SUNWAY

INT'L BUSINESS SCHOOL

Program Name: (BCS )hons

Course Code: 2423

Course Name: Distributed and Parallel Computing

**Individual project**

Date of Submission: 10th September 2021

**Submitted By:**

Student Name: Pradip Dhakal

IUKL ID: 042902900047

Semester: 4th

Intake: September 2019

**Submitted To:**

Faculty Name: Manoj Gautam

Department: LMS

## Introduction

Web log analysis software (also known as a web log analyzer) is a type of web analytics software that parses a server log file from a web server and extracts information about when, how, and by whom a web server is accessed based on the values included in the log file. Reports are normally created right away, but data retrieved from log files can also be kept in a database, allowing for further analysis. This web application is made by using HTML/CSS/bootstrap/ C++. I have also implemented C++ thread in this program. The log file will break into 4 different parts using 4 threads and parse.

## Technologies

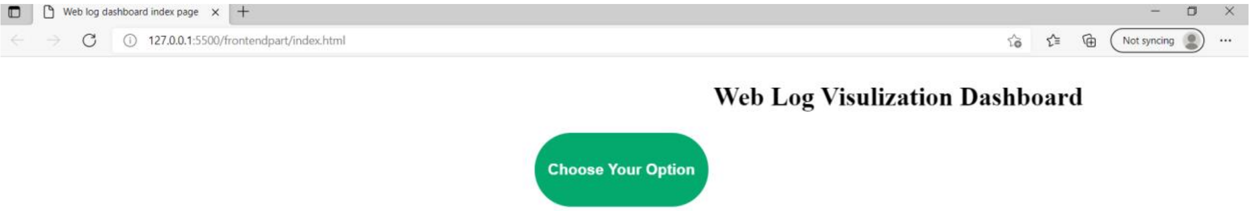
- HTML/CSS (for frontend webapp)
- C++ (for parsing log file)
- Json ( for creating server)

## Feature of the Web Application (Weblogalanyzer)

- It will show country wise data
- It will show operating system wise data
- It will show user access time wise data
- It will show browser wise data
- Graphical representation
- Easy to understand
- Many more

## Testing

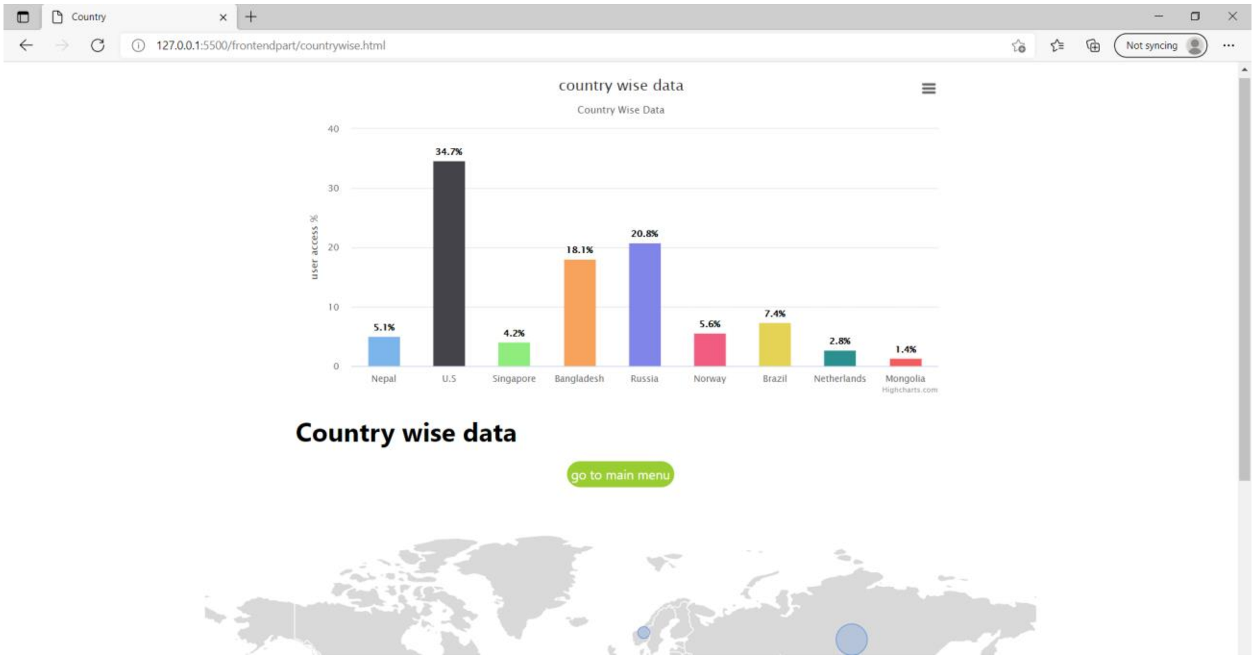
### index page



Web log dashboard index page
127.0.0.1:5500/frontendpart/index.html
Not syncing

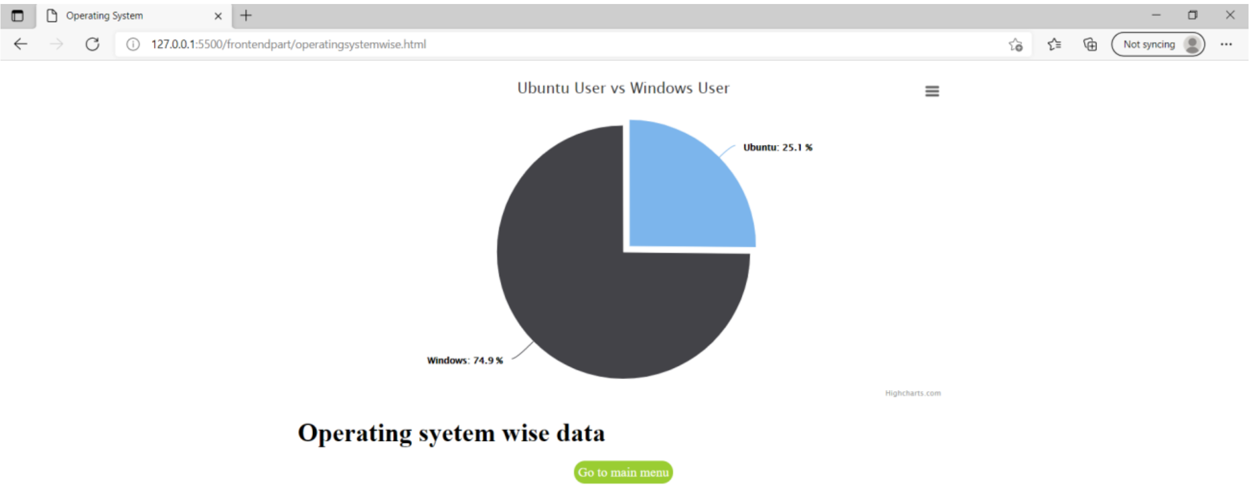
Web Log Visulization Dashboard

## Country Wise data



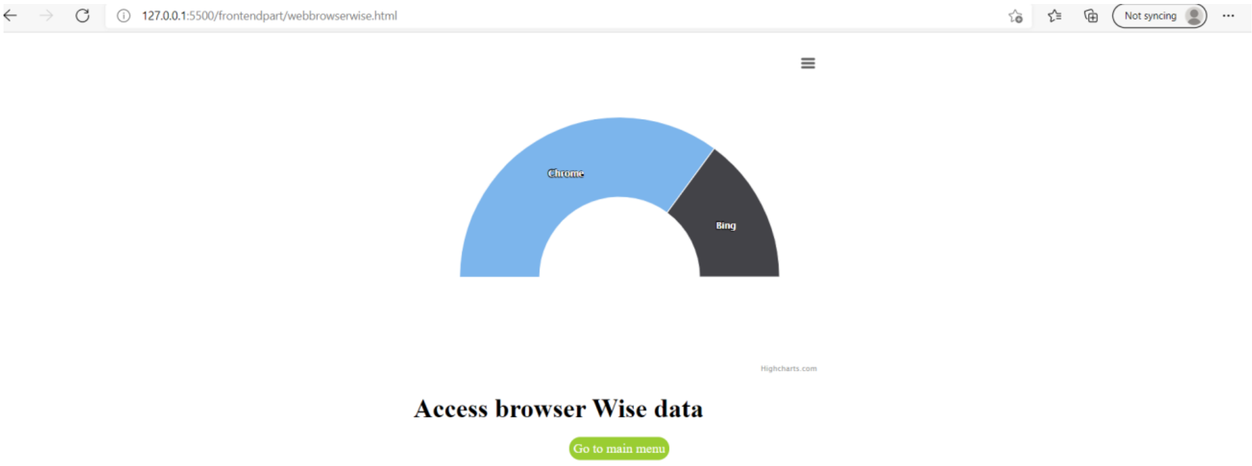
Country wise data

## Operating System wise data



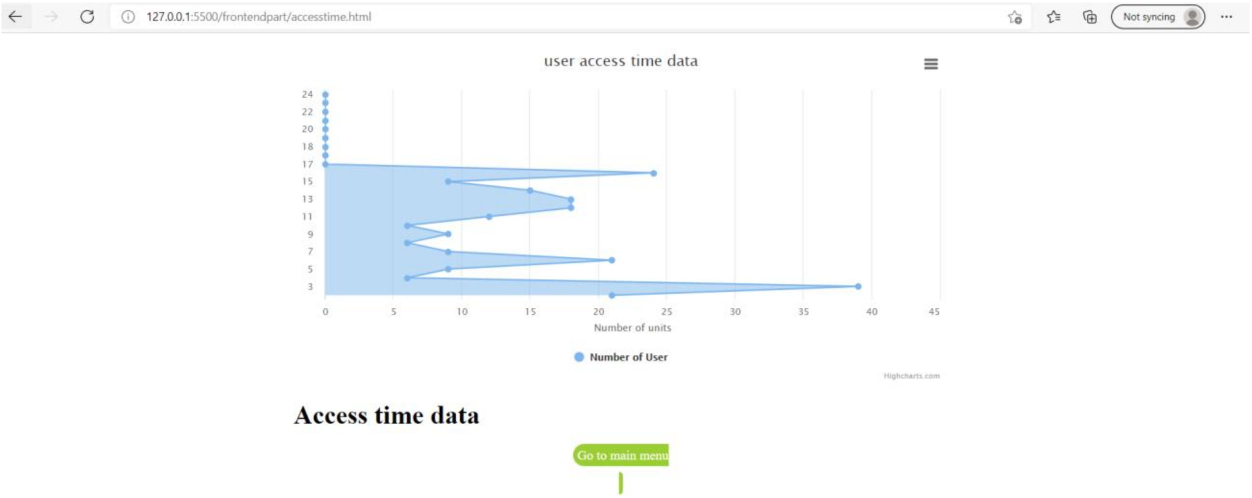
Operating syetem wise data

## Browser wise data



Access browser Wise data

## User access time wise data



Access time data

## Appendix

**Github link: https://github.com/Pradeep-Dhakal/Apache_logfile_visulization**

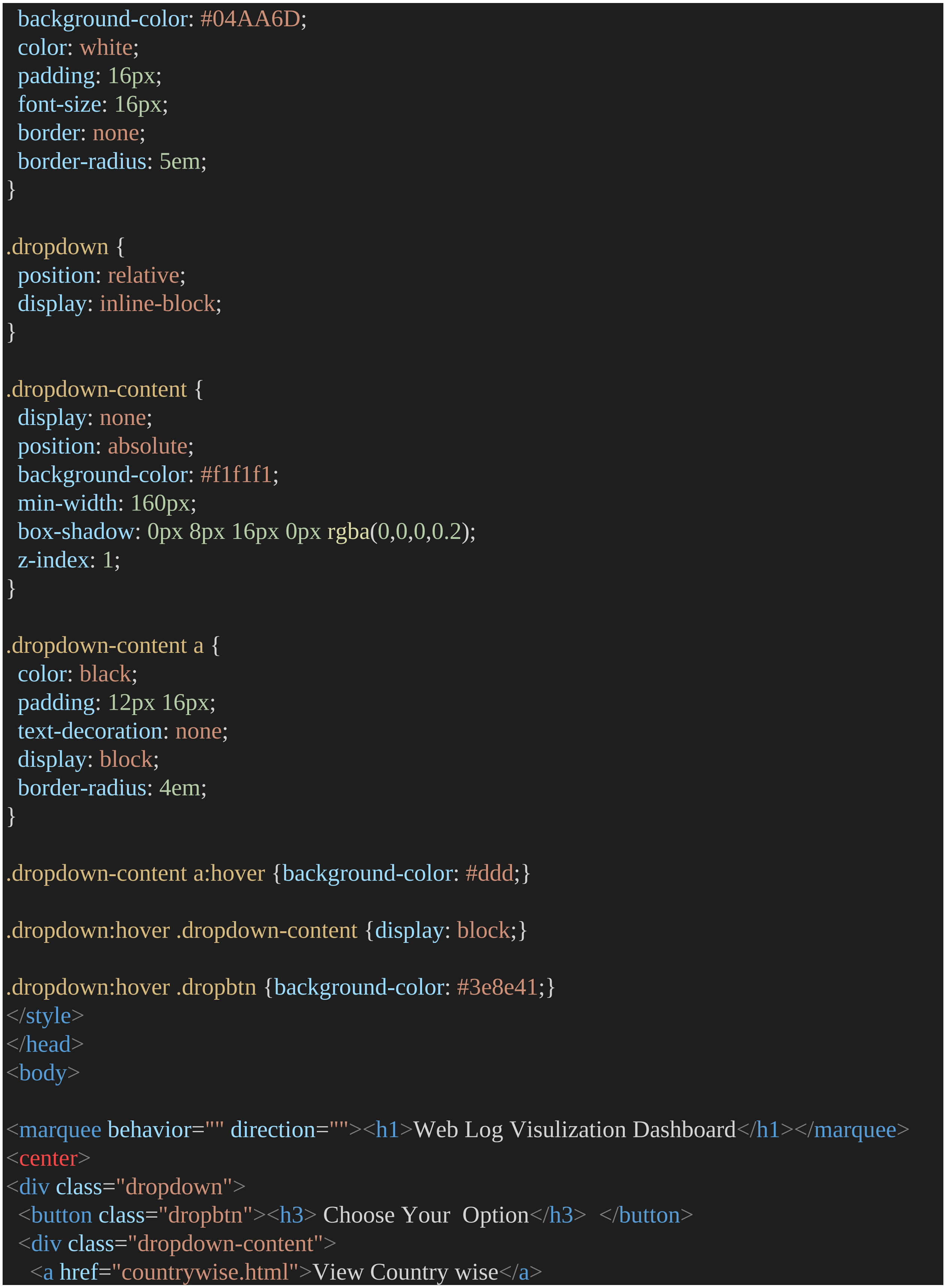
**To start this web app**

**Step1:** npm install -g json-server

**Step2:** json-server –watch loginfo.json

**Index.html**

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Web log dashboard index page </title>
<style>
.dropbtn {
```



```
  background-color: #04AA6D;
  color: white;
  padding: 16px;
  font-size: 16px;
  border: none;
  border-radius: 5em;
}

.dropdown {
  position: relative;
  display: inline-block;
}

.dropdown-content {
  display: none;
  position: absolute;
  background-color: #f1f1f1;
  min-width: 160px;
  box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
  z-index: 1;
}

.dropdown-content a {
  color: black;
  padding: 12px 16px;
  text-decoration: none;
  display: block;
  border-radius: 4em;
}

.dropdown-content a:hover {background-color: #ddd;}

.dropdown:hover .dropdown-content {display: block;}

.dropdown:hover .dropbtn {background-color: #3e8e41;}
</style>
</head>
<body>

<marquee behavior="" direction=""><h1>Web Log Visulization Dashboard</h1></marquee>
<center>
<div class="dropdown">
  <button class="dropbtn"><h3> Choose Your  Option</h3>  </button>
  <div class="dropdown-content">
    <a href="countrywise.html">View Country wise</a>
    <a href="operatingsystemwise.html">View OS wise</a>
    <a href="accesstime.html">view access time wise</a>
    <a href="webbrowserwise.html"> view Access browser wise</a>

  </div>
</div>
</center>

</body>
</html>
```

### Accesstime.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Time</title>

    <style>
#container {
  height: 410px;
}

.highcharts-figure, .highcharts-data-table table {
  min-width: 320px;
  max-width: 800px;
  margin: 1em auto;
}

.highcharts-data-table table {
  font-family: Verdana, sans-serif;
  border-collapse: collapse;
  border: 1px solid #EBEBEB;
  margin: 10px auto;
  text-align: center;
  width: 100%;
  max-width: 500px;
}
.highcharts-data-table caption {
  padding: 1em 0;
  font-size: 1.2em;
  color: #555;
}
.highcharts-data-table th {
  font-weight: 600;
  padding: 0.5em;
}
.highcharts-data-table td, .highcharts-data-table th, .highcharts-data-table caption {
  padding: 0.5em;
}
.highcharts-data-table thead tr, .highcharts-data-table tr:nth-child(even) {
  background: #f8f8f8;
}
.highcharts-data-table tr:hover {
  background: #f1f7ff;
}
a{
    background-color:yellowgreen;
    text-decoration: none;
    width: 20px;
    padding: 5px 5px;
    color: rgb(255, 255, 255);
    border-radius: 4em;

}
a:hover{
    background-color: rgb(129, 65, 36);
    text-decoration: none;
    width: 20px;
    padding: 7px 7px;
    color: black;
    border-radius: 4em;
}

    </style>
</head>
<body>
    <script src="https://code.highcharts.com/highcharts.js"></script>
    <script src="https://code.highcharts.com/modules/exporting.js"></script>
    <script src="https://code.highcharts.com/modules/export-data.js"></script>
    <script src="https://code.highcharts.com/modules/accessibility.js"></script>

    <figure class="highcharts-figure">
      <div id="container"></div>
      <p style="text-align: center;">
       <h1>Access time data</h1>
```

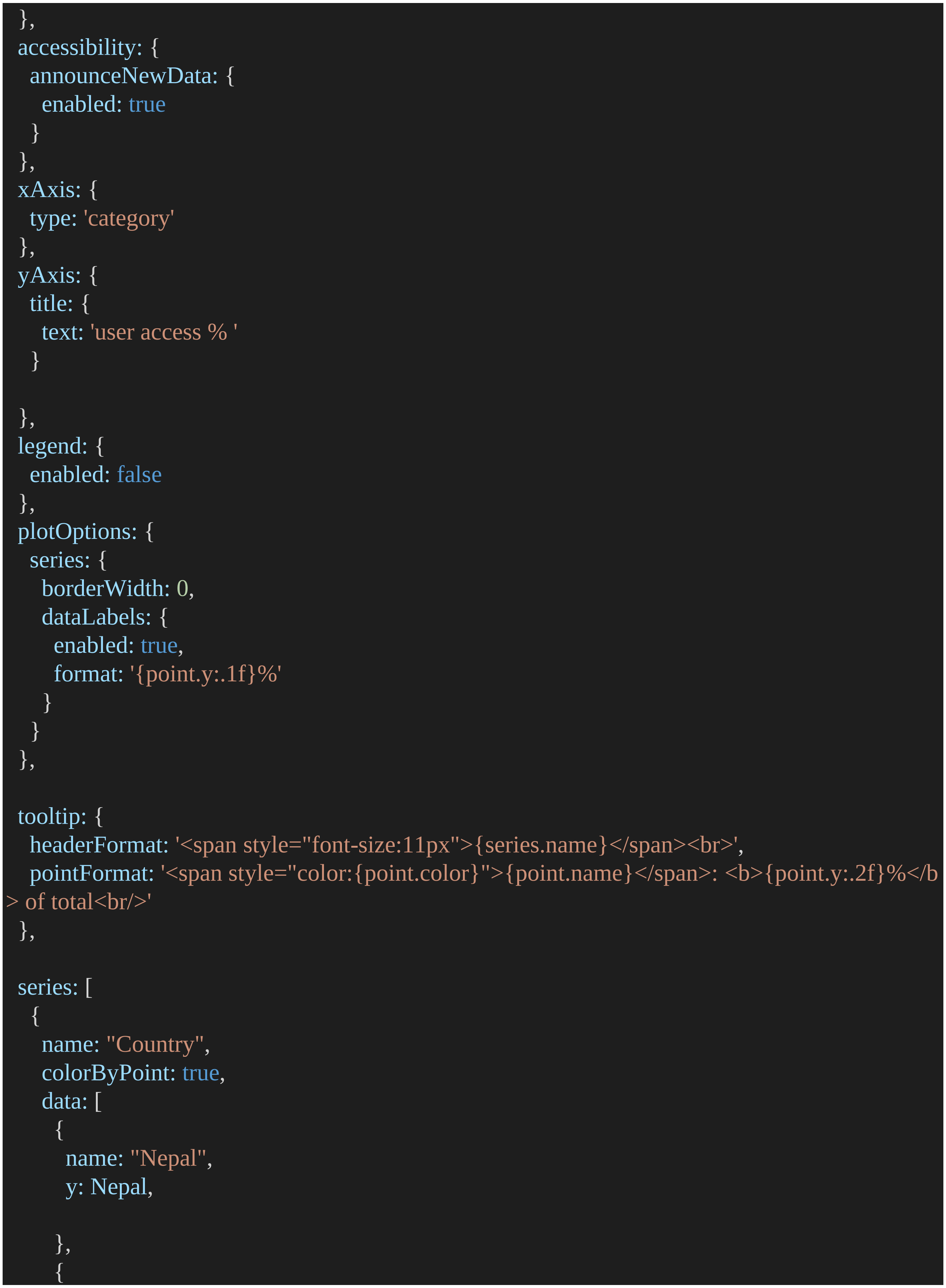
```
   </p>
  </figure>
  <center>
    <a href="index.html">Go to main menu <menu></menu></a>
  </center>
</body>
<script>
   const api = 'http://localhost:3000/Timeuser';
    async function getdata() {
      const repsond = await fetch(api);
      const data = await repsond.json();
      console.log(data);

  Highcharts.chart('container', {
 chart: {
  type: 'area',
  inverted: true
 },
 title: {
  text: 'user access time data'
 },
 accessibility: {
  keyboardNavigation: {
   seriesNavigation: {
    mode: 'serialize'
   }
  }
 },
 xAxis: {
  categories: [
   '24',
   '23',
   '22',
   '21',
   '20',
   '19',
   '18',
   '18',
   '17',
   '16',
   '15',
   '14',
   '13',
   '12',
   '11',
      '10',
      '9',
      '8',
      '7',
      '6',
      '5',
      '4',
      '3',
      '2',
      '1',
    ]
  },
  yAxis: {
    title: {
      text: 'Number of units'
    },
    allowDecimals: false,
    min: 0
  },
  plotOptions: {
    area: {
      fillOpacity: 0.5
    }
  },
  series: [{
    name: 'Number of User',
    data: [data[23].time23,data[22].time22,data[21].time21,data[20].time20,data[19].time19,data[18].time18,data[17].time17,data[16].time16,data[15].time15,data[14].time14,data[13].time13,data[12].time12,data[11].time11,data[10].time10,data[9].time09,data[8].time08,data[7].time07,data[6].time06,data[5].time05,data[4].time04,data[3].time03,data[2].time02,data[1].time01,data[0].time00]
  }]
});
}
getdata();
</script>
</html>
```

**Countrywise.html**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Country</title>
    <style>
        .highcharts-figure, .highcharts-data-table table {
  min-width: 310px;
  max-width: 800px;
  margin: 1em auto;
}

#container {
  height: 400px;
}

.highcharts-data-table table {
  font-family: Verdana, sans-serif;
  border-collapse: collapse;
  border: 1px solid #EBEBEB;
  margin: 10px auto;
  text-align: center;
  width: 100%;
  max-width: 500px;
}
.highcharts-data-table caption {
  padding: 1em 0;
  font-size: 1.2em;
  color: #555;
}
a{
    background-color: yellowgreen;
    text-decoration: initial;
    width: 20px;
    padding: 5px 5px;
    color: rgb(255, 255, 255);
    border: none;
    border-radius: 4em;

}
a:hover{
    background-color: rgb(129, 65, 36);
    text-decoration: none;
    width: 20px;
    padding: 7px 7px;
    color: black;
    border-radius: 4em;
}
```

```
.highcharts-data-table th {
  font-weight: 600;
  padding: 0.5em;
}
.highcharts-data-table td, .highcharts-data-table th, .highcharts-data-table caption {
  padding: 0.5em;
}
.highcharts-data-table thead tr, .highcharts-data-table tr:nth-child(even) {
  background: #f8f8f8;
}
.highcharts-data-table tr:hover {
  background: #f1f7ff;
}
body {
  font-family: -apple-system, BlinkMacSystemFont, "Segoe UI", Roboto, Helvetica, Arial, sans-serif, "Apple Color Emoji", "Segoe UI Emoji", "Segoe UI Symbol";
}

#chartdiv {
  width: 100%;
  height: 500px
}

    </style>
</head>
<body>
    <script src="https://code.highcharts.com/highcharts.js"></script>
    <script src="https://code.highcharts.com/modules/data.js"></script>
    <script src="https://code.highcharts.com/modules/drilldown.js"></script>
    <script src="https://code.highcharts.com/modules/exporting.js"></script>
    <script src="https://code.highcharts.com/modules/export-data.js"></script>
    <script src="https://code.highcharts.com/modules/accessibility.js"></script>

    <figure class="highcharts-figure">
      <div id="container"></div>
      <p style="text-align: center;">
   <H1>Country wise data </H1>
      </p>
    </figure>
    <center>
        <a href="index.html">go to main menu</a>
    </center><br><br><br>
    <script src="https://www.amcharts.com/lib/4/core.js"></script>
<script src="https://www.amcharts.com/lib/4/maps.js"></script>
<script src="https://www.amcharts.com/lib/4/geodata/worldLow.js"></script>
```

```
<script src="https://www.amcharts.com/lib/4/themes/animated.js"></script>
<div id="chartdiv"></div>
  <script>
     const api = 'http://localhost:3000/country';
    async function getdata() {
      const repsond = await fetch(api);
      const data = await repsond.json();
      console.log(data);
      var Mongolia,Bangladesh,Singapore,UnitedStates,Russia,Brazil,Netherlands,Norway,Nepal;
      datMongolia = data[0].Mongolia;
      datBangladesh = data[1].Bangladesh;
      datSingapore = data[2].Singapore;
      datUnitedStates = data[3].UnitedStates;
      datRussia = data[4].Russia;
      datBrazil = data[5].Brazil;
      datNetherlands = data[6].Netherlands;
      datNorway = data[7].Norway;
      datNepal = data[8].Nepal;
      var alltotal = datMongolia+datBangladesh+datSingapore+datUnitedStates+datRussia+datBrazil+datNetherlands+datNorway+datNepal;
      console.log("Total country log : "+alltotal);
      Mongolia = (datMongolia/alltotal)*100;
      Bangladesh = (datBangladesh/alltotal)*100;
      Singapore = (datSingapore/alltotal)*100;
      UnitedStates = (datUnitedStates/alltotal)*100;
      Russia = (datRussia/alltotal)*100;
      Brazil = (datBrazil/alltotal)*100;
      Netherlands = (datNetherlands/alltotal)*100;
      Norway = (datNorway/alltotal)*100;
      Nepal = (datNepal/alltotal)*100;




    // Create the chart
Highcharts.chart('container', {
 chart: {
  type: 'column'
 },
 title: {
  text: 'country wise data '
 },
 subtitle: {
  text: 'Country Wise Data'
```

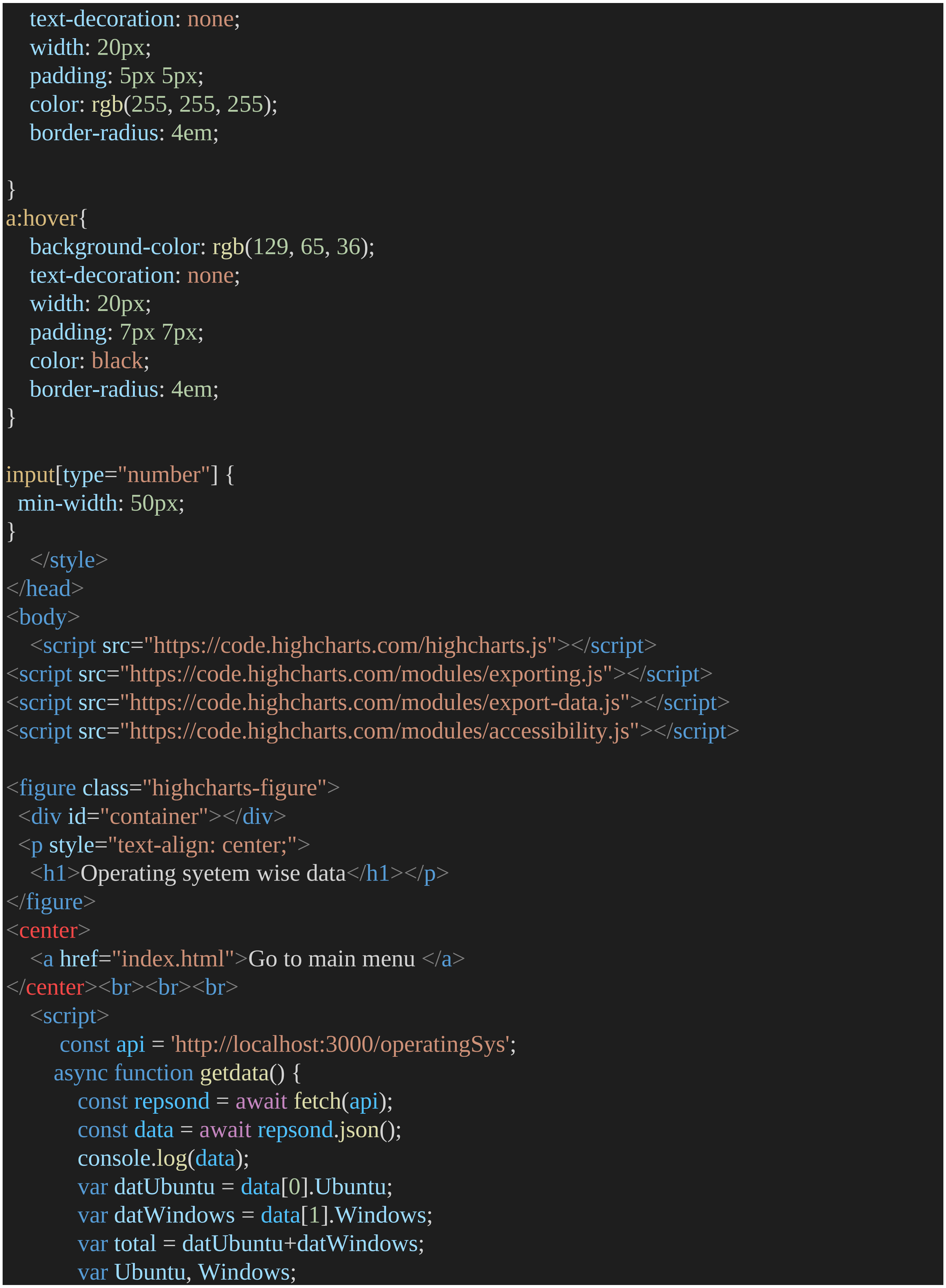


```
  },
  accessibility: {
    announceNewData: {
      enabled: true
    }
  },
  xAxis: {
    type: 'category'
  },
  yAxis: {
    title: {
      text: 'user access % '
    }

  },
  legend: {
    enabled: false
  },
  plotOptions: {
    series: {
      borderWidth: 0,
      dataLabels: {
        enabled: true,
        format: '{point.y:.1f}%'
      }
    }
  },

  tooltip: {
    headerFormat: '<span style="font-size:11px">{series.name}</span><br>',
    pointFormat: '<span style="color:{point.color}">{point.name}</span>: <b>{point.y:.2f}%</b
> of total<br/>'
  },

  series: [
    {
      name: "Country",
      colorByPoint: true,
      data: [
        {
          name: "Nepal",
          y: Nepal,

        },
        {
          name: "U.S",
          y: UnitedStates,

        },
        {
          name: "Singapore",
          y: Singapore,

        },
        {
          name: "Bangladesh",
          y: Bangladesh,

        },
        {
          name: "Russia",
          y: Russia,

        },
        {
          name: "Norway",
          y: Norway,

        },
        {
          name: "Brazil",
          y: Brazil,

        },
        {
          name: "Netherlands",
          y: Netherlands,

        },

        {
          name: "Mongolia",
          y: Mongolia,

        }
      ]
    }
  ],

});
```

```
am4core.useTheme(am4themes_animated);
// Themes end

// Create map instance
var chart = am4core.create("chartdiv", am4maps.MapChart);

var title = chart.titles.create();

title.textAlign = "middle";

var latlong = {
  "AD": {"latitude":42.5, "longitude":1.5},
  "AE": {"latitude":24, "longitude":54},
  "AF": {"latitude":33, "longitude":65},
  "AG": {"latitude":17.05, "longitude":-61.8},
  "AI": {"latitude":18.25, "longitude":-63.1667},
  "AL": {"latitude":41, "longitude":20},
  "AM": {"latitude":40, "longitude":45},
  "AN": {"latitude":12.25, "longitude":-68.75},
  "AO": {"latitude":-12.5, "longitude":18.5},
  "AP": {"latitude":35, "longitude":105},
  "AQ": {"latitude":-90, "longitude":0},
  "AR": {"latitude":-34, "longitude":-64},
  "AS": {"latitude":-14.3333, "longitude":-170},
  "AT": {"latitude":47.3333, "longitude":13.3333},
  "AU": {"latitude":-27, "longitude":133},
  "AW": {"latitude":12.5, "longitude":-69.9667},
  "AZ": {"latitude":40.5, "longitude":47.5},
  "BA": {"latitude":44, "longitude":18},
  "BB": {"latitude":13.1667, "longitude":-59.5333},
  "BD": {"latitude":24, "longitude":90},
  "BE": {"latitude":50.8333, "longitude":4},
  "BF": {"latitude":13, "longitude":-2},
  "BG": {"latitude":43, "longitude":25},
  "BH": {"latitude":26, "longitude":50.55},
  "BI": {"latitude":-3.5, "longitude":30},
  "BJ": {"latitude":9.5, "longitude":2.25},
  "BM": {"latitude":32.3333, "longitude":-64.75},
  "BN": {"latitude":4.5, "longitude":114.6667},
  "BO": {"latitude":-17, "longitude":-65},
  "BR": {"latitude":-10, "longitude":-55},
  "BS": {"latitude":24.25, "longitude":-76},
  "BT": {"latitude":27.5, "longitude":90.5},
"BV": {"latitude":-54.4333, "longitude":3.4},
"BW": {"latitude":-22, "longitude":24},
"BY": {"latitude":53, "longitude":28},
"BZ": {"latitude":17.25, "longitude":-88.75},
"CA": {"latitude":54, "longitude":-100},
"CC": {"latitude":-12.5, "longitude":96.8333},
"CD": {"latitude":0, "longitude":25},
"CF": {"latitude":7, "longitude":21},
"CG": {"latitude":-1, "longitude":15},
"CH": {"latitude":47, "longitude":8},
"CI": {"latitude":8, "longitude":-5},
"CK": {"latitude":-21.2333, "longitude":-159.7667},
"CL": {"latitude":-30, "longitude":-71},
"CM": {"latitude":6, "longitude":12},
"CN": {"latitude":35, "longitude":105},
"CO": {"latitude":4, "longitude":-72},
"CR": {"latitude":10, "longitude":-84},
"CU": {"latitude":21.5, "longitude":-80},
"CV": {"latitude":16, "longitude":-24},
"CX": {"latitude":-10.5, "longitude":105.6667},
"CY": {"latitude":35, "longitude":33},
"CZ": {"latitude":49.75, "longitude":15.5},
"DE": {"latitude":51, "longitude":9},
"DJ": {"latitude":11.5, "longitude":43},
"DK": {"latitude":56, "longitude":10},
"DM": {"latitude":15.4167, "longitude":-61.3333},
"DO": {"latitude":19, "longitude":-70.6667},
"DZ": {"latitude":28, "longitude":3},
"EC": {"latitude":-2, "longitude":-77.5},
"EE": {"latitude":59, "longitude":26},
"EG": {"latitude":27, "longitude":30},
"EH": {"latitude":24.5, "longitude":-13},
"ER": {"latitude":15, "longitude":39},
"ES": {"latitude":40, "longitude":-4},
"ET": {"latitude":8, "longitude":38},
"EU": {"latitude":47, "longitude":8},
"FI": {"latitude":62, "longitude":26},
"FJ": {"latitude":-18, "longitude":175},
"FK": {"latitude":-51.75, "longitude":-59},
"FM": {"latitude":6.9167, "longitude":158.25},
"FO": {"latitude":62, "longitude":-7},
"FR": {"latitude":46, "longitude":2},
"GA": {"latitude":-1, "longitude":11.75},
"GB": {"latitude":54, "longitude":-2},
"GD": {"latitude":12.1167, "longitude":-61.6667},
"GE": {"latitude":42, "longitude":43.5},
"GF": {"latitude":4, "longitude":-53},
"GH": {"latitude":8, "longitude":-2},
"GI": {"latitude":36.1833, "longitude":-5.3667},
"GL": {"latitude":72, "longitude":-40},
"GM": {"latitude":13.4667, "longitude":-16.5667},
"GN": {"latitude":11, "longitude":-10},
"GP": {"latitude":16.25, "longitude":-61.5833},
"GQ": {"latitude":2, "longitude":10},
"GR": {"latitude":39, "longitude":22},
"GS": {"latitude":-54.5, "longitude":-37},
"GT": {"latitude":15.5, "longitude":-90.25},
"GU": {"latitude":13.4667, "longitude":144.7833},
"GW": {"latitude":12, "longitude":-15},
"GY": {"latitude":5, "longitude":-59},
"HK": {"latitude":22.25, "longitude":114.1667},
"HM": {"latitude":-53.1, "longitude":72.5167},
"HN": {"latitude":15, "longitude":-86.5},
"HR": {"latitude":45.1667, "longitude":15.5},
"HT": {"latitude":19, "longitude":-72.4167},
"HU": {"latitude":47, "longitude":20},
"ID": {"latitude":-5, "longitude":120},
"IE": {"latitude":53, "longitude":-8},
"IL": {"latitude":31.5, "longitude":34.75},
"IN": {"latitude":20, "longitude":77},
"IO": {"latitude":-6, "longitude":71.5},
"IQ": {"latitude":33, "longitude":44},
"IR": {"latitude":32, "longitude":53},
"IS": {"latitude":65, "longitude":-18},
"IT": {"latitude":42.8333, "longitude":12.8333},
"JM": {"latitude":18.25, "longitude":-77.5},
"JO": {"latitude":31, "longitude":36},
"JP": {"latitude":36, "longitude":138},
"KE": {"latitude":1, "longitude":38},
"KG": {"latitude":41, "longitude":75},
"KH": {"latitude":13, "longitude":105},
"KI": {"latitude":1.4167, "longitude":173},
"KM": {"latitude":-12.1667, "longitude":44.25},
"KN": {"latitude":17.3333, "longitude":-62.75},
"KP": {"latitude":40, "longitude":127},
"KR": {"latitude":37, "longitude":127.5},
"KW": {"latitude":29.3375, "longitude":47.6581},
"KY": {"latitude":19.5, "longitude":-80.5},
"KZ": {"latitude":48, "longitude":68},
"LA": {"latitude":18, "longitude":105},
"LB": {"latitude":33.8333, "longitude":35.8333},
"LC": {"latitude":13.8833, "longitude":-61.1333},
"LI": {"latitude":47.1667, "longitude":9.5333},
"LK": {"latitude":7, "longitude":81},
"LR": {"latitude":6.5, "longitude":-9.5},
"LS": {"latitude":-29.5, "longitude":28.5},
"LT": {"latitude":55, "longitude":24},
"LU": {"latitude":49.75, "longitude":6},
"LV": {"latitude":57, "longitude":25},
"LY": {"latitude":25, "longitude":17},
"MA": {"latitude":32, "longitude":-5},
"MC": {"latitude":43.7333, "longitude":7.4},
"MD": {"latitude":47, "longitude":29},
"ME": {"latitude":42.5, "longitude":19.4},
"MG": {"latitude":-20, "longitude":47},
"MH": {"latitude":9, "longitude":168},
"MK": {"latitude":41.8333, "longitude":22},
"ML": {"latitude":17, "longitude":-4},
"MM": {"latitude":22, "longitude":98},
"MN": {"latitude":46, "longitude":105},
"MO": {"latitude":22.1667, "longitude":113.55},
"MP": {"latitude":15.2, "longitude":145.75},
"MQ": {"latitude":14.6667, "longitude":-61},
"MR": {"latitude":20, "longitude":-12},
"MS": {"latitude":16.75, "longitude":-62.2},
"MT": {"latitude":35.8333, "longitude":14.5833},
"MU": {"latitude":-20.2833, "longitude":57.55},
"MV": {"latitude":3.25, "longitude":73},
"MW": {"latitude":-13.5, "longitude":34},
"MX": {"latitude":23, "longitude":-102},
"MY": {"latitude":2.5, "longitude":112.5},
"MZ": {"latitude":-18.25, "longitude":35},
"NA": {"latitude":-22, "longitude":17},
"NC": {"latitude":-21.5, "longitude":165.5},
"NE": {"latitude":16, "longitude":8},
"NF": {"latitude":-29.0333, "longitude":167.95},
"NG": {"latitude":10, "longitude":8},
"NI": {"latitude":13, "longitude":-85},
"NL": {"latitude":52.5, "longitude":5.75},
"NO": {"latitude":62, "longitude":10},
"NP": {"latitude":28, "longitude":84},
"NR": {"latitude":-0.5333, "longitude":166.9167},
"NU": {"latitude":-19.0333, "longitude":-169.8667},
"NZ": {"latitude":-41, "longitude":174},
"OM": {"latitude":21, "longitude":57},
"PA": {"latitude":9, "longitude":-80},
"PE": {"latitude":-10, "longitude":-76},
"PF": {"latitude":-15, "longitude":-140},
"PG": {"latitude":-6, "longitude":147},
"PH": {"latitude":13, "longitude":122},
"PK": {"latitude":30, "longitude":70},
"PL": {"latitude":52, "longitude":20},
"PM": {"latitude":46.8333, "longitude":-56.3333},
"PR": {"latitude":18.25, "longitude":-66.5},
"PS": {"latitude":32, "longitude":35.25},
"PT": {"latitude":39.5, "longitude":-8},
"PW": {"latitude":7.5, "longitude":134.5},
"PY": {"latitude":-23, "longitude":-58},
"QA": {"latitude":25.5, "longitude":51.25},
"RE": {"latitude":-21.1, "longitude":55.6},
"RO": {"latitude":46, "longitude":25},
"RS": {"latitude":44, "longitude":21},
"RU": {"latitude":60, "longitude":100},
"RW": {"latitude":-2, "longitude":30},
"SA": {"latitude":25, "longitude":45},
"SB": {"latitude":-8, "longitude":159},
"SC": {"latitude":-4.5833, "longitude":55.6667},
"SD": {"latitude":15, "longitude":30},
"SE": {"latitude":62, "longitude":15},
"SG": {"latitude":1.3667, "longitude":103.8},
"SH": {"latitude":-15.9333, "longitude":-5.7},
"SI": {"latitude":46, "longitude":15},
"SJ": {"latitude":78, "longitude":20},
"SK": {"latitude":48.6667, "longitude":19.5},
"SL": {"latitude":8.5, "longitude":-11.5},
"SM": {"latitude":43.7667, "longitude":12.4167},
"SN": {"latitude":14, "longitude":-14},
"SO": {"latitude":10, "longitude":49},
"SR": {"latitude":4, "longitude":-56},
"ST": {"latitude":1, "longitude":7},
"SV": {"latitude":13.8333, "longitude":-88.9167},
"SY": {"latitude":35, "longitude":38},
"SZ": {"latitude":-26.5, "longitude":31.5},
"TC": {"latitude":21.75, "longitude":-71.5833},
"TD": {"latitude":15, "longitude":19},
"TF": {"latitude":-43, "longitude":67},
"TG": {"latitude":8, "longitude":1.1667},
"TH": {"latitude":15, "longitude":100},
"TJ": {"latitude":39, "longitude":71},
"TK": {"latitude":-9, "longitude":-172},
  "TM": {"latitude":40, "longitude":60},
  "TN": {"latitude":34, "longitude":9},
  "TO": {"latitude":-20, "longitude":-175},
  "TR": {"latitude":39, "longitude":35},
  "TT": {"latitude":11, "longitude":-61},
  "TV": {"latitude":-8, "longitude":178},
  "TW": {"latitude":23.5, "longitude":121},
  "TZ": {"latitude":-6, "longitude":35},
  "UA": {"latitude":49, "longitude":32},
  "UG": {"latitude":1, "longitude":32},
  "UM": {"latitude":19.2833, "longitude":166.6},
  "US": {"latitude":38, "longitude":-97},
  "UY": {"latitude":-33, "longitude":-56},
  "UZ": {"latitude":41, "longitude":64},
  "VA": {"latitude":41.9, "longitude":12.45},
  "VC": {"latitude":13.25, "longitude":-61.2},
  "VE": {"latitude":8, "longitude":-66},
  "VG": {"latitude":18.5, "longitude":-64.5},
  "VI": {"latitude":18.3333, "longitude":-64.8333},
  "VN": {"latitude":16, "longitude":106},
  "VU": {"latitude":-16, "longitude":167},
  "WF": {"latitude":-13.3, "longitude":-176.2},
  "WS": {"latitude":-13.5833, "longitude":-172.3333},
  "YE": {"latitude":15, "longitude":48},
  "YT": {"latitude":-12.8333, "longitude":45.1667},
  "ZA": {"latitude":-29, "longitude":24},
  "ZM": {"latitude":-15, "longitude":30},
  "ZW": {"latitude":-20, "longitude":30}
};

var mapData = [
{ "id":"BD", "name":"Bangladesh", "value":datBangladesh, "color": chart.colors.getIndex(0) },
{ "id":"MN", "name":"Mongolia", "value":datMongolia, "color": chart.colors.getIndex(0) },
{ "id":"SG", "name":"Singapore", "value":datSingapore, "color": chart.colors.getIndex(0) },
{ "id":"US", "name":"United States", "value":datUnitedStates, "color":chart.colors.getIndex(4) },
{ "id":"RU", "name":"Russia", "value":datRussia, "color":chart.colors.getIndex(1) },
{ "id":"BR", "name":"Brazil", "value":datBrazil, "color":chart.colors.getIndex(3) },
{ "id":"NL", "name":"Netherlands", "value":datNetherlands, "color":chart.colors.getIndex(1) },
{ "id":"NO", "name":"Norway", "value":datNorway, "color":chart.colors.getIndex(1) },
{ "id":"NP", "name":"Nepal", "value":datNepal, "color": chart.colors.getIndex(0) }
];

// Add lat/long information to data
for(var i = 0; i < mapData.length; i++) {
  mapData[i].latitude = latlong[mapData[i].id].latitude;
  mapData[i].longitude = latlong[mapData[i].id].longitude;
}

// Set map definition
chart.geodata = am4geodata_worldLow;

// Set projection
chart.projection = new am4maps.projections.Miller();

// Create map polygon series
var polygonSeries = chart.series.push(new am4maps.MapPolygonSeries());
polygonSeries.exclude = ["AQ"];
polygonSeries.useGeodata = true;

var imageSeries = chart.series.push(new am4maps.MapImageSeries());
imageSeries.data = mapData;
imageSeries.dataFields.value = "value";

var imageTemplate = imageSeries.mapImages.template;
imageTemplate.propertyFields.latitude = "latitude";
imageTemplate.propertyFields.longitude = "longitude";
imageTemplate.nonScaling = true

var circle = imageTemplate.createChild(am4core.Circle);
circle.fillOpacity = 0.3;
circle.propertyFields.fill = "color";
circle.propertyFields.stroke = "color";
circle.propertyFields.strokeWidth = 30;
circle.tooltipText = "{name}: [bold]{value}[/]";

imageSeries.heatRules.push({
  "target": circle,
  "property": "radius",
  "min": 4,
  "max": 30,
  "dataField": "value"
})
}
        getdata();
    </script>
</body>
</html>
```

**Operatingsystemwise.html**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Operating System</title>
    <style>
        .highcharts-figure, .highcharts-data-table table {
  min-width: 320px;
  max-width: 800px;
  margin: 1em auto;
}

.highcharts-data-table table {
  font-family: Verdana, sans-serif;
  border-collapse: collapse;
  border: 1px solid #EBEBEB;
  margin: 10px auto;
  text-align: center;
  width: 100%;
  max-width: 500px;
}
.highcharts-data-table caption {
  padding: 1em 0;
  font-size: 1.2em;
  color: #555;
}
.highcharts-data-table th {
  font-weight: 600;
  padding: 0.5em;
}
.highcharts-data-table td, .highcharts-data-table th, .highcharts-data-table caption {
  padding: 0.5em;
}
.highcharts-data-table thead tr, .highcharts-data-table tr:nth-child(even) {
  background: #f8f8f8;
}
.highcharts-data-table tr:hover {
  background: #f1f7ff;
}
a{
    background-color:yellowgreen;
```



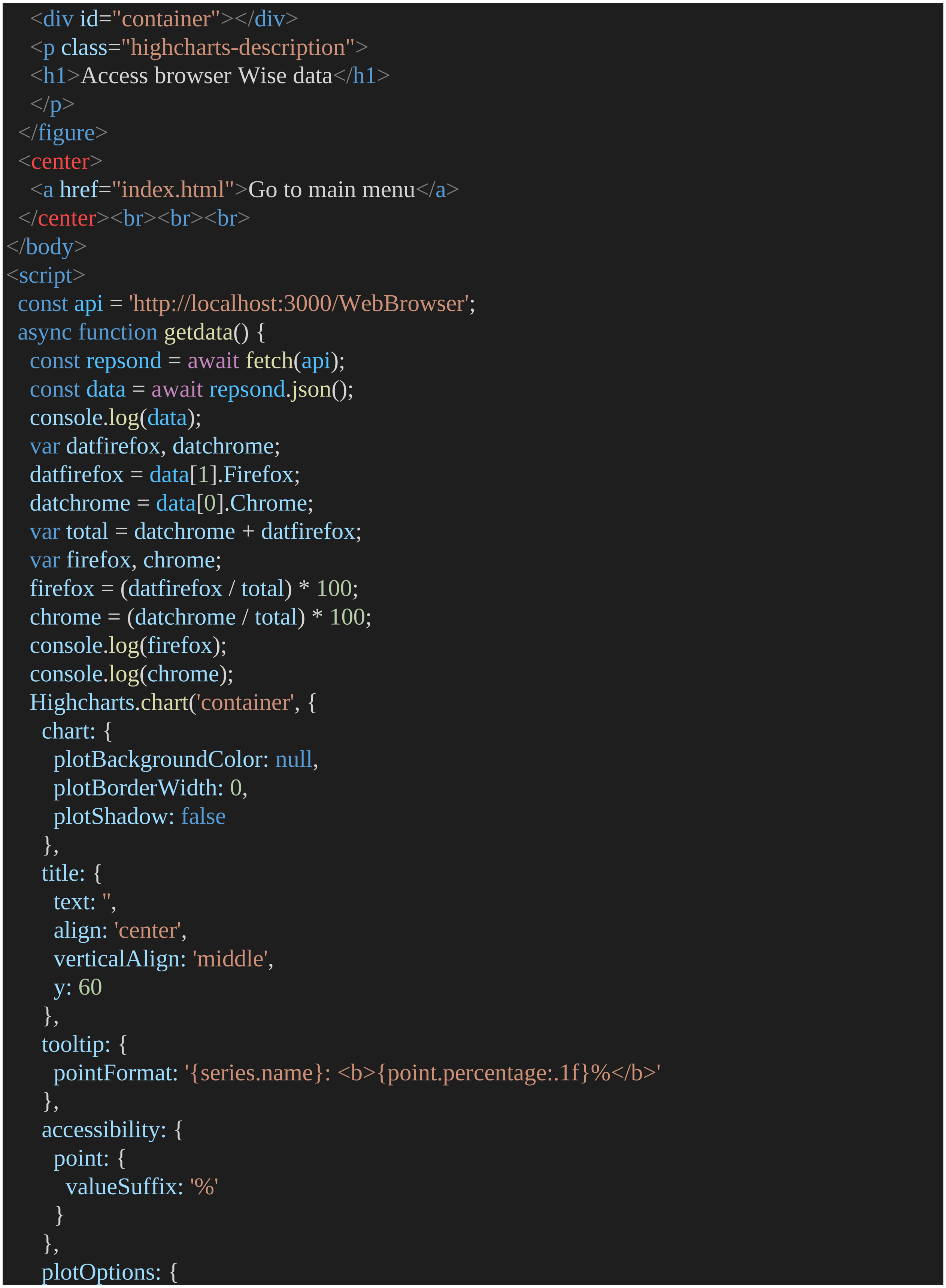
```
    text-decoration: none;
    width: 20px;
    padding: 5px 5px;
    color: rgb(255, 255, 255);
    border-radius: 4em;

}
a:hover{
    background-color: rgb(129, 65, 36);
    text-decoration: none;
    width: 20px;
    padding: 7px 7px;
    color: black;
    border-radius: 4em;
}

input[type="number"] {
  min-width: 50px;
}
    </style>
</head>
<body>
    <script src="https://code.highcharts.com/highcharts.js"></script>
<script src="https://code.highcharts.com/modules/exporting.js"></script>
<script src="https://code.highcharts.com/modules/export-data.js"></script>
<script src="https://code.highcharts.com/modules/accessibility.js"></script>

<figure class="highcharts-figure">
  <div id="container"></div>
  <p style="text-align: center;">
    <h1>Operating syetem wise data</h1></p>
</figure>
<center>
    <a href="index.html">Go to main menu </a>
</center><br><br><br>
    <script>
         const api = 'http://localhost:3000/operatingSys';
        async function getdata() {
            const repsond = await fetch(api);
            const data = await repsond.json();
            console.log(data);
            var datUbuntu = data[0].Ubuntu;
            var datWindows = data[1].Windows;
            var total = datUbuntu+datWindows;
            var Ubuntu, Windows;
```

```
      Ubuntu = (datUbuntu/total)*100;
      Windows = (datWindows/total)*100;

Highcharts.chart('container', {
  chart: {
    plotBackgroundColor: null,
    plotBorderWidth: null,
    plotShadow: false,
    type: 'pie'
  },
  title: {
    text: 'Ubuntu User vs Windows User'
  },
  tooltip: {
    pointFormat: '{series.name}: <b>{point.percentage:.1f}%</b>'
  },
  accessibility: {
    point: {
      valueSuffix: '%'
    }
  },
  plotOptions: {
    pie: {
      allowPointSelect: true,
      cursor: 'pointer',
      dataLabels: {
        enabled: true,
        format: '<b>{point.name}</b>: {point.percentage:.1f} %'
      }
    }
  },
  series: [{
    name: 'Brands',
    colorByPoint: true,
    data: [{
      name: 'Ubuntu',
      y: datUbuntu,
      sliced: true,
      selected: true
    }, {
      name: 'Windows',
      y: Windows
    }]
  }]
});}
```

```
      getdata();
    </script>
</body>
</html>
```

**Webbrowserwise.html**

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Browser Wise</title>
  <style>
    .highcharts-figure,
    .highcharts-data-table table {
      min-width: 320px;
      max-width: 500px;
      margin: 1em auto;
    }

    #container {
      height: 400px;
    }

    .highcharts-data-table table {
      font-family: Verdana, sans-serif;
      border-collapse: collapse;
      border: 1px solid #EBEBEB;
      margin: 10px auto;
      text-align: center;
      width: 100%;
      max-width: 500px;
    }

    .highcharts-data-table caption {
      padding: 1em 0;
      font-size: 1.2em;
      color: #555;
    }

    .highcharts-data-table th {
      font-weight: 600;
      padding: 0.5em;
    }

    .highcharts-data-table td,
    .highcharts-data-table th,
    .highcharts-data-table caption {
      padding: 0.5em;
    }

    .highcharts-data-table thead tr,
    .highcharts-data-table tr:nth-child(even) {
      background: #f8f8f8;
    }

    .highcharts-data-table tr:hover {
      background: #f1f7ff;
    }

    a {
      background-color: yellowgreen;
      text-decoration: none;
      width: 20px;
      padding: 5px 5px;
      color: rgb(255, 255, 255);
      border-radius: 4em;

    }

    a:hover {
      background-color: rgb(129, 65, 36);
      text-decoration: none;
      width: 20px;
      padding: 7px 7px;
      color: black;
      border-radius: 4em;
    }
  </style>
</head>

<body>
  <script src="https://code.highcharts.com/highcharts.js"></script>
  <script src="https://code.highcharts.com/modules/exporting.js"></script>
  <script src="https://code.highcharts.com/modules/accessibility.js"></script>

  <figure class="highcharts-figure">
```



```
    <div id="container"></div>
    <p class="highcharts-description">
    <h1>Access browser Wise data</h1>
    </p>
  </figure>
  <center>
    <a href="index.html">Go to main menu</a>
  </center><br><br><br>
</body>
<script>
  const api = 'http://localhost:3000/WebBrowser';
  async function getdata() {
    const repsond = await fetch(api);
    const data = await repsond.json();
    console.log(data);
    var datfirefox, datchrome;
    datfirefox = data[1].Firefox;
    datchrome = data[0].Chrome;
    var total = datchrome + datfirefox;
    var firefox, chrome;
    firefox = (datfirefox / total) * 100;
    chrome = (datchrome / total) * 100;
    console.log(firefox);
    console.log(chrome);
    Highcharts.chart('container', {
      chart: {
        plotBackgroundColor: null,
        plotBorderWidth: 0,
        plotShadow: false
      },
      title: {
        text: '',
        align: 'center',
        verticalAlign: 'middle',
        y: 60
      },
      tooltip: {
        pointFormat: '{series.name}: <b>{point.percentage:.1f}%</b>'
      },
      accessibility: {
        point: {
          valueSuffix: '%'
        }
      },
      plotOptions: {
        pie: {
          dataLabels: {
            enabled: true,
            distance: -50,
            style: {
              fontWeight: 'bold',
              color: 'white'
            }
          },
          startAngle: -90,
          endAngle: 90,
          center: ['50%', '75%'],
          size: '110%'
        }
      },
      series: [{
        type: 'pie',
        name: 'Browser share',
        innerSize: '50%',
        data: [
          ['Chrome', chrome],
          ['Bing', firefox],
          {
            name: 'Other',
            y: 0,
            dataLabels: {
              enabled: false
            }
          }
        ]
      }]
    });


  }
  getdata();
</script>

</html>
```

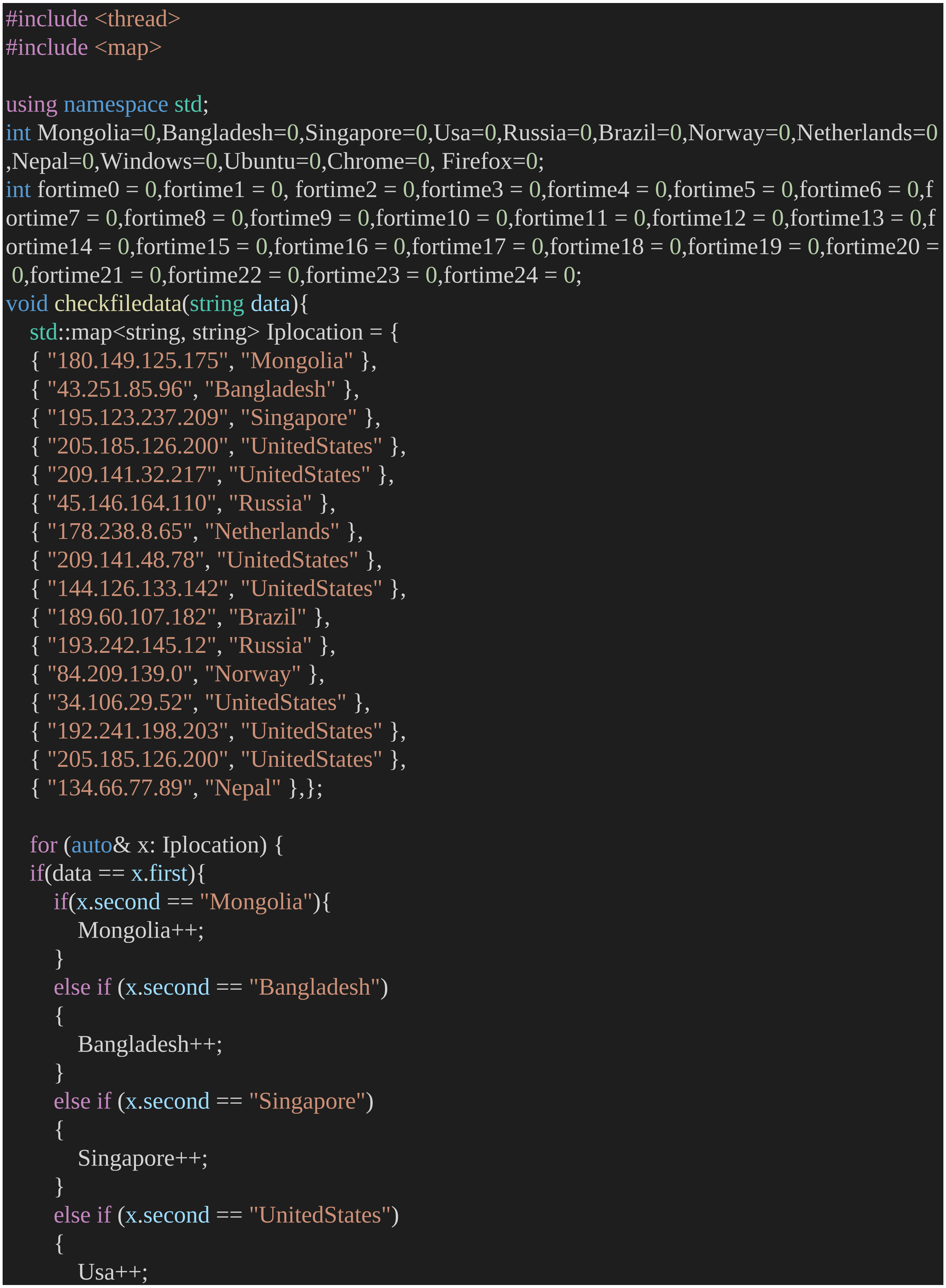
**Style.css**

```
*{
    font-family: Arial, Helvetica, sans-serif;
    margin: 0px 0px;
```

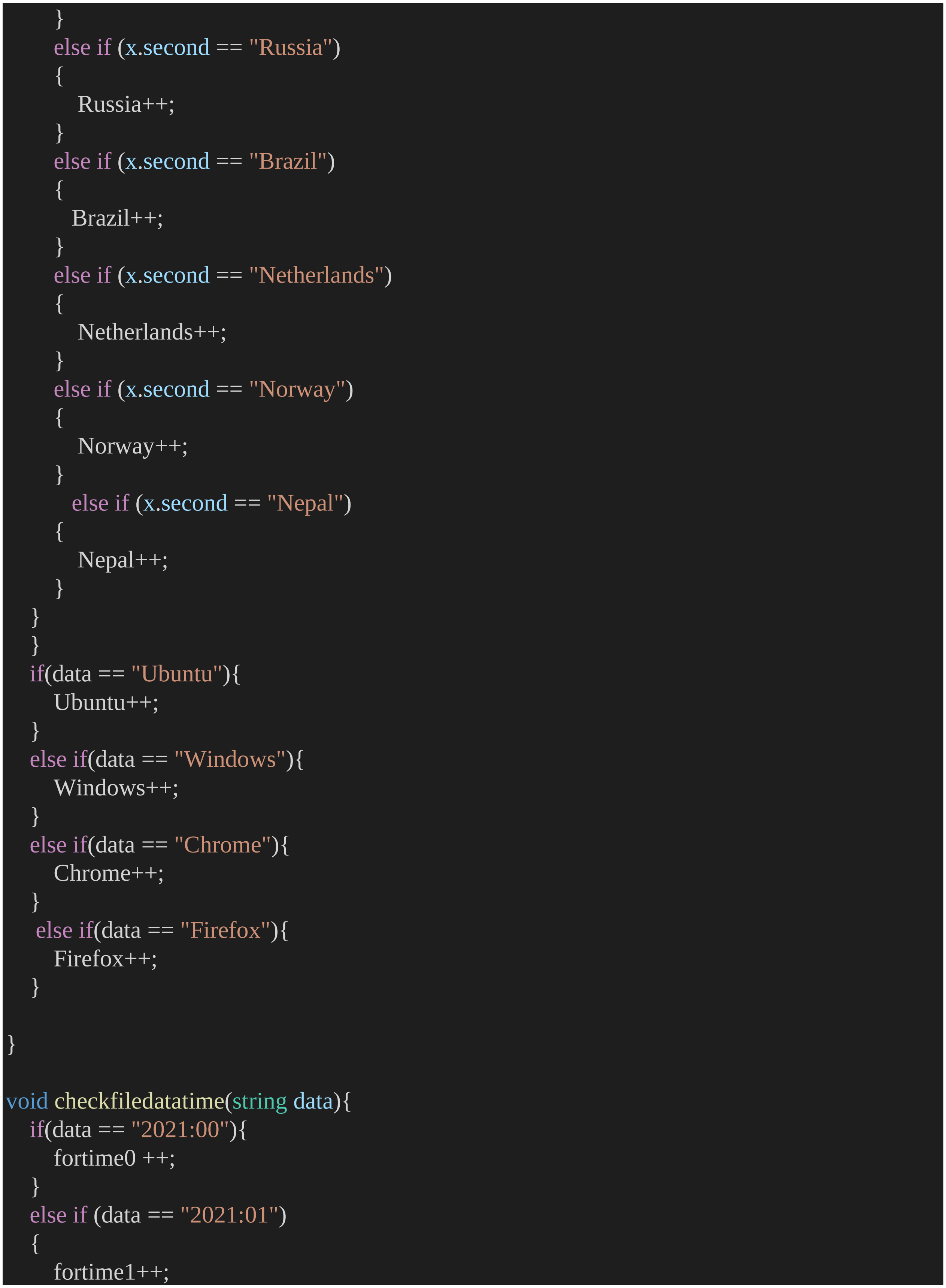
```
}
body{
    background-color: rgb(0, 0, 0);
}
.header{
    color: white;
    font-size: 40px;
    text-align: center;
    height: 60px;
    background-color: rgb(32, 32, 32);
}
.textbold{
    color: white;
    text-align: center;
    margin-top: 20px;
    font-weight: bold;
}
a{
    background-color: white;
    text-decoration: none;
    width: 20px;
    padding: 5px 5px;
    color: black;
    border-radius: 10%;

}
a:hover{
    background-color: rgb(36, 129, 78);
    text-decoration: none;
    width: 20px;
    padding: 7px 7px;
    color: black;
    border-radius: 10%;
}
footer{
    color: white;
    text-align: center;
    height: 60px;
    opacity: 75%;
    background-color: rgb(32, 32, 32);
}
.dropbtn {
    background-color: #3498DB;
    color: white;
    padding: 16px;
    font-size: 16px;
    border: none;
    cursor: pointer;
  }

  .dropbtn:hover, .dropbtn:focus {
    background-color: #2980B9;
  }

  .dropdown {
    position: relative;
    display: inline-block;
  }

  .dropdown-content {
    display: none;
    position: absolute;
    background-color: #f1f1f1;
    min-width: auto;
    overflow: auto;
    box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
    z-index: 1;
  }

  .dropdown-content a {
    color: black;
    padding: 12px 16px;
    text-decoration:none;
    display: block;
  }

  .dropdown a:hover {background-color: #ddd;}

  .show {display: block;
}
.abc{
  height: inherit;
  width: fit-content;

}
```

**Backend.CPP**

```
#include <iostream>
#include <fstream>
```



```
#include <thread>
#include <map>

using namespace std;
int Mongolia=0,Bangladesh=0,Singapore=0,Usa=0,Russia=0,Brazil=0,Norway=0,Netherlands=0
,Nepal=0,Windows=0,Ubuntu=0,Chrome=0, Firefox=0;
int fortime0 = 0,fortime1 = 0, fortime2 = 0,fortime3 = 0,fortime4 = 0,fortime5 = 0,fortime6 = 0,f
ortime7 = 0,fortime8 = 0,fortime9 = 0,fortime10 = 0,fortime11 = 0,fortime12 = 0,fortime13 = 0,f
ortime14 = 0,fortime15 = 0,fortime16 = 0,fortime17 = 0,fortime18 = 0,fortime19 = 0,fortime20 =
 0,fortime21 = 0,fortime22 = 0,fortime23 = 0,fortime24 = 0;
void checkfiledata(string data){
    std::map<string, string> Iplocation = {
    { "180.149.125.175", "Mongolia" },
    { "43.251.85.96", "Bangladesh" },
    { "195.123.237.209", "Singapore" },
    { "205.185.126.200", "UnitedStates" },
    { "209.141.32.217", "UnitedStates" },
    { "45.146.164.110", "Russia" },
    { "178.238.8.65", "Netherlands" },
    { "209.141.48.78", "UnitedStates" },
    { "144.126.133.142", "UnitedStates" },
    { "189.60.107.182", "Brazil" },
    { "193.242.145.12", "Russia" },
    { "84.209.139.0", "Norway" },
    { "34.106.29.52", "UnitedStates" },
    { "192.241.198.203", "UnitedStates" },
    { "205.185.126.200", "UnitedStates" },
    { "134.66.77.89", "Nepal" },};

    for (auto& x: Iplocation) {
    if(data == x.first){
        if(x.second == "Mongolia"){
            Mongolia++;
        }
        else if (x.second == "Bangladesh")
        {
            Bangladesh++;
        }
        else if (x.second == "Singapore")
        {
            Singapore++;
        }
        else if (x.second == "UnitedStates")
        {
            Usa++;
```



```
        }
        else if (x.second == "Russia")
        {
            Russia++;
        }
        else if (x.second == "Brazil")
        {
           Brazil++;
        }
        else if (x.second == "Netherlands")
        {
            Netherlands++;
        }
        else if (x.second == "Norway")
        {
            Norway++;
        }
           else if (x.second == "Nepal")
        {
            Nepal++;
        }
    }
    }
    if(data == "Ubuntu"){
        Ubuntu++;
    }
    else if(data == "Windows"){
        Windows++;
    }
    else if(data == "Chrome"){
        Chrome++;
    }
     else if(data == "Firefox"){
        Firefox++;
    }

}

void checkfiledatatime(string data){
    if(data == "2021:00"){
        fortime0 ++;
    }
    else if (data == "2021:01")
    {
        fortime1++;
```

```
}
 else if (data == "2021:02")
{
   fortime2++;
}
 else if (data == "2021:03")
{
   fortime3++;
}
 else if (data == "2021:04")
{
   fortime4++;
}
 else if (data == "2021:05")
{
   fortime5++;
}
 else if (data == "2021:06")
{
   fortime6++;
}
 else if (data == "2021:07")
{
   fortime7++;
}
 else if (data == "2021:08")
{
   fortime8++;
}
 else if (data == "2021:09")
{
   fortime9++;
}
 else if (data == "2021:10")
{
   fortime10++;
}
 else if (data == "2021:11")
{
   fortime11++;
}
 else if (data == "2021:12")
{
   fortime12++;
}
```

```
 else if (data == "2021:13")
{
   fortime13++;
}
 else if (data == "2021:14")
{
   fortime14++;
}
 else if (data == "2021:15")
{
   fortime15++;
}
 else if (data == "2021:16")
{
   fortime16++;
}
 else if (data == "2021:17")
{
   fortime17++;
}
 else if (data == "2021:18")
{
   fortime18++;
}
 else if (data == "2021:19")
{
   fortime19++;
}
 else if (data == "2021:20")
{
   fortime20++;
}
 else if (data == "2021:21")
{
   fortime21++;
}
 else if (data == "2021:22")
{
   fortime22++;
}
 else if (data == "2021:23")
{
   fortime23++;
}
```

```
}



int main(){
string myText;
ifstream MyReadFile("logfile.log");

while (getline (MyReadFile, myText)) {
    string temp = "";
    for(int i = 0;i<myText.size();i++){
        int num = 0;
        if(!isspace(myText[i])){
            temp += myText[i];
        }
        else{
            std::thread t1(checkfiledata,temp);
            std::thread t2(checkfiledatatime,temp);
            std::thread t3(checkfiledatatime,temp);
            std::thread t4(checkfiledatatime,temp);

            t1.join();
            t2.join();
            t3.join();
            t4.join();
            temp = "";
        }
    }

}

MyReadFile.close();

ofstream MyFile("loginfo.json");
  MyFile <<"{"<<"\n";
  MyFile <<"\"country\" : [{\"Mongolia\":"<<Mongolia<<"},{\"Bangladesh\":"<<Bangladesh<<"},{\"Singapore\":"<<Singapore<<"},{\"UnitedStates\":"<<Usa<<"},{\"Russia\":"<<Russia<<"},{\"Brazil\":"<<Brazil<<"},{\"Netherlands\":"<<Netherlands<<"},{\"Norway\":"<<Norway<<"},{\"Nepal\":"<<Nepal<<"}],"<<'\n';
  MyFile <<"\"operatingSys\" : [{\"Ubuntu\":"<<Ubuntu<<"},{\"Windows\":"<<Windows<<"}],"<<'\n';
  MyFile <<"\"WebBrowser\" : [{\"Chrome\":"<<Chrome<<"},{\"Firefox\":"<<Firefox<<"}],"<<'\n';
  MyFile <<"\"Timeuser\" : [{\"time00\":"<<fortime0<<"},{\"time01\":"<<fortime1<<"},{\"time02\":"<<fortime2<<"},{\"time03\":"<<fortime3<<"},{\"time04\":"<<fortime4<<"},{\"time05\":"<<fortime5<<"},{\"time06\":"<<fortime6<<"},{\"time07\":"<<fortime7<<"},{\"time08\":"<<fortime8<<"},{\"time09\":"<<fortime9<<"},{\"time10\":"<<fortime10<<"},{\"time11\":"<<fortime11<<"},"<<'\n';
  MyFile <<"{\"time12\":"<<fortime12<<"},{\"time13\":"<<fortime13<<"},{\"time14\":"<<fortime14<<"},{\"time15\":"<<fortime15<<"},{\"time16\":"<<fortime16<<"},{\"time17\":"<<fortime17<<"},{\"time18\":"<<fortime18<<"},{\"time19\":"<<fortime19<<"},{\"time20\":"<<fortime20<<"},{\"time21\":"<<fortime21<<"},{\"time22\":"<<fortime22<<"},{\"time23\":"<<fortime23<<"}]"<<'\n';
  MyFile <<"}"<<"\n";

  // Close the file
  MyFile.close();



return 0;
}
```

**Loginfo.json**

```
{
"country" : [{"Mongolia":3},{"Bangladesh":39},{"Singapore":9},{"UnitedStates":75},{"Russia":45},{"Brazil":16},{"Netherlands":6},{"Norway":12},{"Nepal":11}],
"operatingSys" : [{"Ubuntu":24},{"Windows":60}],
"WebBrowser" : [{"Chrome":99},{"Firefox":42}],
"Timeuser" : [{"time00":21},{"time01":39},{"time02":6},{"time03":9},{"time04":21},{"time05":9},{"time06":6},{"time07":9},{"time08":6},{"time09":12},{"time10":18},{"time11":18},
{"time12":15},{"time13":9},{"time14":24},{"time15":0},{"time16":0},{"time17":0},{"time18":0},{"time19":0},{"time20":0},{"time21":0},{"time22":0},{"time23":0}]
}
```

**Thank you**