWidth - 200;
myCanvas.height = window.innerHeight - 120;
```

```
var ctx = myCanvas.getContext("2d");
```

```
function drawLine(ctx, startX, startY, endX, endY, color) {
    ctx.save();
    ctx.strokeStyle = color;
    ctx.beginPath();
    ctx.moveTo(startX, startY);
    ctx.lineTo(endX, endY);
    ctx.stroke();
    ctx.restore();
}
```

```
}
```

```
function drawBar(ctx, upperLeftCornerX, upperLeftCornerY, width, height, color) {  
    ctx.save();  
    ctx.fillStyle = color;  
    ctx.fillRect(upperLeftCornerX, upperLeftCornerY, width, height);  
    ctx.restore();  
}
```

```
var Barchart = function (options) {  
    this.options = options;  
    this.canvas = options.canvas;  
    this.ctx = this.canvas.getContext("2d");  
    this.colors = options.colors;  
  
    this.draw = function () {  
        var maxValue = 0;  
        for (var categ in this.options.data) {  
            maxValue = Math.max(maxValue, this.options.data[categ]);  
        }  
        var canvasActualHeight = this.canvas.height - this.options.padding * 2;  
        var canvasActualWidth = this.canvas.width - this.options.padding * 2;  
  
        //drawing the grid lines  
        var gridValue = 0;  
        while (gridValue <= maxValue) {
```

```
        var gridY = canvasActualHeight * (1 - gridValue / maxValue) +  
this.options.padding;
```

```
        drawLine(  
            this.ctx,  
            0,  
            gridY,  
            this.canvas.width,  
            gridY,  
            this.options.gridColor  
        );
```

```
        //writing grid markers
```

```
        this.ctx.save();  
        this.ctx.fillStyle = this.options.gridColor;  
        this.ctx.textBaseline = "bottom";  
        this.ctx.font = "bold 10px Arial";  
        this.ctx.fillText(gridValue, 10, gridY - 2);  
        this.ctx.restore();
```

```
        gridValue += this.options.gridScale;
```

```
    }
```

```
    //drawing the bars
```

```
    var barIndex = 0;  
    var numberOfBars = Object.keys(this.options.data).length;  
    var barSize = (canvasActualWidth) / numberOfBars;
```



```

for (categ in this.options.data) {
    var val = this.options.data[categ];
    var barHeight = Math.round(canvasActualHeight * val / maxValue);
    drawBar(
        this.ctx,
        this.options.padding + barIndex * barSize,
        this.canvas.height - barHeight - this.options.padding,
        barSize,
        barHeight,
        this.colors[barIndex % this.colors.length]
    );

    barIndex++;
}

//drawing series name
this.ctx.save();
this.ctx.textBaseline = "bottom";
this.ctx.textAlign = "center";
this.ctx.fillStyle = "#fff";
this.ctx.font = "bold 24px Arial";
this.ctx.fillText(this.options.seriesName, this.canvas.width / 2, this.canvas.height);
this.ctx.restore();

//draw legend
barIndex = 0;
var legend = document.querySelector("legend[for='myCanvas']");

```

```

var ul = document.createElement("ul");

legend.append(ul);

for (categ in this.options.data) {

    var li = document.createElement("li");

    li.style.listStyle = "none";

    li.style.borderLeft = "20px solid " + this.colors[barIndex % this.colors.length];

    li.style.padding = "5px";

    li.textContent = categ;

    ul.append(li);

    barIndex++;

}

}

}

//just change month number

var month = parseInt(document.getElementById('month').value);

var myVinyls = {};

var data;

var months = document.getElementById('month');

$(document).ready(function () {

    //just change the name of .txt file to load file

    $.get('data/data.csv', function (theData) {

        theData = theData.replace(/"/g, "");

        theData = theData.split(/\r?\n\r/);

        console.log(theData);

        totalRows = theData.length;

        data = theData;

```

```

totalRows = theData.length;

theHead = theData[0].split(',');

theRow = theData[month].split(',');

for (let i = 1; i < theHead.length - 2; ++i)
    myVinyls[theHead[i]] = theRow[i];

var myBarchart = new Barchart({
    canvas: myCanvas,
    seriesName: "Sales Data of " + months.options[months.selectedIndex].text,
    padding: 40,
    gridScale: 500,
    gridColor: "#eeeeee",
    data: myVinyls,
    colors: ["#2A9D8F", "#E9C46A", "#F4A261", "#30BCED", "#D9BA41",
"#7D001A"]
});

myBarchart.draw();

});

});

function draw() {
    theData = data;

    month = parseInt(document.getElementById('month').value);

    theHead = theData[0].split(',');

    theRow = theData[month].split(',');

    for (let i = 1; i < theHead.length - 2; ++i)
        myVinyls[theHead[i]] = theRow[i];

    var legend = document.querySelector("legend[for='myCanvas']");

```

```
legend.remove();

var x = document.createElement("LEGEND");

x.setAttribute("for", "myCanvas");

document.body.appendChild(x);

ctx.clearRect(0, 0, myCanvas.width, myCanvas.height);

var myBarchart = new Barchart({

    canvas: myCanvas,

    seriesName: "Sales Data of " + months.options[months.selectedIndex].text,

    padding: 40,

    gridScale: 500,

    gridColor: "#eeeeee",

    data: myVinyls,

    colors: ["#2A9D8F", "#E9C46A", "#F4A261", "#30BCED", "#D9BA41",
"#7D001A"]

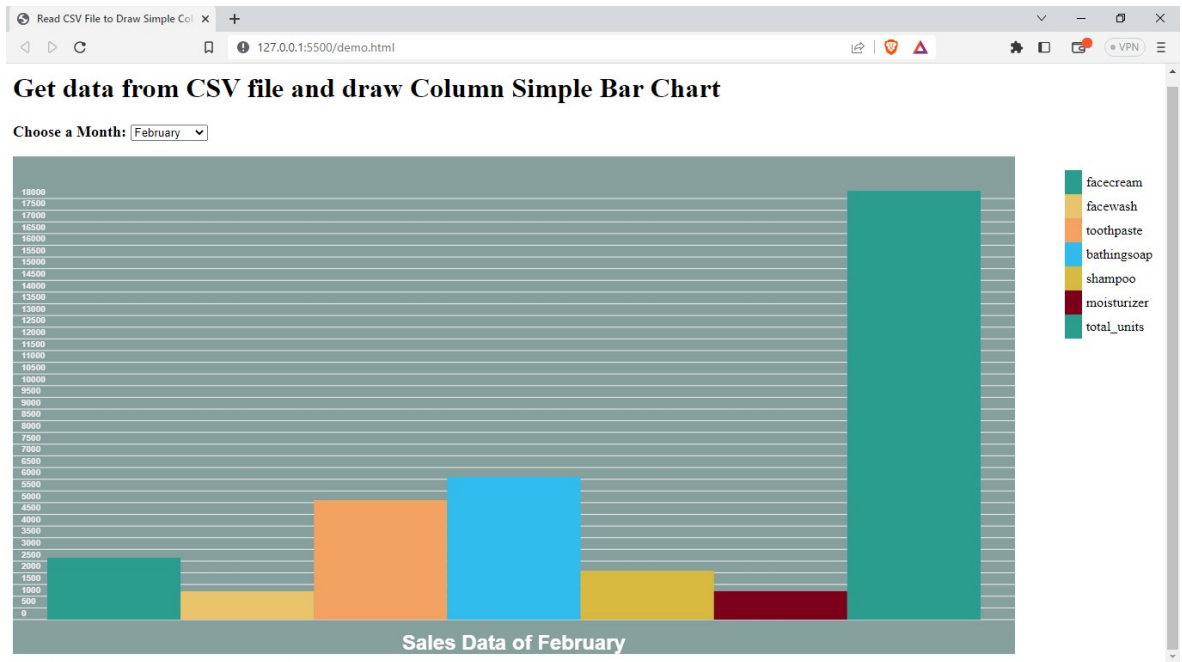
});

myBarchart.draw();}

</script>

</html>
```

**Output:**



### 3(F) : Read the data XML file and draw Data Table

```
<!DOCTYPE html>

<html lang="en-IN" dir="ltr">

<head>

    <title>Read XML File to Data Table</title>

    <meta charset="utf-8">

    <meta name="author" content="SidPro" />

    <meta name="viewport" content="width=device-width,initial-scale=1.0" />

    <meta name="description" content="Data visualization using javaScript and HTML" />

    <script src="js/jquery-3.5.1.min.js"></script>

    <script src="js/jquery.csv.js"></script>

    <style>

        #mytable {

            font-family: Arial, Helvetica, sans-serif;

            border-collapse: collapse;

            width: 100%;

        }

        #mytable td,

        #mytable th {

            border: 1px solid #ddd;

            padding: 8px;

        }

        #mytable tr:nth-child(even) {

            background-color: #f2f2f2;

        }
```

```
#mytable tr:hover {  
    background-color: #ddd;  
}
```

```
#mytable th {  
    padding-top: 12px;  
    padding-bottom: 12px;  
    text-align: left;  
    background-color: cornflowerblue;  
    color: white;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>Get data from XML file</h1>
```

```
<table id="mytable">
```

```
<tr>
```

```
<th>Territory</th>
```

```
<th>Employees</th>
```

```
<th>Sales</th>
```

```
<th>Year</th>
```

```
</tr>
```

```
</table>
```

```
<script>
```

```
$(document).ready(function () {
```

```

$.ajax({
    type: "GET",
    url: "data/regional_sales.xml",
    dataType: "xml",
    success: xmlParser
});

});

function xmlParser(xml) {
    $(xml).find('region').each(function () {
        var theTerritory = $(this).find('territory').text();
        var numEmployees = $(this).find('employees').text();
        var theAmount = $(this).find('amount').text();

        $('#mytable').append('<tr><td>' + theTerritory + '</td><td>' + numEmployees +
        '</td><td>' + theAmount + '</td><td>' + '2013' + '</td></tr>');

    });
}

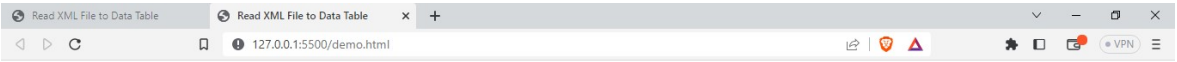
</script>

</html>

```

**Output:**





**Get data from XML file**

Territory	Employees	Sales	Year
Northeast	150	\$115,000	2013
Southeast	125	\$95,000	2013
Midwest	225	\$195,000	2013
West	325	\$265,000	2013



### 3(G) : Read the data XML file and draw Simple Chart

```
<!DOCTYPE html>

<html lang="en-IN" dir="ltr">

<head>

  <title>Read XML File to Draw Simple Bar Chart</title>

  <meta charset="utf-8">

  <meta name="author" content="SidPro" />

  <meta name="viewport" content="width=device-width,initial-scale=1.0" />

  <meta name="description" content="Data visualization using javaScript and HTML" />

  <script src="js/jquery-3.5.1.min.js"></script>

  <style>

    #mytable {

      font-family: Arial, Helvetica, sans-serif;

      border-collapse: collapse;

      width: 100%;

    }

    body {

      background: cornflowerblue;

    }

    #mytable td,

    #mytable th {

      border: 1px solid #ddd;

      padding: 8px;

    }

    #mytable tr:nth-child(even) {

      background-color: #f2f2f2;
```

```
}
```

```
#mytable tr:hover {  
    background-color: #ddd;  
}
```

```
#mytable th {  
    padding-top: 12px;  
    padding-bottom: 12px;  
    text-align: left;  
    background-color: cornflowerblue;  
    color: white;  
}
```

```
legend {  
    float: right;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>Get data from XML file and draw Simple Bar Chart</h1>
```

```
<canvas id="myCanvas" style="background: #2e3340;"></canvas>
```

```
<legend for="myCanvas"></legend>
```

```
</body>
```

```
<script>
```

```
var myCanvas = document.getElementById("myCanvas");
```

```
myCanvas.width = window.innerWidth - 200;
myCanvas.height = window.innerHeight - 120;
var ctx = myCanvas.getContext("2d");
function drawLine(ctx, startX, startY, endX, endY, color) {
    ctx.save();
    ctx.strokeStyle = color;
    ctx.beginPath();
    ctx.moveTo(startX, startY);
    ctx.lineTo(endX, endY);
    ctx.stroke();
    ctx.restore();
}
```

```
function drawBar(ctx, upperLeftCornerX, upperLeftCornerY, width, height, color) {
    ctx.save();
    ctx.fillStyle = color;
    ctx.fillRect(upperLeftCornerX, upperLeftCornerY, width, height);
    ctx.restore();
}
```

```
var Barchart = function (options) {
    this.options = options;
    this.canvas = options.canvas;
    this.ctx = this.canvas.getContext("2d");
    this.colors = options.colors;
```

```

this.draw = function () {
    var maxValue = 0;
    for (var categ in this.options.data) {
        maxValue = Math.max(maxValue, this.options.data[categ]);
    }
    var canvasActualHeight = this.canvas.height - this.options.padding * 2;
    var canvasActualWidth = this.canvas.width - this.options.padding * 2;

    //drawing the grid lines
    var gridValue = 0;
    while (gridValue <= maxValue) {
        var gridY = canvasActualHeight * (1 - gridValue / maxValue) +
this.options.padding;
        drawLine(
            this.ctx,
            0,
            gridY,
            this.canvas.width,
            gridY,
            this.options.gridColor
        );

        //writing grid markers
        this.ctx.save();
        this.ctx.fillStyle = this.options.gridColor;
        this.ctx.textBaseline = "bottom";
        this.ctx.font = "bold 10px Arial";
    }
}

```

```

        this.ctx.fillText(gridValue, 10, gridY - 2);

        this.ctx.restore();

        gridValue += this.options.gridScale;
    }

    //drawing the bars

    var barIndex = 0;

    var numberOfBars = Object.keys(this.options.data).length;

    var barSize = (canvasActualWidth) / numberOfBars;

    for (categ in this.options.data) {

        var val = this.options.data[categ];

        var barHeight = Math.round(canvasActualHeight * val / maxValue);

        drawBar(

            this.ctx,

            this.options.padding + barIndex * barSize,

            this.canvas.height - barHeight - this.options.padding,

            barSize,

            barHeight,

            this.colors[barIndex % this.colors.length]

        );

        barIndex++;

    }

    //drawing series name

```

```

this.ctx.save();

this.ctx.textBaseline = "bottom";

this.ctx.textAlign = "center";

this.ctx.fillStyle = "#fff";

this.ctx.font = "bold 24px Arial";

this.ctx.fillText(this.options.seriesName, this.canvas.width / 2, this.canvas.height);

this.ctx.restore();


//draw legend

barIndex = 0;

var legend = document.querySelector("legend[for='myCanvas']");

var ul = document.createElement("ul");

legend.append(ul);

for (categ in this.options.data) {

    var li = document.createElement("li");

    li.style.listStyle = "none";

    li.style.borderLeft = "20px solid " + this.colors[barIndex % this.colors.length];

    li.style.padding = "5px";

    li.textContent = categ;

    ul.append(li);

    barIndex++;

}

}

}

var myVinyls = {};

var data;

$(document).ready(function () {

```

```

$.ajax({
    type: "GET",
    url: "data/regional_sales.xml",
    dataType: "xml",
    success: xmlParser
});

});

function xmlParser(xml) {
    $(xml).find('region').each(function () {
        var theTerritory = $(this).find('territory').text();
        var numEmployees = $(this).find('employees').text();
        var theAmount = $(this).find('amount').text();
        theAmount = theAmount.replace(/,/g, "");
        myVinyls[theTerritory] = parseInt(theAmount);
        console.log(parseInt(theAmount));
        console.log(theTerritory);
    });
}

var myBarchart = new Barchart({
    canvas: myCanvas,
    seriesName: "Regional Sales Data",
    padding: 50,
    gridScale: 50000,
    gridColor: "#eeeeee",
    data: myVinyls,

```



```

        colors: ["#2A9D8F", "#E9C46A", "#F4A261", "#30BCED", "#D9BA41",
"#7D001A"]

    });

    myBarchart.draw();

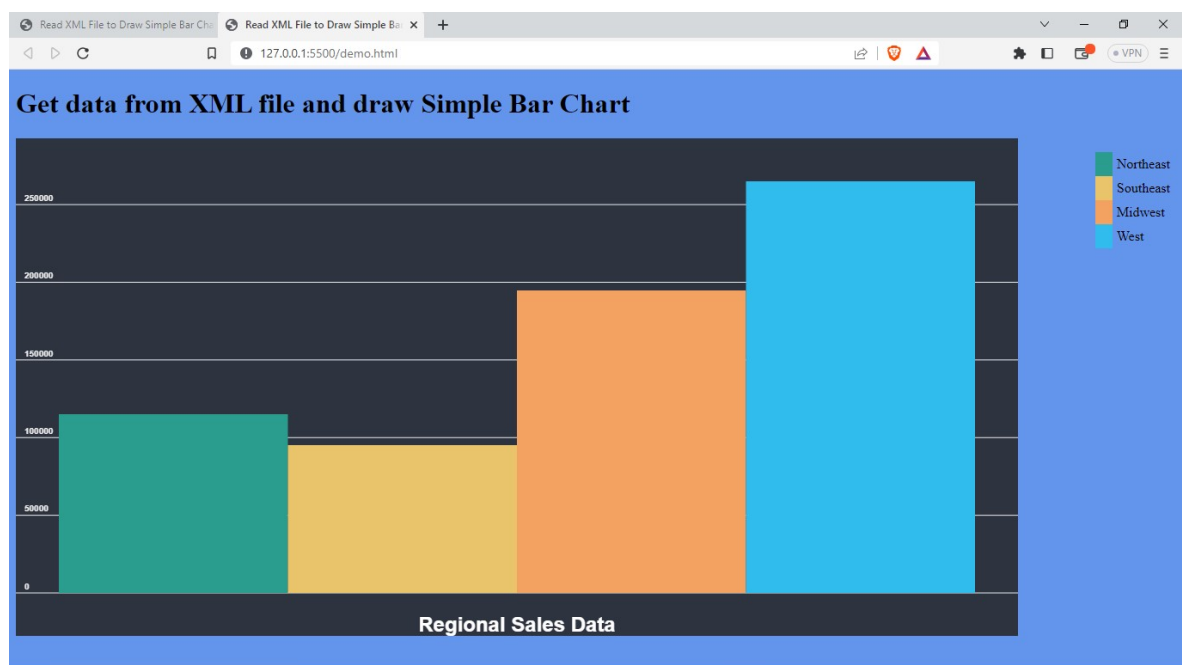
}

</script>

</html>

```

## Output:



### 3(H) : Read JSON Data and draw Data Table

```
<!DOCTYPE html>
```

```
<html lang="en-IN" dir="ltr">
```

```
<head>
```

```
<title>Read JSON File to Data Table</title>
```

```
<meta charset="utf-8">
```

```
<meta name="author" content="SidPro" />
```

```
<meta name="viewport" content="width=device-width,initial-scale=1.0" />
```

```
<meta name="description" content="Data visualization using javaScript and HTML" />
```

```
<script src="js/jquery-3.5.1.min.js"></script>
```

```
<script src="js/jquery.csv.js"></script>
```

```
<style>
```

```
#mytable {  
    font-family: Arial, Helvetica, sans-serif;  
    border-collapse: collapse;  
    width: 100%;  
}
```

```
#mytable td,
```

```
#mytable th {  
    border: 1px solid #ddd;  
    padding: 8px;  
}
```

```
#mytable tr:nth-child(even) {  
    background-color: #f2f2f2;
```

```
}
```

```
#mytable tr:hover {  
    background-color: #ddd;  
}
```

```
#mytable th {  
    padding-top: 12px;  
    padding-bottom: 12px;  
    text-align: left;  
    background-color: cornflowerblue;  
    color: white;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>Get data from JSON file</h1>
```

```
<table id="mytable">
```

```
</table>
```

```
</body>
```

```
<script>
```

```
var content = "";
```

```
$(document).ready(function () {
```

```
    $.ajax({
```

```
        type: "Get",
```

```
        url: "data/regional_sales.json",
```

```

dataType: "json",

success: function (data) {

    var theData = data.sales.region;

    var theHead = Object.keys(theData[0]);


    content += "<tr>";

    theHead.forEach(TH);

    content += "</tr>";


    theData.forEach(function (v) {

        v = Object.values(v);

        content += "<tr>";

        v.forEach(TD);

        content += "</tr>";

    });


    $('#mytable').html(content);

},

error: function () {

    alert("json not found");

}

});

});

function TH(value) {

    content += "<th>" + value.toUpperCase() + "</th>";

}

```

```
function TD(value) {  
    content += "<td>" + value + "</td>";  
}  
  
</script>  
  
</html>
```

**Output:**



The screenshot shows a web browser window with two tabs, both titled "Read JSON File to Data Table". The address bar displays the URL "127.0.0.1:5500/demo.html". Below the browser window, the text "Get data from JSON file" is displayed. Underneath this text is a table with four columns: "YEAR", "TERRITORY", "EMPLOYEES", and "AMOUNT". The table contains four rows of data for the year 2013, representing different territories: Northeast, Southeast, Midwest, and West, with their respective employee counts and amounts.

YEAR	TERRITORY	EMPLOYEES	AMOUNT
2013	Northeast	150	115,000
2013	Southeast	125	95,000
2013	Midwest	225	195,000
2013	West	325	265,000

### 3(I) : Read JSON Data and draw Simple Chart

```
<!DOCTYPE html>

<html lang="en-IN" dir="ltr">

<head>

  <title>Read JSON File to Draw Simple Bar Chart</title>

  <meta charset="utf-8">

  <meta name="author" content="SidPro" />

  <meta name="viewport" content="width=device-width,initial-scale=1.0" />

  <meta name="description" content="Data visualization using javaScript and HTML" />

  <script src="js/jquery-3.5.1.min.js"></script>

  <style>

    #mytable {

      font-family: Arial, Helvetica, sans-serif;

      border-collapse: collapse;

      width: 100%;

    }

    body {

      background: #e4fde1;

    }

    #mytable td,

    #mytable th {

      border: 1px solid #ddd;

      padding: 8px;

    }
```

```
#mytable tr:nth-child(even) {  
    background-color: #f2f2f2;  
}
```

```
#mytable tr:hover {  
    background-color: #ddd;  
}
```

```
#mytable th {  
    padding-top: 12px;  
    padding-bottom: 12px;  
    text-align: left;  
    background-color: cornflowerblue;  
    color: white;  
}
```

```
legend {  
    float: right;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>Get data from JSON file and draw Simple Bar Chart</h1>
```

```
<canvas id="myCanvas" style="background: #2e3340;"></canvas>
```

```
<legend for="myCanvas"></legend>
```

```
</body>
```

```
<script>
```

```
var myCanvas = document.getElementById("myCanvas");
```

```
myCanvas.width = window.innerWidth - 200;
```

```
myCanvas.height = window.innerHeight - 120;
```

```
var ctx = myCanvas.getContext("2d");
```

```
function drawLine(ctx, startX, startY, endX, endY, color) {
```

```
    ctx.save();
```

```
    ctx.strokeStyle = color;
```

```
    ctx.beginPath();
```

```
    ctx.moveTo(startX, startY);
```

```
    ctx.lineTo(endX, endY);
```

```
    ctx.stroke();
```

```
    ctx.restore();
```

```
}
```

```
function drawBar(ctx, upperLeftCornerX, upperLeftCornerY, width, height, color) {
```

```
    ctx.save();
```

```
    ctx.fillStyle = color;
```

```
    ctx.fillRect(upperLeftCornerX, upperLeftCornerY, width, height);
```

```
    ctx.restore();
```

```
}
```

```
var Barchart = function (options) {
```

```
    this.options = options;
```

```
    this.canvas = options.canvas;
```



```
this.ctx = this.canvas.getContext("2d");

this.colors = options.colors;


this.draw = function () {

    var maxValue = 0;

    for (var categ in this.options.data) {

        maxValue = Math.max(maxValue, this.options.data[categ]);

    }

    var canvasActualHeight = this.canvas.height - this.options.padding * 2;

    var canvasActualWidth = this.canvas.width - this.options.padding * 2;

    //drawing the grid lines

    var gridValue = 0;

    while (gridValue <= maxValue) {

        var gridY = canvasActualHeight * (1 - gridValue / maxValue) +
this.options.padding;

        drawLine(

            this.ctx,

            0,

            gridY,

            this.canvas.width,

            gridY,

            this.options.gridColor

        );

        //writing grid markers

        this.ctx.save();

        this.ctx.fillStyle = this.options.gridColor;

        this.ctx.textBaseline = "bottom";
```

```
this.ctx.font = "bold 10px Arial";  
this.ctx.fillText(gridValue, 10, gridY - 2);  
this.ctx.restore();
```

```
gridValue += this.options.gridScale;  
}
```

/drawing the bars

```
var barIndex = 0;  
var numberOfBars = Object.keys(this.options.data).length;  
var barSize = (canvasActualWidth) / numberOfBars;  
  
for (categ in this.options.data) {  
    var val = this.options.data[categ];  
    var barHeight = Math.round(canvasActualHeight * val / maxValue);  
    drawBar(  
        this.ctx,  
        this.options.padding + barIndex * barSize,  
        this.canvas.height - barHeight - this.options.padding,  
        barSize,  
        barHeight,  
        this.colors[barIndex % this.colors.length]  
    );  
    barIndex++;  
}  
  
//drawing series name  
this.ctx.save();  
this.ctx.textBaseline = "bottom";
```

```

        this.ctx.textAlign = "center";

        this.ctx.fillStyle = "#fff";

        this.ctx.font = "bold 24px Arial";

        this.ctx.fillText(this.options.seriesName, this.canvas.width / 2, this.canvas.height);

        this.ctx.restore();

        //draw legend

        barIndex = 0;

        var legend = document.querySelector("legend[for='myCanvas']");

        var ul = document.createElement("ul");

        legend.append(ul);

        for (categ in this.options.data) {

            var li = document.createElement("li");

            li.style.listStyle = "none";

            li.style.borderLeft = "20px solid " + this.colors[barIndex % this.colors.length];

            li.style.padding = "5px";

            li.textContent = categ;

            ul.append(li);

            barIndex++;

        }

    }

}

var myVinyls = {};

var data;

$(document).ready(function () {

    $.ajax({

        type: "GET",

```

```

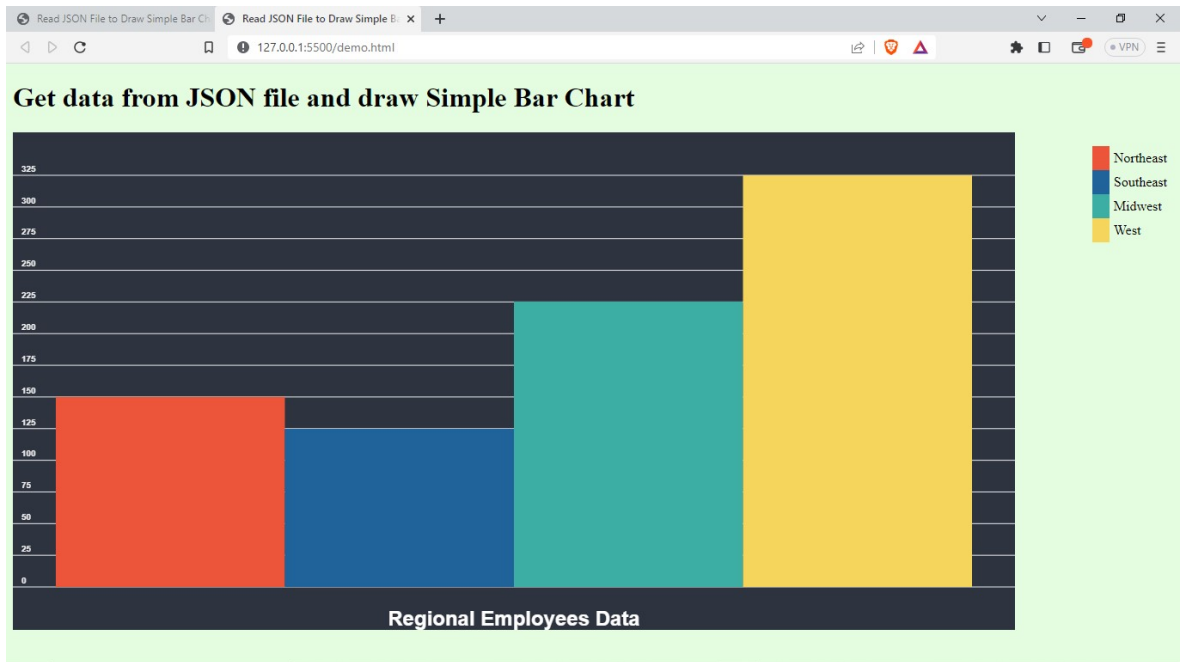
        url: "data/regional_sales.json",
        dataType: "json",
        success: jsonParser
    });
});

function jsonParser(json) {
    var theData = json.sales.region;
    theData.forEach(function (v) {
        myVinyls[v.territory] = parseInt(v.employees);
    });

    var myBarchart = new Barchart({
        canvas: myCanvas,
        seriesName: "Regional Employees Data",
        padding: 50,
        gridScale: 25,
        gridColor: "#eeeeee",
        data: myVinyls,
        colors: ["#ED553B", "#20639B", "#3CAEA3", "#F6D55C", "#173F5F", "#7D001A"]
    });

    myBarchart.draw();
}
</script>
</html>

```



## PRACTICAL:4

### 4(A) : Showing the data as a column chart (simple) Using HTML5 and D3.js and Canvas.js

```
<!DOCTYPE HTML>

<html>

<head>

  <title>Showing the Data as a column chart using D3.js</title>

  <meta charset="utf-8">

  <meta name="author" content="SidPro" />

  <meta name="viewport" content="width=device-width,initial-scale=1.0" />

  <meta name="description" content="Data visualization using javaScript and HTML" />

  <script src="js/jquery-3.5.1.min.js"></script>

  <script src="js/d3.v5.min.js"></script>

  <style>

    body {

      font-family: 'Open Sans', sans-serif;

    }

    div#layout {

      text-align: center;

    }

    div#container {

      width: 1000px;

      height: 600px;

      margin: auto;

      background-color: #2F4A6D;
```

```
}
```

```
svg {  
  width: 100%;  
  height: 100%;  
}
```

```
.bar {  
  fill: #80cbc4;  
}
```

```
text {  
  font-size: 12px;  
  fill: #fff;  
}
```

```
path {  
  stroke: gray;  
}
```

```
line {  
  stroke: gray;  
}
```

```
line#limit {  
  stroke: #FED966;  
  stroke-width: 3;  
  stroke-dasharray: 3 6;
```

```
}
```

```
.grid path {  
  stroke-width: 0;  
}
```

```
.grid .tick line {  
  stroke: #9FAAAE;  
  stroke-opacity: 0.3;  
}
```

```
text.divergence {  
  font-size: 14px;  
  fill: #2F4A6D;  
}
```

```
text.value {  
  font-size: 14px;  
}
```

```
text.title {  
  font-size: 22px;  
  font-weight: 600;  
}
```

```
text.label {  
  font-size: 14px;
```



```

        font-weight: 400;
    }

    text.source {
        font-size: 10px;
    }
</style>
</head>

<body>

    <h2>Showing the Data as a column chart using D3.js </h2>

    <div id='layout'>

        <div id='container'>

            <svg />

        </div>

    </div>

</body>

<script>

    var month = 2;

    var data = [];

    var data;

    $(document).ready(function () {

        $.get('data/data.csv', function (theData) {

            theData = theData.replace(/"/g, "");

            theData = theData.split(/\r?\n\r/);

            totalRows = theData.length;

```

```

totalRows = theData.length;
theHead = theData[0].split(',');
theRow = theData[month].split(',');
for (let i = 1; i < theHead.length - 2; ++i) {
    myVinyls = {};
    myVinyls["saleType"] = theHead[i];
    myVinyls["saleAmount"] = parseInt(theRow[i]);
    data.push(myVinyls);
}
console.log(data);

```

[//https://blog.risingstack.com/d3-js-tutorial-bar-charts-with-javascript/](https://blog.risingstack.com/d3-js-tutorial-bar-charts-with-javascript/)

```

const margin = 80; // margin value which gives a little extra padding to the chart
const width = 1000 - 2 * margin;
const height = 600 - 2 * margin;

```

```

const svg = d3.select('svg');
const svgContainer = d3.select('#container');
// Padding can be applied with a <g> element translated by the desired value.
const chart = svg.append('g').attr('transform', `translate(${margin}, ${margin})`);

```

/\*

It converts a continuous input domain into a continuous output range.

Notice the range and domain method. The first one takes the length that should be divided between the limits of the domain values.

```

*/

// the SVG coordinate system starts from the top left corner that's why
// the range takes the height as the first parameter and not zero.
const yScale = d3.scaleLinear().range([height, 0]).domain([0, 6000]);

// axis on the left is as simple as adding another group and calling d3's
// axisLeft method with the scaling function as a parameter
chart.append('g').call(d3.axisLeft(yScale));

const xScale = d3.scaleBand()
  .range([0, width])
  .domain(data.map((d) => d.saleType))
  .padding(0.2);

chart.append('g')
  .attr('transform', `translate(0, ${height})`)
  .call(d3.axisBottom(xScale));

const makeYLines = () => d3.axisLeft().scale(yScale)
chart.append('g').call(d3.axisLeft(yScale));

// vertical grid lines
// chart.append('g')
//   .attr('class', 'grid')
//   .attr('transform', `translate(0, ${height})`)
//   .call(makeXLines())
//   .tickSize(-height, 0, 0)

```

```
// .tickFormat("")
```

```
// )
```

```
chart.append('g')
```

```
  .attr('class', 'grid')
```

```
  .call(makeYLines())
```

```
    .tickSize(-width, 0, 0)
```

```
    .tickFormat("")
```

```
  );
```

```
const barGroups = chart.selectAll()
```

```
  .data(data)
```

```
  .enter()
```

```
  .append('g');
```

```
barGroups
```

```
  .append('rect')
```

```
  .attr('class', 'bar')
```

```
  .attr('x', (g) => xScale(g.saleType))
```

```
  .attr('y', (g) => yScale(g.saleAmount))
```

```
  .attr('height', (g) => height - yScale(g.saleAmount))
```

```
  .attr('width', xScale.bandwidth())
```

```
  .on('mouseenter', function (actual, i) {
```

```
    console.log(actual.saleAmount);
```

```
    d3.selectAll('.value')
```

```
      .attr('opacity', 0)
```

```
    d3.select(this)
```

```
      .transition()
```

```
.duration(300)

.attr('opacity', 0.6)

.attr('x', (a) => xScale(a.saleType) - 5)

.attr('width', xScale.bandwidth() + 10)
```

```
const y = yScale(actual.saleAmount)
```

```
line = chart.append('line')

.attr('id', 'limit')

.attr('x1', 0)

.attr('y1', y)

.attr('x2', width)

.attr('y2', y)
```

```
barGroups.append('text')

.attr('class', 'divergence')

.attr('x', (a) => xScale(a.saleType) + xScale.bandwidth() / 2)

.attr('y', (a) => yScale(a.saleAmount) + 30)

.attr('fill', 'white')

.attr('text-anchor', 'middle')

.text((a, idx) => {

  console.log(a.saleAmount);

  console.log(actual.saleAmount);

  const divergence = (a.saleAmount - actual.saleAmount).toFixed(1)

  let text = "

  if (divergence > 0) text += '+'
```

```

        text += `${divergence}`

        return idx !== i ? text : "";
    })
})

.on('mouseleave', function () {
    d3.selectAll('.value')
        .attr('opacity', 1)
    d3.select(this)
        .transition()
        .duration(300)
        .attr('opacity', 1)
        .attr('x', (a) => xScale(a.saleType))
        .attr('width', xScale.bandwidth())

    chart.selectAll('#limit').remove()
    chart.selectAll('.divergence').remove()
})

/*chart.selectAll()
    .data(data.length)
    .enter()
    .append('rect')
    .attr('x', (s) => xScale(s.saleType))
    .attr('y', (s) => yScale(s.saleAmount))
    .attr('height', (s) => height - yScale(s.saleAmount))
    .attr('width', xScale.bandwidth())
    .attr('x', (actual, index, data) => xScale(actual.saleAmount));    */

```

barGroups

```
.append('text')  
  
.attr('class', 'value')  
  
.attr('x', (a) => xScale(a.saleType) + xScale.bandwidth() / 2)  
  
.attr('y', (a) => yScale(a.saleAmount) + 30)  
  
.attr('text-anchor', 'middle')  
  
.text((a) => `${a.saleAmount}`)
```

svg

```
.append('text')  
  
.attr('class', 'label')  
  
.attr('x', -(height / 2) - margin)  
  
.attr('y', margin / 2.4)  
  
.attr('transform', 'rotate(-90)')  
  
.attr('text-anchor', 'middle')  
  
.text('sales Unit')
```

svg.append('text')

```
.attr('class', 'label')  
  
.attr('x', width / 2 + margin)  
  
.attr('y', height + margin * 1.7)  
  
.attr('text-anchor', 'middle')  
  
.text('Products sales')
```

svg.append('text')

```
.attr('class', 'title')
```

```

.attr('x', width / 2 + margin)

.attr('y', 40)

.attr('text-anchor', 'middle')

.text('Sales Data of February Month')


svg.append('text')

.attr('class', 'source')

.attr('x', width - margin / 2)

.attr('y', height + margin * 1.7)

.attr('text-anchor', 'start')

.text('Source: PDS sales.csv, 2020')

});

});

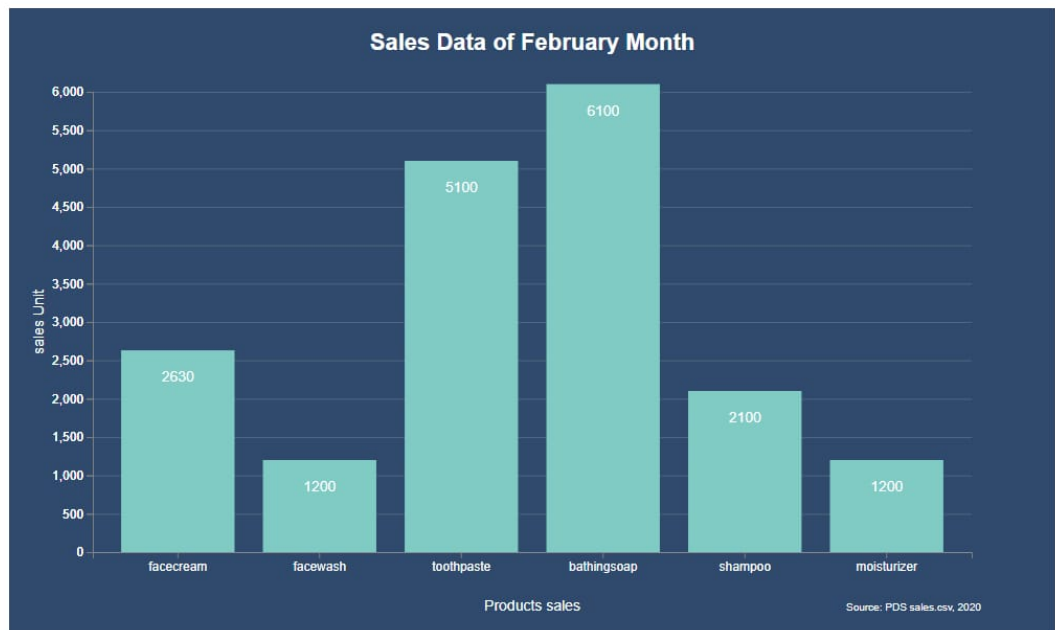
</script>

</html>

```

### **Output:**

**Showing the Data as a column chart using D3.js**





#### 4(B) : Showing the data as a stacked column chart Using HTML5 and D3.js and Canvas.js

```
<!DOCTYPE HTML>

<html>

<head>

  <title>Showing the Data as a column chart for four age group</title>

  <meta charset="utf-8">

  <meta name="author" content="SidPro" />

  <meta name="viewport" content="width=device-width,initial-scale=1.0" />

  <meta name="description" content="Data visualization using javaScript and HTML" />

  <script src="js/jquery-3.5.1.min.js"></script>

  <script src="js/canvasjs.min.js"></script>

</head>

<body>

  <h2>Showing the Data as a column chart for died and survived person</h2>

  <div id="chartContainer" style="height: 600px; width: 100%;"></div>

</body>

<script type="text/javascript">

  // Millennials = ages of 18 to 34

  // Gen X = ages of 35 to 50

  // Baby Boomers = ages of 51 to 69

  // Silent generation = ages of 70 to 87

  var age1 = 0, age2 = 0, age3 = 0, age4 = 0, j = -1;

  var age1died = 0, age2died = 0, age3died = 0, age4died = 0;

  window.onload = function () {
```

```
//just change the name of .csv file to load file
```

```
$.get('data/titanic.csv', function (theData) {
```

```
    theData = theData.replace(/"/g, "");
```

```
    theData = theData.split(/\r?\n\r/);
```

```
    totalRows = theData.length;
```

```
    for (let i = 1; i < totalRows; ++i) {
```

```
        theTD = theData[i].split(',');
```

```
        value = parseInt(theTD[4]);
```

```
        died = (theTD[2] === "died") ? true : false;
```

```
        if (value >= 18 && value <= 34) {
```

```
            if (died) age1died += 1;
```

```
            else age1 += 1;
```

```
        }
```

```
        else if (value >= 35 && value <= 50) {
```

```
            if (died) age2died += 1;
```

```
            else age2 += 1;
```

```
        }
```

```
        else if (value >= 51 && value <= 69) {
```

```
            if (died) age3died += 1;
```

```
            else age3 += 1;
```

```
        }
```

```
        else if (value >= 70 && value <= 87) {
```

```

        if (died) age4died += 1;

        else age4 += 1;

    }

}

var chart = new CanvasJS.Chart("chartContainer", {

    title: {

        text: "Stacked Column chart for Four Age Group in Titanic"

    },

    axisX: {

        title: "Four Age group",

    },

    axisY: {

        interval: 50

    },

    legend: {

        fontSize: 20

    },

    theme: "dark2", // "light1", "light2", "dark1", "dark2"

    data: [

        {

            // Change type to "stackedArea", "stackedColumn", "column", "doughnut",
            "line", "splineArea" etc.

            type: "stackedColumn",

            showInLegend: true,

            legendText: "died",

            dataPoints: [

                { label: "Millennials died", y: age1died },

```

```

        { label: "Gen X died", y: age2died },
        { label: "Baby Boomers died", y: age3died },
        { label: "Silent generation died", y: age4died }
    ]

    }, {
        // Change type to "stackedArea", "stackedColumn", "column", "doughnut",
        "line", "splineArea" etc.

        type: "stackedColumn",
        showInLegend: true,
        legendText: "survived",
        dataPoints: [
            { label: "Millennials", y: age1 },
            { label: "Gen X", y: age2 },
            { label: "Baby Boomers", y: age3 },
            { label: "Silent generation", y: age4 }
        ]

    }

]

});

chart.render();

});

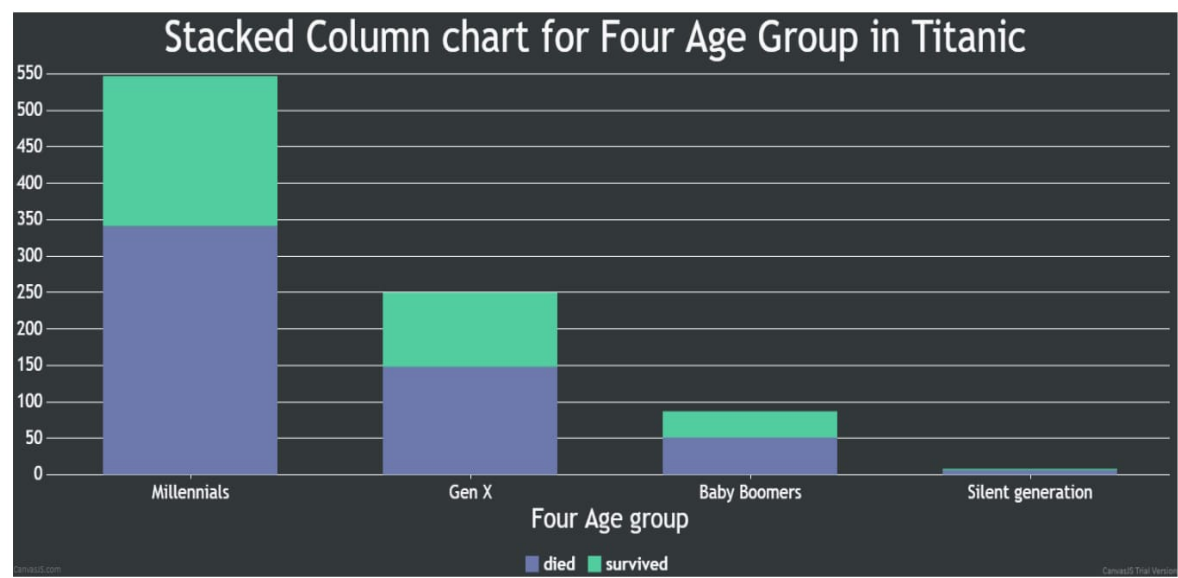
}

</script>
</html>

```

**Output:**

Showing the Data as a column chart for died and survived person



#### 4(C) : Showing the Data as a column chart for four age group m Using HTML5 and D3.js and Canvas.js

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<head>
```

```
  <title>Showing the Data as a column chart for four age group</title>
```

```
  <meta charset="utf-8">
```

```
  <meta name="author" content="SidPro" />
```

```
  <meta name="viewport" content="width=device-width,initial-scale=1.0" />
```

```
  <meta name="description" content="Data visualization using javaScript and HTML" />
```

```
  <script src="js/jquery-3.5.1.min.js"></script>
```

```
  <script src="js/canvasjs.min.js"></script>
```

```
</head>
```

```
<body>
```

```
  <h2>Showing the Data as a column chart for four age group</h2>
```

```
  <div id="chartContainer" style="height: 600px; width: 100%;"></div>
```

```
</body>
```

```
<script type="text/javascript">
```

```
  // Millennials = ages of 18 to 34
```

```
  // Gen X = ages of 35 to 50
```

```
  // Baby Boomers = ages of 51 to 69
```

```
  // Silent generation = ages of 70 to 87
```

```

var age1 = 0, age2 = 0, age3 = 0, age4 = 0;

window.onload = function () {

    //just change the name of .csv file to load file

    $.get('data/titanic.csv', function (theData) {

        theData = theData.replace(/"/g, "");

        theData = theData.split(/\r?\n\r/);

        totalRows = theData.length;

        for (let i = 1; i < totalRows; ++i) {

            theTD = theData[i].split(',');

            value = parseInt(theTD[4]);

            if (value >= 18 && value <= 34) age1 += 1;

            else if (value >= 35 && value <= 50) age2 += 1;

            else if (value >= 51 && value <= 69) age3 += 1;

            else if (value >= 70 && value <= 87) age4 += 1;

        }

        var chart = new CanvasJS.Chart("chartContainer", {

            title: {

                text: "Column chart for Four Age Group in Titanic"

            },

            theme: "dark1", //"light1", "light2", "dark1", "dark2"

            data: [

                {

                    // Change type to "column", "doughnut", "line", "splineArea", etc.

                    type: "column",

                    dataPoints: [

                        { label: "Millennials", y: age1 },

                        { label: "Gen X", y: age2 },

```

```

        { label: "Baby Boomers", y: age3 },
        { label: "Silent generation", y: age4 }
    ]
}

]

});

chart.render();

});

}

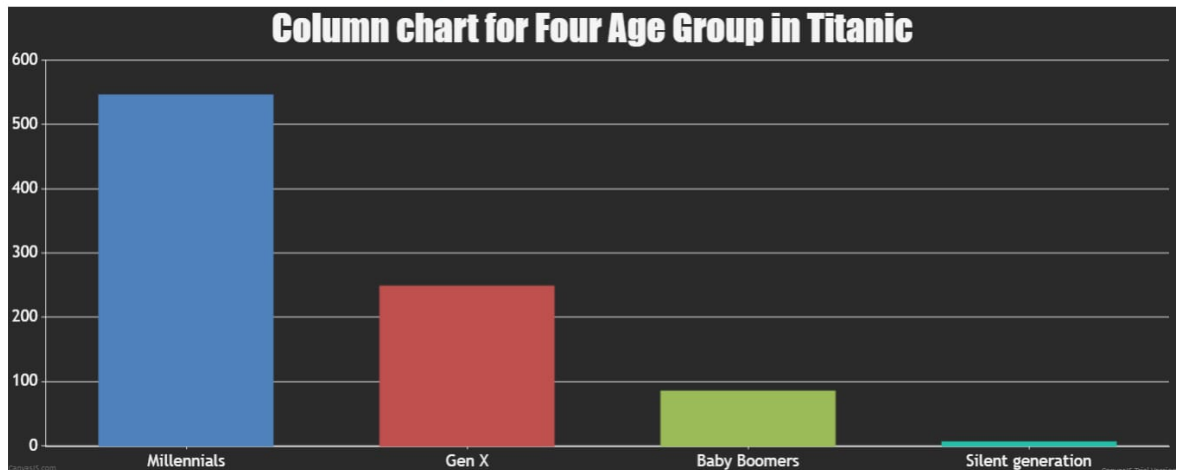
</script>

</html>

```

### **Output:**

Showing the Data as a column chart for four age group





#### 4(D) : Showing the data as a Line chart (single, fewer and multiple lines) Using HTML5 and D3.js and Canvas.js

```
<!DOCTYPE HTML>

<html>

<head>

  <title>Single Line chart</title>

  <meta charset="utf-8">

  <meta name="author" content="SidPro" />

  <meta name="viewport" content="width=device-width,initial-scale=1.0" />

  <meta name="description" content="Showing the data as a Single Line chart" />

  <meta name="keywords" content="Single,Line,Chart,Single Line Chart,Line" />

  <script src="js/jquery-3.5.1.min.js"></script>

  <script src="js/canvasjs.min.js"></script>

</head>

<body>

  <h2>Showing the data as a Single Line chart</h2>

  <div id="chartContainer" style="height: 600px; width: 100%;"></div>

</body>

<script type="text/javascript">

  window.onload = function () {

    /*CanvasJS.addColorSet("colors",

      [//colorSet Array

        "#2F4F4F",

        "#008080",

        "#2E8B57",

        "#3CB371",

        "#90EE90"

      ]);*/
```

```

var chart = new CanvasJS.Chart("chartContainer", {
    //colorSet: "colors",

    title: {
        text: "Showing the sales data of Facewash as a Single Line chart",
        fontSize: 30,
    },

    axisX: {
        title: "Month",
    },

    axisY: {
        title: "Sales Unit",
    },

    theme: "dark1", //"light1", "light2", "dark1", "dark2"

    data: [
        {
            // Change type to "column", "doughnut", "line", "splineArea", etc.
            type: "line",
            showInLegend: true,
            legendText: "Facewash",
            dataPoints: [
                { label: "Jan", y: 1200 },
                { label: "Feb", y: 2100 },
                { label: "Mar", y: 3550 },
                { label: "Apr", y: 1870 },
                { label: "May", y: 1560 },
                { label: "Jun", y: 1890 },
                { label: "Jul", y: 1780 },
                { label: "Aug", y: 2860 },
                { label: "Sep", y: 2100 },
            ]
        }
    ]
});

```

```

        { label: "Oct", y: 2300 },
        { label: "Nov", y: 2400 },
        { label: "Dec", y: 1800 }
    ]
}

]

});

chart.render();

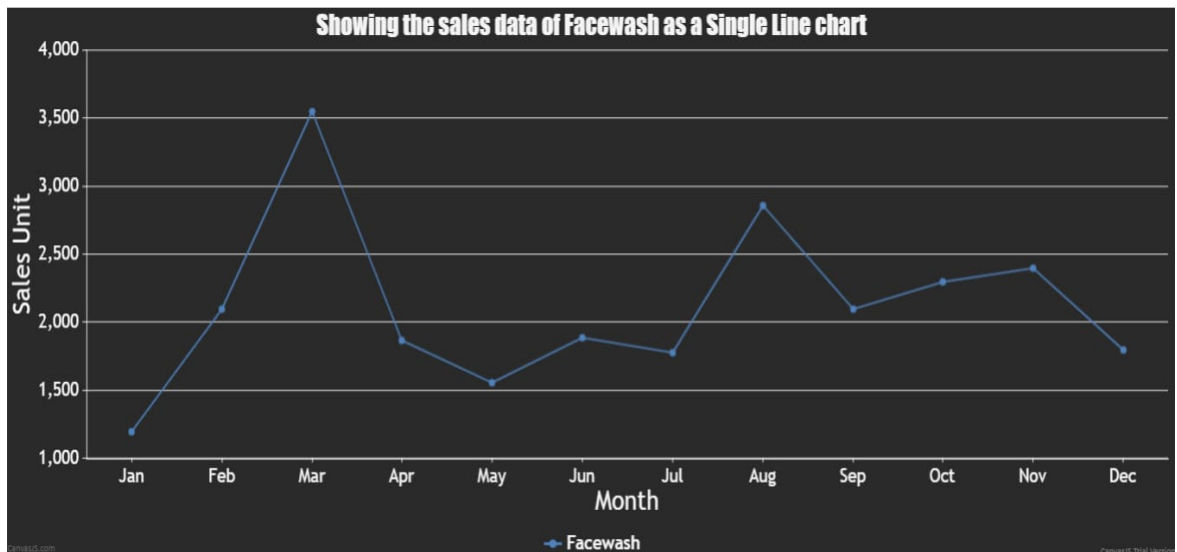
}

</script>
</html>

```

### **Output:**

Showing the data as a Single Line chart



#### 4(E) : Showing the data as a Pie Chart (single and multiple pie) m Using HTML5 and D3.js and Canvas.js

```
<!DOCTYPE HTML>

<html>

<head>

  <title>Single Pie chart</title>

  <meta charset="utf-8">

  <meta name="author" content="SidPro" />

  <meta name="viewport" content="width=device-width,initial-scale=1.0" />

  <meta name="description" content="Showing the data as a Single Pie chart" />

  <meta name="keywords" content="Single,Pie,Chart,Single Pie Chart,Line" />

  <script src="js/jquery-3.5.1.min.js"></script>

  <script src="js/canvasjs.min.js"></script>

</head>

<body>

  <h2>Showing the data as a Single Pie chart</h2>

  <div id="chartContainer" style="height: 600px; width: 100%;"></div>

</body>

<script type="text/javascript">

  var facecream = 0, facewash = 0, toothpaste = 0

    , bathingsoap = 0, shampoo = 0, moisturizer = 0;

  window.onload = function () {

    $.get('data/data.txt', function (theData) {

      theData = theData.replace(/\r/g, "");

      theData = theData.replace(/\t/g, ' ');
```

```

theData = theData.split('\n');
totalRows = theData.length;
data = theData;
for (let i = 1; i < totalRows; ++i) {
    theData = data[i].split(' ');
    facecream += parseInt(theData[1]);
    facewash += parseInt(theData[2]);
    toothpaste += parseInt(theData[3]);
    bathingsoap += parseInt(theData[4]);
    shampoo += parseInt(theData[5]);
    moisturizer += parseInt(theData[6]);
}

//console.log(facecream,facewash,toothpaste,bathingsoap,shampoo,moisturizer);

total = facecream + facewash + toothpaste + bathingsoap + shampoo + moisturizer;

//console.log(total);

facecream = facecream * 100 / total;
facewash = facewash * 100 / total;
toothpaste = toothpaste * 100 / total;
bathingsoap = bathingsoap * 100 / total;
shampoo = shampoo * 100 / total;
moisturizer = moisturizer * 100 / total;

//console.log(facecream,facewash,toothpaste,bathingsoap,shampoo,moisturizer);

CanvasJS.addColorSet("colors",

    [//colorSet Array

        "#2F4F4F",

        "#008080",

        "#2E8B57",

```

```
        "#3CB371",
        "#90EE90",
        "#5d9e9e"
    ]);

var chart = new CanvasJS.Chart("chartContainer", {
    colorSet: "colors",
    theme: "light2",
    exportEnabled: false,
    animationEnabled: true,
    title: {
        text: "Yearly sales of Products"
    },
    legend: {
        cursor: "pointer"
    },
    subtitles: [{
        text: "Single Pie chart",
        fontSize: 16
    }],
    data: [
        {
            type: "pie",
            showInLegend: true,
            indexLabelFontSize: 18,
            radius: 180,
            indexLabel: "{name} - {y}",
            yValueFormatString: "###0.0\\'%\\'",
```

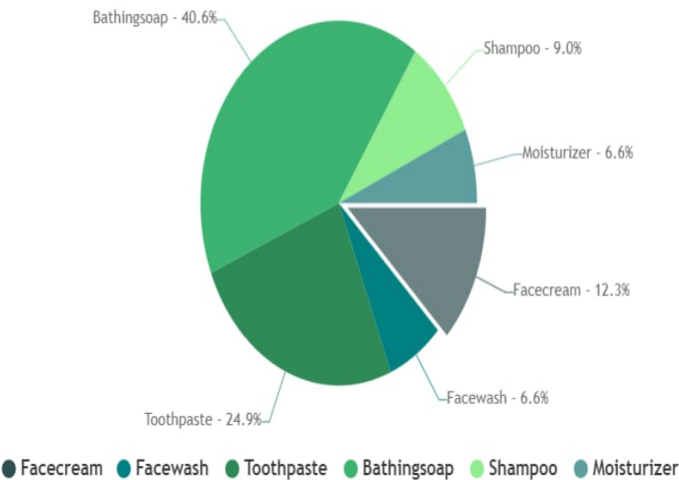
```
dataPoints: [  
    { y: facecream, name: "Facecream", exploded: true },  
    { y: facewash, name: "Facewash" },  
    { y: toothpaste, name: "Toothpaste" },  
    { y: bathingsoap, name: "Bathingsoap" },  
    { y: shampoo, name: "Shampoo" },  
    { y: moisturizer, name: "Moisturizer" }  
]  
}  
]  
});  
chart.render();  
});  
  
}  
</script>  
</html>
```

**Output:**

Showing the data as a Single Pie chart

# Yearly sales of Products

Single Pie chart





#### 4(F) : Showing the data as a Bar Chart (Simple and multiple) Using HTML5 and D3.js and Canvas.js

```
<!DOCTYPE HTML>

<html>

<head>

  <title>Showing the data as a Simple Bar Chart</title>

  <meta charset="utf-8">

  <meta name="author" content="SidPro" />

  <meta name="viewport" content="width=device-width,initial-scale=1.0" />

  <meta name="description" content="Showing the sales data of Facecream as a Simple Bar
chart" />

  <meta name="keywords" content="Single,Bar,Chart,Single Bar Chart,Line" />

  <script src="js/jquery-3.5.1.min.js"></script>

  <script src="js/canvasjs.min.js"></script>

</head>

<body>

  <h2>Showing the data as a Simple Bar Chart</h2>

  <div id="chartContainer" style="height: 600px; width: 100%;"></div>

</body>

<script type="text/javascript">

  window.onload = function () {

    var chart = new CanvasJS.Chart("chartContainer", {

      title: {

        text: "the sales data of Facecream"

      },

      axisX: {
```

```

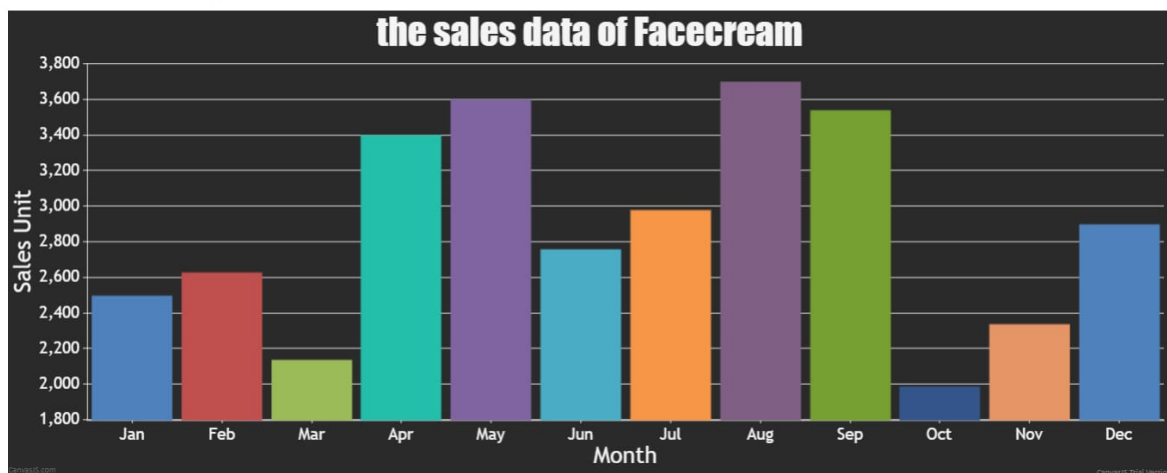
        title: "Month",
    },
    axisY: {
        title: "Sales Unit",
    },
    theme: "dark1", //"light1", "light2", "dark1", "dark2"
    data: [
        {
            // Change type to "column", "doughnut", "line", "splineArea", etc.
            type: "column",
            dataPoints: [
                { label: "Jan", y: 2500 },
                { label: "Feb", y: 2630 },
                { label: "Mar", y: 2140 },
                { label: "Apr", y: 3400 },
                { label: "May", y: 3600 },
                { label: "Jun", y: 2760 },
                { label: "Jul", y: 2980 },
                { label: "Aug", y: 3700 },
                { label: "Sep", y: 3540 },
                { label: "Oct", y: 1990 },
                { label: "Nov", y: 2340 },
                { label: "Dec", y: 2900 }
            ]
        }
    ]
});

```

```
        chart.render();  
    }  
</script>  
</html>
```

### **Output:**

Showing the data as a Simple Bar Chart



## PRACTICAL:5

### 5(A) : Using Google Charts API Basics draw charts like a Bar chart Using HTML5 and Google Chats API and Map API

```
<!DOCTYPE HTML>

<html>

<head>

  <title>Google API Bar chart</title>

  <meta charset="utf-8">

  <meta name="author" content="SidPro" />

  <meta name="viewport" content="width=device-width,initial-scale=1.0" />

  <meta name="description" content="Using Google Charts API Basics draw charts like a
Bar chart" />

  <meta name="keywords" content="Google,API,Chart,Bar Chart,Bar" />

  <script src="js/jquery-3.5.1.min.js"></script>

  <script src="js/loader.js"></script>

</head>

<body>

  <h2>Using Google Charts API Basics draw charts like a Bar chart</h2>

  <div id="chart" style="width: 900px; height: 500px"></div>

</body>

<script type="text/javascript">

  google.charts.load('current', { 'packages': ['bar'] });

  google.charts.setOnLoadCallback(drawChart);

  function drawChart() {

    var data = google.visualization.arrayToDataTable([

      ['Month', 'Toothpaste', 'Bathingsoap', 'Shampoo'],
```

```
['Jan', 5200, 9200, 1200],  
['Feb', 5100, 6100, 2100],  
['Mar', 4550, 9550, 3550],  
['Apr', 5870, 8870, 1870],  
['May', 4560, 7760, 1560],  
['Jun', 4890, 7490, 1890],  
['Jul', 4780, 8980, 1780],  
['Aug', 5860, 9960, 2860],  
['Sep', 6100, 8100, 2100],  
['Oct', 8300, 10300, 2300],  
['Nov', 7300, 13300, 2400],  
['Dec', 7400, 14400, 1800],  
]);
```

```
var options = {  
  chart: {  
    title: 'Company Sales',  
    subtitle: 'Toothpaste ,Bathingsoap and Shampoo Monthly',  
  },  
  chartArea: {  
    backgroundColor: '#2a2a2a'  
  },  
  backgroundColor: '#2a2a2a',  
  colors: ['#9bbb58', '#4f81bc', '#c0504e'],  
  titleTextStyle: { color: 'white', bold: true, fontSize: 20, italic: false },  
  hAxis: {  
    title: 'Months',
```

```

        titleTextStyle: { color: 'white', bold: false, italic: false },
        textStyle: { color: 'white', bold: false, italic: false },
        baselineColor: '#51cda0',
        gridlines: { color: 'green', minSpacing: 20 }
    },
    vAxis: {
        title: 'Sales Unit',
        titleTextStyle: { color: 'white', bold: false, italic: false },
        baselineColor: 'white',
        gridlines: { color: 'gray' },
        textStyle: { color: 'white', bold: false, italic: false }
    },
    animation: { duration: 1000, startup: true, easing: 'linear' },
    tooltip: { textStyle: { color: '#871b47' }, showColorCode: true },
    legend: { position: 'bottom', textStyle: { color: 'white', fontSize: 16 } }
};

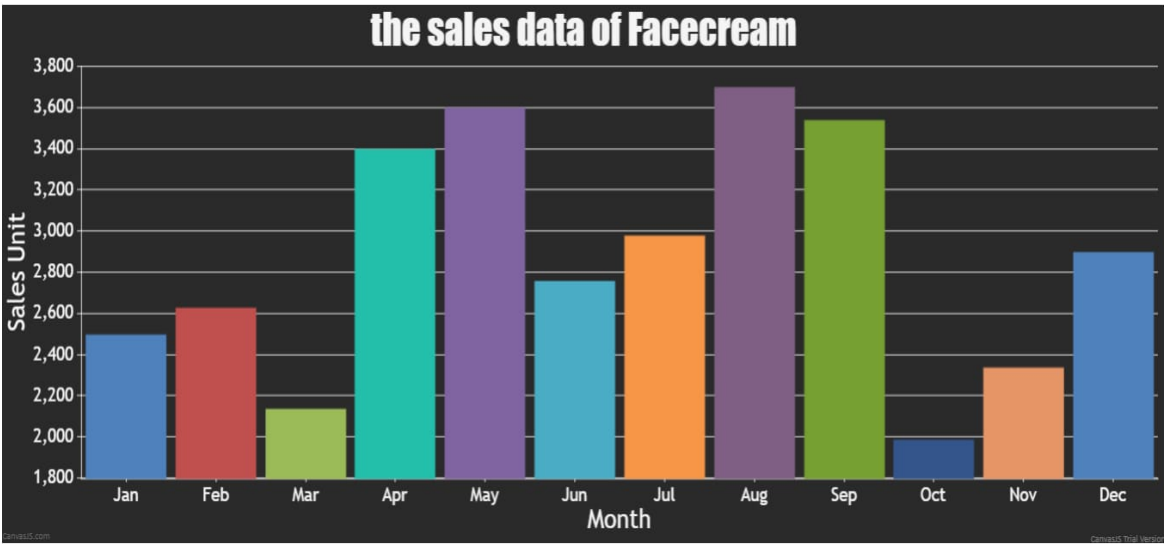
var chart = new google.charts.Bar(document.getElementById('chart'));

chart.draw(data, google.charts.Bar.convertOptions(options));
}
</script>
</html>

```

**Output:**

Showing the data as a Simple Bar Chart



## 5(B) : Using Google Charts API Basics draw charts like a Line chart

```
<!DOCTYPE HTML>

<html>

<head>

  <title>Google API Line chart</title>

  <meta charset="utf-8">

  <meta name="author" content="SidPro" />

  <meta name="viewport" content="width=device-width,initial-scale=1.0" />

  <meta name="description" content="Using Google Charts API Basics draw charts like a
Line chart" />

  <meta name="keywords" content="Google,API,Chart,Line Chart,Line" />

  <script src="js/jquery-3.5.1.min.js"></script>

  <script src="js/loader.js"></script>

</head>

<body>

  <h2>Using Google Charts API Basics draw charts like a Line chart</h2>

  <div id="chart" style="width: 900px; height: 500px"></div>

</body>

<script type="text/javascript">

  google.charts.load('current', { 'packages': ['corechart'] });

  google.charts.setOnLoadCallback(drawChart);

  function drawChart() {

    var data = google.visualization.arrayToDataTable([

      ['Month', 'Shampoo', 'Moisturizer'],

      ['Jan', 1500, 2110],

      ['Feb', 1200, 1833],

      ['Mar', 1340, 2247],

      ['Apr', 1130, 2227],

      ['May', 1740, 2096],

      ['Jun', 1555, 2014],
```



```
['Jul', 1120, 2955],  
['Aug', 1400, 3614],  
['Sep', 1780, 2340],  
['Oct', 1890, 2667],  
['Nov', 2100, 4128],  
['Dec', 1760, 3002],  
]);
```

```
var options = {  
  title: 'Monthly Sales Data of Shampoo and Moisturizer',  
  curveType: 'none',  
  backgroundColor: '#2a2a2a',  
  colors: ['#51cda0', '#6d78ad'],  
  titleTextStyle: { color: 'white', bold: true, fontSize: 20, italic: false },  
  hAxis: {  
    title: 'Months',  
    titleTextStyle: { color: 'white', bold: false, italic: false },  
    textStyle: { color: 'white', bold: false, italic: false },  
    baselineColor: '#51cda0',  
    gridlines: { color: 'green', minSpacing: 20 }  
  },  
  vAxis: {  
    title: 'Sales Unit',  
    titleTextStyle: { color: 'white', bold: false, italic: false },  
    baselineColor: 'white',  
    gridlines: { color: 'gray' },  
    textStyle: { color: 'white', bold: false, italic: false }  
  },  
  animation: { duration: 1000, startup: true, easing: 'linear' },
```

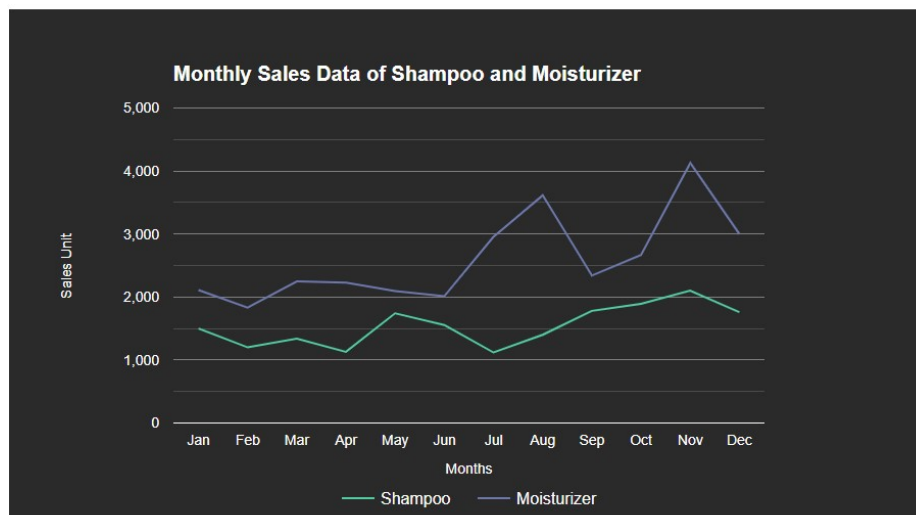
```

        tooltip: { textStyle: { color: '#871b47' }, showColorCode: true },
        legend: { position: 'bottom', textStyle: { color: 'white', fontSize: 16 } }
    };

    var chart = new google.visualization.LineChart(document.getElementById('chart'));
    chart.draw(data, options);
}
</script>
</html>

```

### **Output:**



## 5(C) : Using Google Charts API Basics draw PieChart.

```
<!DOCTYPE HTML>

<html>

<head>

  <title>Google API Pie chart</title>

  <meta charset="utf-8">

  <meta name="author" content="SidPro" />

  <meta name="viewport" content="width=device-width,initial-scale=1.0" />

  <meta name="description" content="Using Google Charts API Basics draw charts like a Pie chart" />

  <meta name="keywords" content="Google,API,Chart,Pie Chart,Pie" />

  <script src="js/jquery-3.5.1.min.js"></script>

  <script src="js/loader.js"></script>

</head>

<body>

  <h2>Using Google Charts API Basics draw charts like a Pie chart</h2>

  <div id="chart" style="width: 900px; height: 500px"></div>

</body>

<script type="text/javascript">

  google.charts.load("current", { packages: ["corechart"] });

  google.charts.setOnLoadCallback(drawChart);

  function drawChart() {

    var data = google.visualization.arrayToDataTable([

      ['Products', 'Sales'],

      ['Bathingsoap', 114010],

      ['Shampoo', 25410],

      ['Moisturizer', 18515],

      ['Facecream', 34480],

      ['Facewash', 18515],

      ['Toothpaste', 69910]

    ]);

    var options = {
```

```

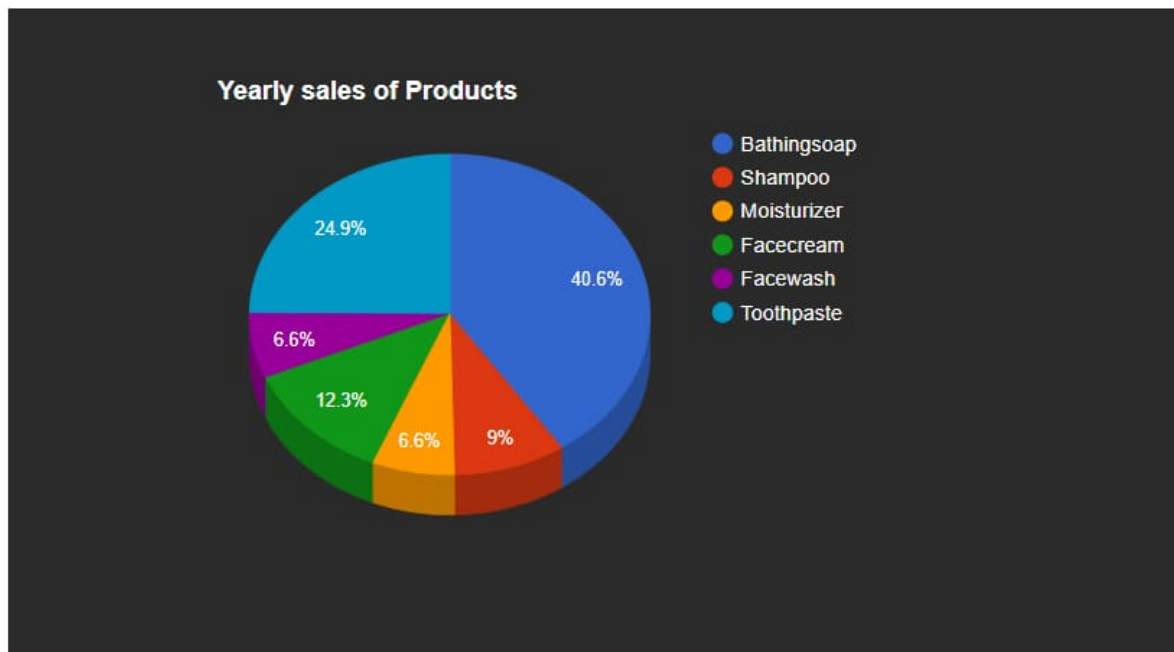
    title: 'Yearly sales of Products',
    titleTextStyle: { color: 'white', bold: true, fontSize: 20, italic: false },
    is3D: true,
    tooltip: { textStyle: { color: '#871b47' }, showColorCode: true },
    backgroundColor: '#2a2a2a',
    legend: { position: 'right', textStyle: { color: 'white', fontSize: 16 } }
  };

  var chart = new google.visualization.PieChart(document.getElementById('chart'));
  chart.draw(data, options);
}
</script>
</html>

```

### **Output:**

**Using Google Charts API Basics draw charts like a Pie chart**



## 5(D) : Using Google Charts API Basics draw Donut Chart.

```
<!DOCTYPE HTML>

<html>

<head>

  <title>Google API Donut chart</title>

  <meta charset="utf-8">

  <meta name="author" content="SidPro" />

  <meta name="viewport" content="width=device-width,initial-scale=1.0" />

  <meta name="description" content="Using Google Charts API Basics draw charts like a Donut chart" />

  <meta name="keywords" content="Google,API,Chart,Donut Chart,Donut" />

  <script src="js/jquery-3.5.1.min.js"></script>

  <script src="js/loader.js"></script>

</head>

<body>

  <h2>Using Google Charts API Basics draw charts like a Donut chart</h2>

  <div id="chart" style="width: 900px; height: 500px"></div>

</body>

<script type="text/javascript">

  google.charts.load("current", { packages: ["corechart"] });

  google.charts.setOnLoadCallback(drawChart);

  function drawChart() {

    var data = google.visualization.arrayToDataTable([

      ['Products', 'Sales'],

      ['Bathingsoap', 114010],

      ['Shampoo', 25410],

      ['Moisturizer', 18515],

      ['Facecream', 34480],

      ['Facewash', 18515],

      ['Toothpaste', 69910]

    ]);

    var options = {
```

```

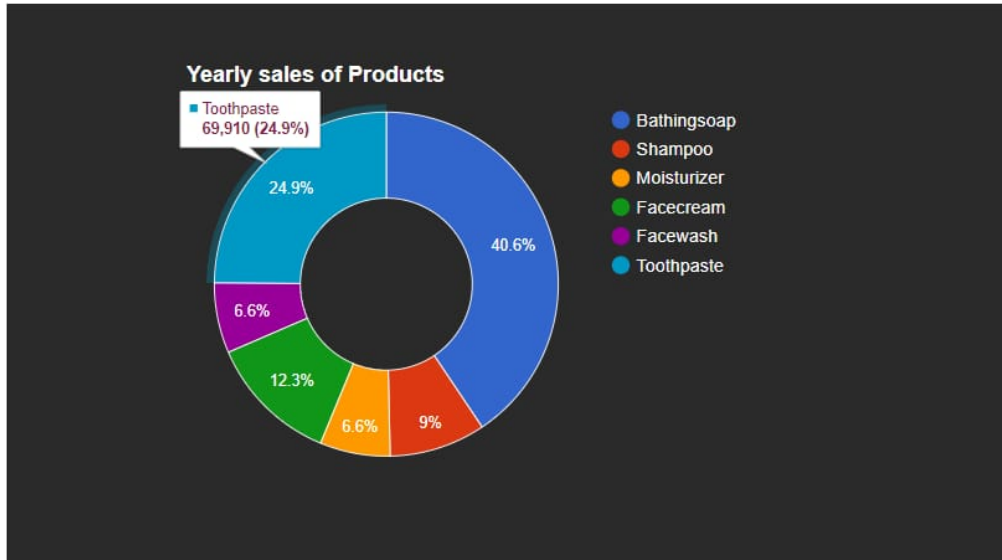
    title: 'Yearly sales of Products',
    titleTextStyle: { color: 'white', bold: true, fontSize: 20, italic: false },
    //colors:['#102b2c','#503491','#3c3f17','#551e3a','#2b240f','#724628'],
    pieHole: 0.5,
    tooltip: { textStyle: { color: '#871b47' }, showColorCode: true },
    backgroundColor: '#2a2a2a',
    legend: { position: 'right', textStyle: { color: 'white', fontSize: 16 } }
  };

  var chart = new google.visualization.PieChart(document.getElementById('chart'));
  chart.draw(data, options);
}
</script>
</html>

```

### **Output:**

#### **Using Google Charts API Basics draw charts like a Donut chart**



## 5(E) : Using Google Charts API Basics draw Candle Chart.

```
<!DOCTYPE HTML>

<html>

<head>

  <title>Google API CandleStick chart</title>

  <meta charset="utf-8">

  <meta name="author" content="SidPro" />

  <meta name="viewport" content="width=device-width,initial-scale=1.0" />

  <meta name="description" content="Using Google Charts API Basics draw charts like a CandleStick chart"
/>

  <meta name="keywords" content="Google,API,Chart,CandleStick Chart,CandleStick" />

  <script src="js/jquery-3.5.1.min.js"></script>

  <script src="js/loader.js"></script>

</head>

<body>

  <h2>Using Google Charts API Basics draw charts like a CandleStick chart</h2>

  <div id="chart" style="width: 900px; height: 500px"></div>

</body>

<script type="text/javascript">

  google.charts.load("current", { packages: ["corechart"] });

  google.charts.setOnLoadCallback(drawChart);

  function drawChart() {

    var data = google.visualization.arrayToDataTable([

      //(NSE Indices - Nifty 500)

      //data format low,open,close,high if open<close then candle will be filled;else candle will be hollow

      ['Just Dial', 784, 820, 870, 885],

      ['Tata Chemical', 590, 614, 732, 743],

      ['Canara Bank', 680, 840, 706, 860],

      ['Torrent Power', 560, 600, 710, 743],
```

```

['IRCTC', 520, 560, 712, 743],

['Airtel', 630, 720, 670, 743],

// https://www.investopedia.com/trading/candlestick-charting-what-is-it/
// Treat first row as data as well.

], true);

var options = {
  title: '(NSE Indices - Nifty 500)',
  titleTextStyle: { color: 'white', bold: true, fontSize: 20, italic: false },
  colors: ['#6d78ad'],
  backgroundColor: '#2a2a2a',
  animation: { duration: 1000, easing: 'linear', startup: true },
  tooltip: { textStyle: { color: '#871b47' }, showColorCode: true },
  hAxis: {
    title: 'Company Name',
    titleTextStyle: { color: 'white', bold: false, italic: false },
    textStyle: { color: 'white', bold: false, italic: false },
    baselineColor: '#51cda0',
    gridlines: { color: 'green', minSpacing: 20 }
  },
  vAxis: {
    title: 'Stock stat',
    titleTextStyle: { color: 'white', bold: false, italic: false },
    baselineColor: 'white',
    gridlines: { color: 'gray' },
    textStyle: { color: 'white', bold: false, italic: false }
  },
  candlestick: {
    hollowIsRising: true,
    fallingColor: { fill: 'white', stroke: '#6d78ad', strokeWidth: 2 },

```



```

        risingColor: { fill: '#6d78ad', stroke: '#6d78ad', strokeWidth: 2 }
    },
    legend: 'none'
};

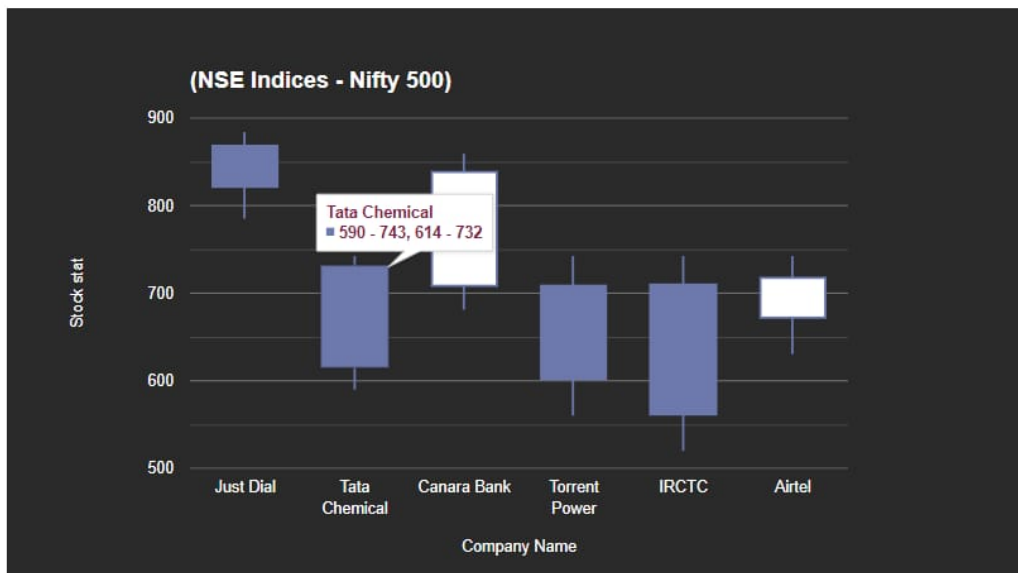
var chart = new google.visualization.CandlestickChart(document.getElementById('chart'));

chart.draw(data, options);
}
</script>
</html>

```

### **Output:**

#### **Using Google Charts API Basics draw charts like a CandleStick chart**



## 5(F) : Using Google Charts API Basics draw other types of Chart.

```
<!DOCTYPE HTML>

<html>

<head>

  <title>Google API Bubble chart</title>

  <meta charset="utf-8">

  <meta name="author" content="SidPro" />

  <meta name="viewport" content="width=device-width,initial-scale=1.0" />

  <meta name="description" content="Using Google Charts API Basics draw chart(other type)
bubble chart" />

  <meta name="keywords" content="Google,API,Chart,Bubble Chart,Bubble" />

  <script src="js/jquery-3.5.1.min.js"></script>

  <script src="js/loader.js"></script>

</head>

<body>

  <h2>Using Google Charts API Basics draw chart(other type) bubble chart</h2>

  <div id="chart" style="width: 900px; height: 500px"></div>

</body>

<script type="text/javascript">

  google.charts.load('current', { 'packages': ['corechart'] });

  google.charts.setOnLoadCallback(drawSeriesChart);

  function drawSeriesChart() {

    var data = google.visualization.arrayToDataTable([

      ['ID', 'Life Expectancy', 'Fertility Rate', 'Region', 'Population'],

      ['CAN', 80.66, 1.67, 'North America', 33739900],

      ['DEU', 79.84, 1.36, 'Europe', 81902307],

      ['DNK', 78.6, 1.84, 'Europe', 5523095],

      ['EGY', 72.73, 2.78, 'Middle East', 79716203],

      ['GBR', 80.05, 2, 'Europe', 61801570],

      ['IRN', 72.49, 1.7, 'Middle East', 73137148],

      ['IRQ', 68.09, 4.77, 'Middle East', 31090763],
```

```

[ISR', 81.55, 2.96, 'Middle East', 7485600],
[RUS', 68.6, 1.54, 'Europe', 141850000],
[USA', 78.09, 2.05, 'North America', 307007000]
]);

var options = {
  title: 'Correlation between life expectancy, fertility rate ' +
    'and population of some world countries (2010)',
  curveType: 'none',
  backgroundColor: '#2a2a2a',
  colors: ['#51cda0', '#6d78ad'],
  titleTextStyle: { color: 'white', bold: true, fontSize: 20, italic: false },
  hAxis: {
    title: 'Life Expectancy',
    titleTextStyle: { color: 'white', bold: false, italic: false },
    textStyle: { color: 'white', bold: false, italic: false },
    baselineColor: '#51cda0'
  },
},
vAxis: {
  title: 'Fertility Rate',
  titleTextStyle: { color: 'white', bold: false, italic: false },
  baselineColor: 'white',
  gridlines: { color: 'gray' },
  textStyle: { color: 'white', bold: false, italic: false }
},
animation: { duration: 1000, startup: true, easing: 'linear' },
tooltip: { textStyle: { color: '#871b47' }, showColorCode: true },
bubble: { textStyle: { color: 'white', fontSize: 11 } },
legend: { textStyle: { color: 'white', fontSize: 16 } }
};

var chart = new google.visualization.BubbleChart(document.getElementById('chart'));

```

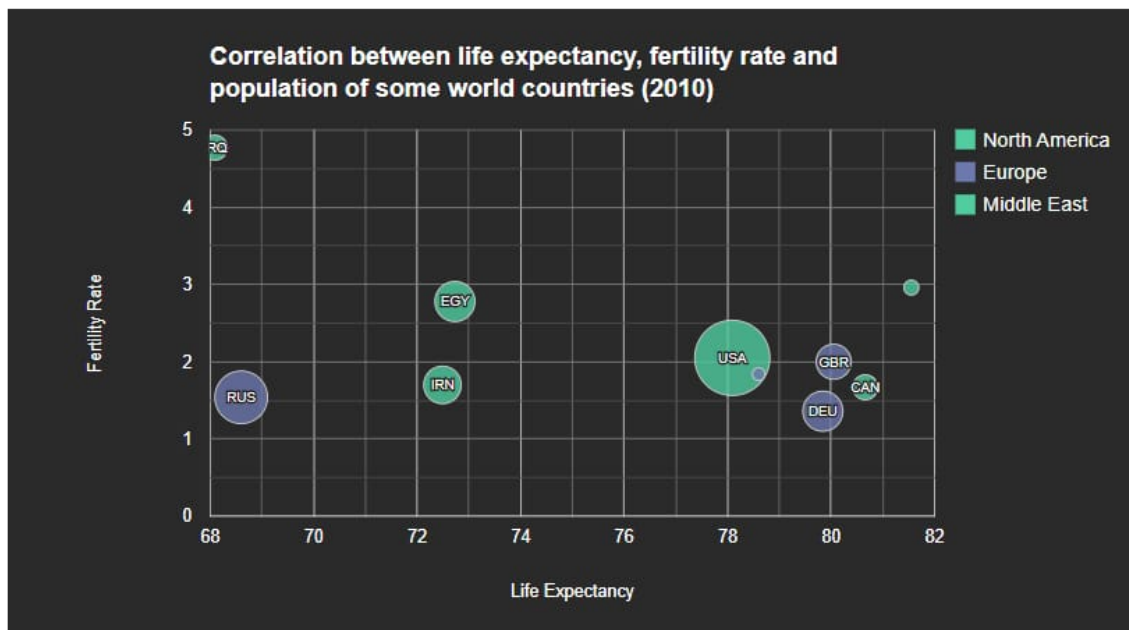
```

        chart.draw(data, options);
    }
</script>
</html>

```

**Output:**

**Using Google Charts API Basics draw chart(other type) bubble chart**



## 5(G) : Using Google API read JSON file and create Google Map.

```
<!DOCTYPE HTML>

<html>

<head>

  <title>Google API JSON to Google Map</title>

  <meta charset="utf-8">

  <meta name="author" content="SidPro" />

  <meta name="viewport" content="width=device-width,initial-scale=1.0" />

  <meta name="description" content="Using Google API read JSON file and create Google Map." />

  <meta name="keywords" content="Google,API,Geo Chart,Covid19 Geo Chart,Covid19" />

  <script src="js/jquery-3.5.1.min.js"></script>

  <script type="text/javascript" src="js/loader.js"></script>

</head>

<!-- Live link: https://sidpro-hash.github.io/HTML-Canvas/Data%20visulization%20JavaScript/google%20map%20json.html -->

<body>

  <h2>Read JSON file and create Google Map of Total COVID-19 cases in country [Date:01/04/2021]</h2>

  <div id="chart" style="width: 900px; height: 500px"></div>

</body>

<script>

  // Visualization API with the 'corechart' package.

  google.charts.load('current', {

    'packages': ['geochart'],

    // Note: you will need to get a mapsApiKey for your project.
```

```
// See: https://developers.google.com/chart/interactive/docs/basic\_load\_libs#load-  
settings
```

```
  'mapsApiKey': 'AIzaSyD-9tSrke72PouQMnMX-a7eZSW0jkFMBWY'  
});
```

```
google.charts.setOnLoadCallback(drawChart);
```

```
function drawChart() {
```

```
  $.ajax({
```

```
    url: "data/Covid-19.json",
```

```
    dataType: "json",
```

```
    type: "GET",
```

```
    contentType: "application/json; charset=utf-8",
```

```
    success: function (data) {
```

```
      var arr = [['Country', 'Total COVID-19 Cases']];
```

```
      $.each(data, function (k, v) {
```

```
        arr.push([v.city, v.n]);
```

```
      });
```

```
      var Data = google.visualization.arrayToDataTable(arr);
```

```
      //var Data = google.visualization.arrayToDataTable(covid19);
```

```
      var options = {
```

```
        colorAxis: { colors: ['#9bbb58', '#4f81bc', '#c0504e'] },
```

```
        backgroundColor: '#81d4fa',
```

```
        defaultColor: '#f5f5f5',
```

```
        tooltip: { textStyle: { color: '#871b47' }, showColorCode: true }
```

```
      };
```

```
      var chart = new google.visualization.GeoChart(document.getElementById('chart'));
```

```

        chart.draw(Data, options);
    },
    error: function (XMLHttpRequest, textStatus, errorThrown) {
        alert('File Error or Cross-origin police Error');
    }
});
}
</script>
</html>

```

### **Output:**

**Read JSON file and create Google Map of Total COVID-19 cases in country [Date:01/04/2021]**

