Lab Assignment 11 - Dashboards

Problem Statement 1:-

In teams of 2 or 3 or as an individual, create a dashboard keeping in mind the key pointers discussed in class.Take a dataset snippet from UCI repository or Kaggle for the same..

Team Members :-

Shiv Chevli – (190420107008)

Vishal Parmar – (190420107039)

Yuvraj Chauhan – (190420107007)

Algorithm/ Code :

// Index.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>Deashbord</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <nav>
    <h1 class="nav-heading">Country List</h1>
    <input type="text" id="find_country">
    <ul class="nav-country-list" id="countryList">
    </ul>
  </nav>
  <main>
    <section class="main_section">
      <section class="main-chart">
        <h1 class="main-h1-heading">Covid - 19 Case study </h1>
        <div id="geo-chart"></div>
        <div class="info-card-container">
          <div class="info-card">
            <h3>Max Case Registered</h3>
```

```
            <div id="max_case">
              2500
            </div>
          </div>
          <div class="info-card">
            <h3>Max Case Registration Date</h3>
            <div id="max_case_date">
              12-03-2022
            </div>
          </div>
          <div class="info-card">
            <h3>Average Cases</h3>
            <div id="avg_case">
              12-03-2022
            </div>
          </div>
        </div>
      </section>
      <section class="side-bar">
        <h1 class="main-heading">Country Name : <span id="country_name"></span></h1>
        <div class="chart-div">
          <div class="chart-container" id="activate-case"></div>
          <div class="chart-container" id="confiremed-case"></div>
          <div class="chart-container" id="new-case"></div>
          <div class="chart-container" id="new-recover"></div>
          <div class="chart-container" id="new-death"></div>
          <div class="chart-container" id="dount"></div>
        </div>
      </section>
    </section>
  </main>
</body>
<script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
<script src="https://cdn.jsdelivr.net/npm/apexcharts"></script>
<script src="script.js"></script>
</html>
```

```
// Style.css

* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}
:root {
  --orange: #f07d51;
```

```
    --text-while: #fff;
}
body {
    height: 100vh;
    overflow: hidden;
    display: flex;
    flex-direction: row;
    background: var(--backgound);
}
nav {
    display: inline-block;
    color: var(--text-while);
    width: 300px;
    height: 100vh;
    padding: 30px 30px 10px;
    border: 1px solid black;
    background-color: var(--orange);
    overflow-y: auto;
    overflow-x: hidden;
}
.nav-heading {
    font-weight: 900;
    font-size: 1.5em;
    text-align: center;
    padding: 15px;
    margin: 0px 0 30px;
    border-bottom: 2px solid var(--text-while);
}
.active {
    font-size: 1.2rem;
    border-bottom: 1px solid var(--text-while);
}
#find_country {
    display: block;
    padding: 10px;
    border-radius: 3px;
    font-size: 18px;
    border: none;
}
#find_country:focus {
    border: 1px solid var(--orange);
    outline: 1px solid var(--orange);
}
.nav-country-list {
    font-weight: 600;
    /* list-style: none; */
}
```

```
.nav-country-list li {
   font-size: 1.2rem;
   margin: 5px 20px;
   cursor: pointer;
}
main {
   display: inline-block;
   height: 100vh;
}
.main_section {
   display: flex;
   flex-direction: row;
   flex-wrap: wrap;
   justify-content: space-between;
}
.main_section section {
   height: 100vh;
}
.main-chart {
   width: 60%;
   padding: 20px 30px;
}
.main-h1-heading {
   font-size: 3em;
   border-bottom: 1px solid #000;
   margin-bottom: 20px;
}
.main-heading {
   font-style: 2em;
   border-bottom: 2px solid #000;
   margin: 20px 30px !important;
   padding: 10px;
}
.chart-div {
   margin: 0px auto;
   text-align: center;
}
.chart-div .chart-container {
   width: 100%;
   display: inline-block;
}
.side-bar {
   width: 450px;
   margin-right: 30px;
   padding: 10px 20px;
}
.info-card-container {
```

```
    margin-top: 50px;
    display: flex;
    justify-content: space-between;
}
.info-card {
    display: inline-block;
    padding: 10px;
    margin: 0 10px;
    border-radius: 5px;
    border: 1px solid black;
    margin-bottom: 100px;
    vertical-align: top;
    background-color: var(--box-color);
}
.info-card h3 {
    border-bottom: 1px solid #000;
    padding: 5px;
}
.info-card div {
    font-size: 20px;
    height: 30px;
    display: flex;
    justify-content: center;
    align-items: center;
}
#geo-chart {
    width: 80%;
    height: 400px;
    margin: 0 auto;
}
```

// script.js

```
function csvToArray(str, delimiter = ",") {
    const headers = str.slice(0, str.indexOf("\n")).split(delimiter);
    const rows = str.slice(str.indexOf("\n") + 1).split("\n");
    const arr = rows.map(function (row) {
        const values = row.split(delimiter);
        const el = headers.reduce(function (object, header, index) {
            object[header] = values[index];
            return object;
        }, {});
        return el;
    });
    // return the array
    return arr;
}
```

```
var country = Array();
var geoData = Array();
var OverView = {};
var mainData;
fetch('Data/full_grouped.csv').then(respons => respons.text())
   .then(data => {
      mainData = csvToArray(data);
      console.log(mainData);
      let back = mainData[0]["Country/Region"];
      country.push(mainData[0]["Country/Region"])
      for (i = 1; i < mainData.length; i++) {
         if (back == mainData[i]["Country/Region"]) {
            break;
         }
         country.push(mainData[i]["Country/Region"])
      }
      OverView = overViewData(mainData);
      console.log("OverView");
      console.log(OverView);
      geoData.push(['Country', 'Active Case On Average']);
      function FindataFromGeo() {
         for (const key in OverView) {
            geoData.push([key, (OverView[key].activte_case / mainData.length)]);
         }
      }
      FindataFromGeo();
      setCountry(country);
      updateChart("India");
      document.querySelector("#country_name").innerHTML = "India";
      google.charts.load('current', {
         'packages': ['geochart'],
      });
      google.charts.setOnLoadCallback(drawRegionsMap);
      function drawRegionsMap() {
         var data = google.visualization.arrayToDataTable(geoData);
         var options = {
            tooltip: { trigger: 'selection' },
            colorAxis: { colors: ["#10af0d", "#fff", "#fff", "#950101"] },
         };
         var chart = new google.visualization.GeoChart(document.getElementById('geo-chart'));
         chart.draw(data, options);
         google.visualization.events.addListener(chart, 'select', myalret);
         async function myalret(e) {
            var selection = chart.getSelection();
            if (selection[0].row != null) {
               updateChart(geoData[selection[0].row][0]);
               setTimeout(() => {
```

```
                let count = 0;
                let tm = undefined;
                while (true) {
                  tm = document.querySelector(".google-visualization-tooltip");
                  if (tm != undefined || count == 500) {
                    break;
                  }
                  count++;
                }
                tm = tm.children;
                document.querySelector("#country_name").innerHTML = tm[1].children[0].innerHTML;
              },
                1000);
            }
          }
        }
        document.querySelectorAll("#countryList li").forEach(li => {
          li.onclick = function () {
            console.log(this.dataset.name);
            document.querySelector("#country_name").innerHTML = this.dataset.name;
            updateChart(this.dataset.name);
            if (document.querySelector(".active") != null) {
              document.querySelector(".active").classList.remove("active");
            }
            this.classList.add("active");
            document.querySelectorAll("#countryList li").forEach(li => {
              li.style.display = "list-item";
            })
            document.querySelector("#find_country").value = "";
          }
        })
      })
      .catch(error => {
        console.log("Error : ");
        console.log(error);
      })

    function setCountry(list) {
      let listUL = document.querySelector("#countryList");
      listUL.innerHTML = "";
      for (i = 0; i < list.length; i++) {
        let li = document.createElement("li");
        li.innerHTML = list[i];
        li.dataset.name = list[i];
        listUL.appendChild(li);
      }
    }
```

```
function extrextData(conName) {
   let max_case = 0;
   let max_case_date = "";
   let total = 0;
   let actvate_case = Array();
   let new_case = Array();
   let confirm = Array();
   let deaths = Array();
   let new_recover = Array();
   let new_death = Array();
   let date = Array();
   for (i = 0; i < mainData.length; i++) {
      if (mainData[i]["Country/Region"] == conName) {
         if (max_case < parseInt(mainData[i]["Confirmed"])) {
            max_case = parseInt(mainData[i]["Confirmed"]);
            max_case_date = mainData[i]["Date"];
         }
         total += parseInt(mainData[i]["Confirmed"]);
         date.push(mainData[i]["Date"]);
         confirm.push(mainData[i]["Confirmed"]);
         actvate_case.push(mainData[i]["Active"]);
         deaths.push(mainData[i]["Deaths"]);
         new_case.push(mainData[i]["New cases"]);
         new_death.push(mainData[i]["New deaths"]);
         new_recover.push(mainData[i]["New recovered"]);
      }
   }
   let obj = {
      "date": date,
      "confirm": confirm,
      "deaths": deaths,
      "actvate_case": actvate_case,
      "new_case": new_case,
      "new_death": new_death,
      "new_recover": new_recover,
      "max_case": max_case,
      "max_case_date": max_case_date,
      "total": total,
   }
   return obj;
}

function updateChart(conName) {
```

```
let obj = extrextData(conName);
document.querySelector(".chart-div").innerHTML = `
   <div class="chart-container" id="activate-case"></div>
   <div class="chart-container" id="confiremed-case"></div>
   <div class="chart-container" id="new-case"></div>
   <div class="chart-container" id="new-death"></div>
   <div class="chart-container" id="new-recover"></div>
`;
document.querySelector("#max_case").innerHTML = obj.max_case;
document.querySelector("#max_case_date").innerHTML = obj.max_case_date;
document.querySelector("#avg_case").innerHTML = parseInt(obj.total / obj.date.length);


window.Apex = {
   chart: {
      width: 400,
      height: 150,
      type: "bar",
   },
   dataLabels: {
      enabled: false
   },
   xaxis: {
      type: "datetime",
      labels: {
         datetimeFormatter: {
            year: 'yyyy',
            month: 'MMM \'yy',
            day: 'dd MMM',
            hour: 'HH:mm'
         }
      },
      categories: obj.date,
   },
   legend: {
      horizontalAlign: "center",
   },
   tooltip: {
      enabled: true,
      intersect: false,
      inverseOrder: false,
      custom: undefined,
      fillSeriesColor: true,
      theme: false,
      style: {
         fontSize: '12px',
         fontFamily: undefined
```

```
        },
        onDatasetHover: {
          highlightDataSeries: true,
        },
        marker: {
          show: true,
        },
        items: {
          display: "flex",
        },
        fixed: {
          enabled: true,
          position: 'topright',
          offsetX: 0,
          offsetY: 0,
        },
      }

    }

    // createChart("Confirmed Case", obj.confirm, obj.date, document.querySelector("#confiremed-case"));
    // createChart("Actvate case", obj.actvate_case, obj.date, document.querySelector("#activate-case"));

    createChart("New case", obj.new_case, obj.date, document.querySelector("#new-case"));
    createChart("New Recoverd", obj.new_recover, obj.date, document.querySelector("#new-recover"));
    createChart("New Death", obj.new_death, obj.date, document.querySelector("#new-death"));

}

function createChart(lable, ydata, xdata, elment) {
    var options = {
      chart: {
        id: 'line-1',
        group: 'social',
        type: 'line',
      },
      series: [
        {
          name: lable,
          data: ydata
        },
      ],
      stroke: {
        width: 1
```

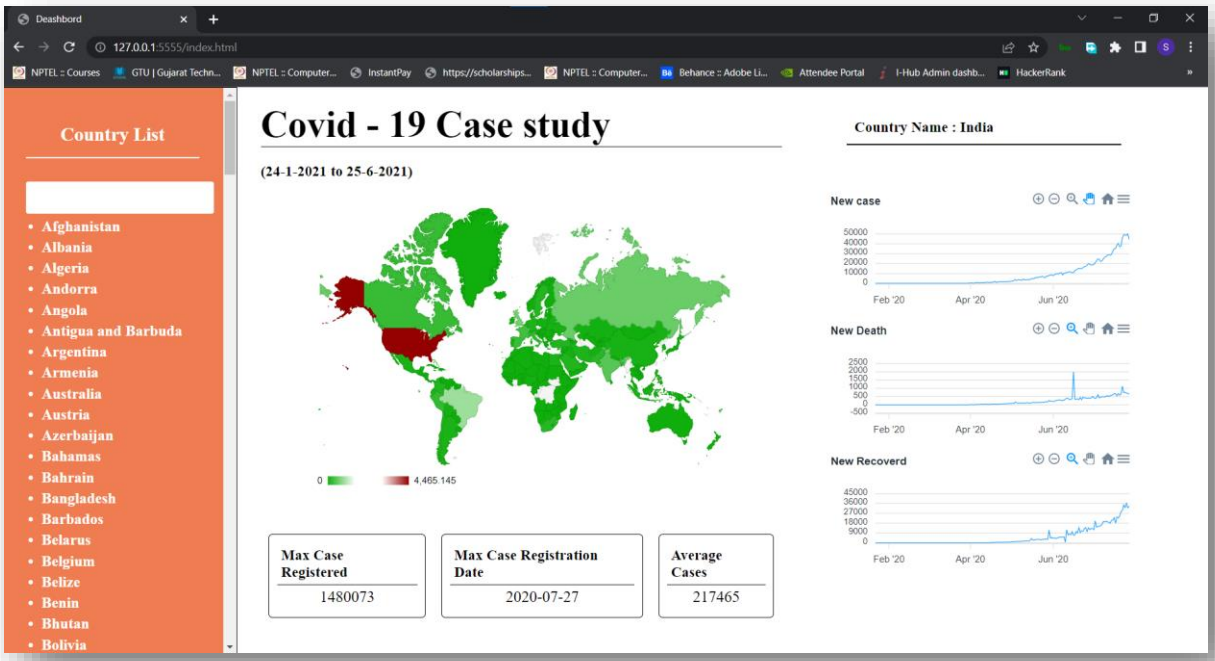
```
      },
      yaxis: {
        labels: {
          minWidth: 40
        }
      },
      title: {
        text: lable,
        align: 'left',
        margin: 10,
        offsetX: 0,
        offsetY: 0,
        floating: false,
        style: {
          fontSize: '14px',
          fontWeight: 'bold',
          fontFamily: undefined,
          color: '#263238'
        },
      }
    };
    var chart = new ApexCharts(elment, options);
    chart.render();
}
document.querySelector("#find_country").addEventListener("keyup", function () {

    let val = this.value.toLowerCase();
    console.log(val);
    if (val != "") {
      document.querySelectorAll("#countryList li").forEach(li => {
        let temp = li.innerHTML.toLowerCase();
        if (temp.indexOf(val) > -1) {
          li.style.display = "list-item;";
        }
        else {
          li.style.display = "none";
        }
      })
    }
    else {
      document.querySelectorAll("#countryList li").forEach(li => {
        li.style.display = "list-item";
      })
    }
})
function overViewData(mainData) {
    let OverView = {};
```

```
  for (i = 0; i < mainData.length; i++) {
    if (OverView[mainData[i]["Country/Region"]] == undefined) {
      OverView[mainData[i]["Country/Region"]] = {
        "activte_case": 0,
        "confirmed_case": 0,
        "death_case": 0,
        "recovered": 0,
      };
    }
    else {
      OverView[mainData[i]["Country/Region"]].confirmed_case =
parseInt(OverView[mainData[i]["Country/Region"]].confirmed_case) +
parseInt(mainData[i]["Confirmed"]);
      OverView[mainData[i]["Country/Region"]].activte_case =
parseInt(OverView[mainData[i]["Country/Region"]].activte_case) + parseInt(mainData[i]["Active"]);
      OverView[mainData[i]["Country/Region"]].death_case =
parseInt(OverView[mainData[i]["Country/Region"]].death_case) + parseInt(mainData[i]["Deaths"]);
      OverView[mainData[i]["Country/Region"]].recovered =
parseInt(OverView[mainData[i]["Country/Region"]].recovered) +
parseInt(mainData[i]["Recovered"]);
    }
  }
  return OverView;
}
```

This Project is live on below link :
https://shivchevli.github.io/covid_19_dashboard/

Output :

// Insight of country Afghanistan



// Insight of country Angola