SOFTWARE MODELING DEVELOPMENT WITH UML PROJECT REPORT

Under the guidance of

- Prof. Dr. Atef Bader

Bangalore Raghunath, Abhay Nagaraj

A20382006

Hongal, Vaibhav

A20378220

Katiyar, Karan

A20381532

ILLINOIS INSTITUTE OF TECHNOLOGY Chicago

- Objective: To design and implement an object oriented charting Dashboard for data analytics and visualization of Demographics datasets. This helps the user in drawing vital insights.
- The following are the forms of plots that we are planning to achieve.
  - Pie chart
  - Bar chart
  - Line chart
  - Stacked chart
  - Pivot chart

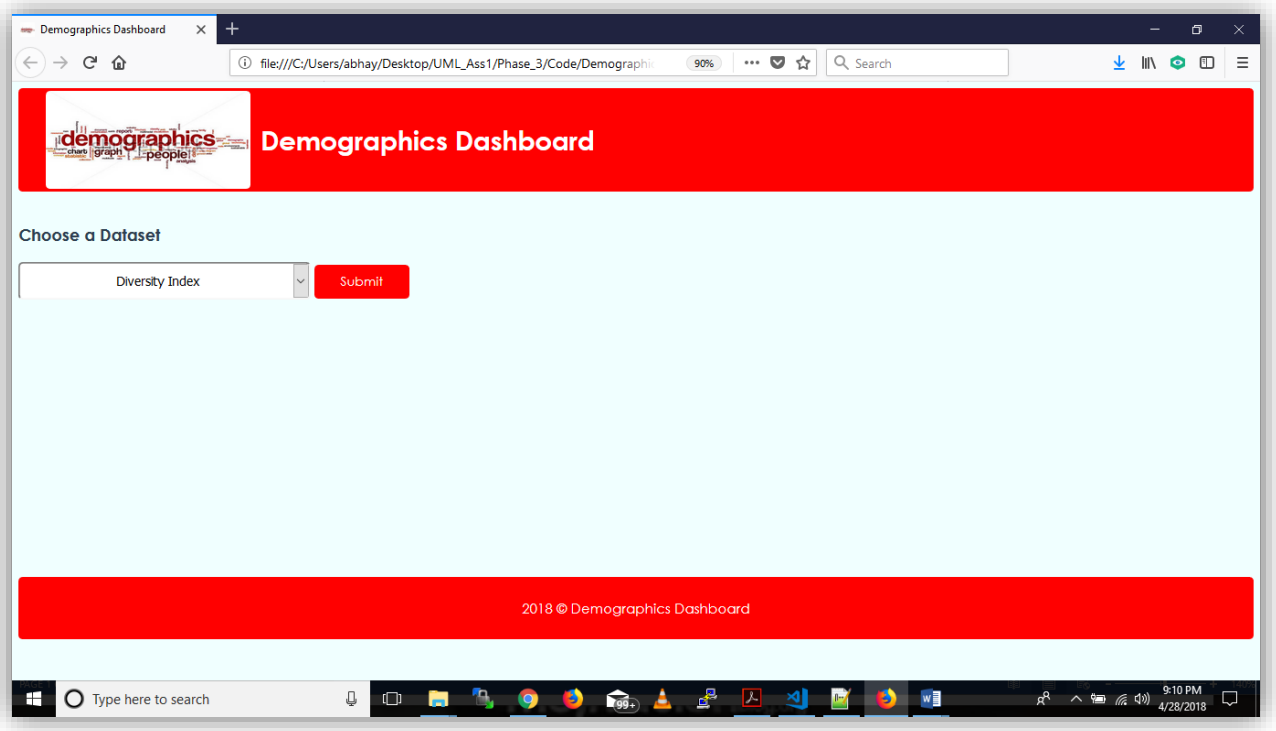
Project Features:

- *User will be able to load the data into data frame object using Dataframe.js*
- *File based client sided framework used wherein no exclusive server required.*
- *Chrome and Firefox have inbuilt ES6 compilers and servers like v8 which are operate the hood.*
- *Plot the data as chart using Chart.js*
- *Information must be displayed in a table*
- *Dashboard must work in the latest version of Chrome and Firefox*

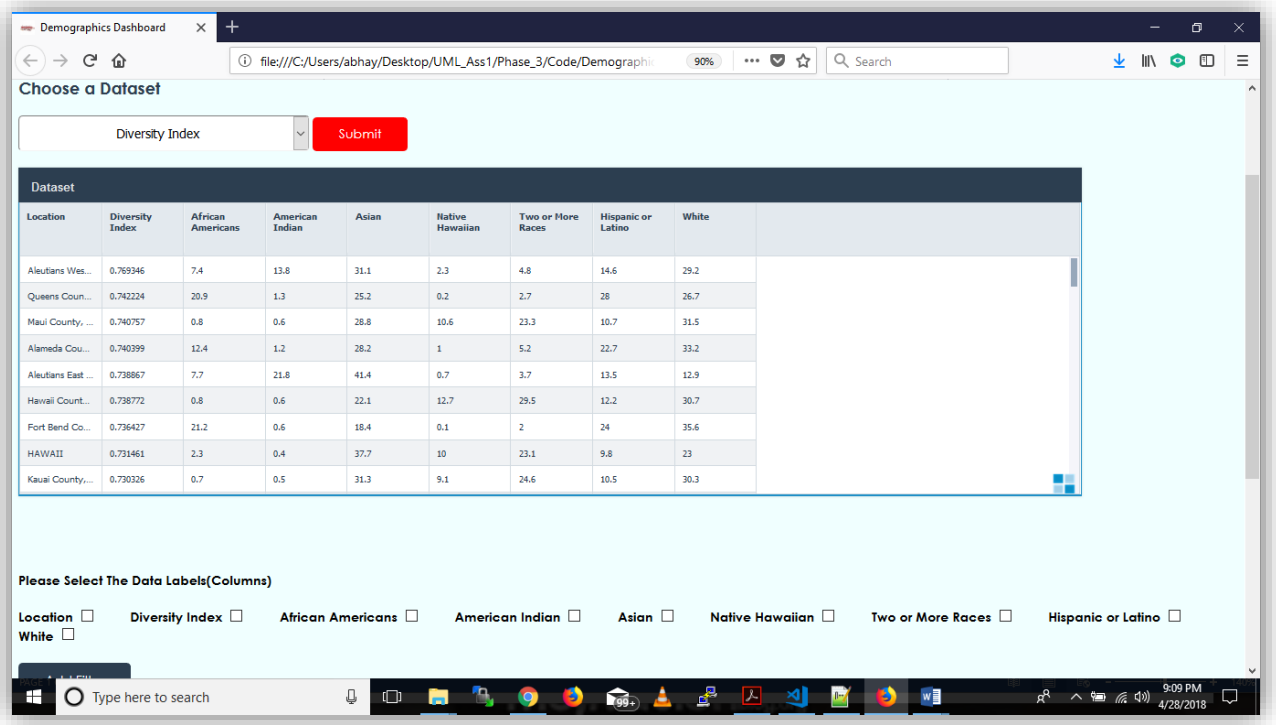
List of use cases:

1. Load Dataset- Can load the different datasets
2. Export Dataset- Can export either as CSV FILE or export as JSON FILE
3. Select Dataset- Select the desired dataset
4. Filter Dataset- Can be able to filter the desired datasets by rows, columns or queries
5. View Chart- Can view charts in various forms like bar, line, pie, stacked and pivot charts
6. View Dataset- various options like race (ethnicity), income, job, sexuality and zip codes in the USA

The launch Screen is as follows, which opens on double clicking the html file in the code directory of the unzipped project.



Select a dataset from the dropdown list and hit Submit to see the tabulated data as follows.

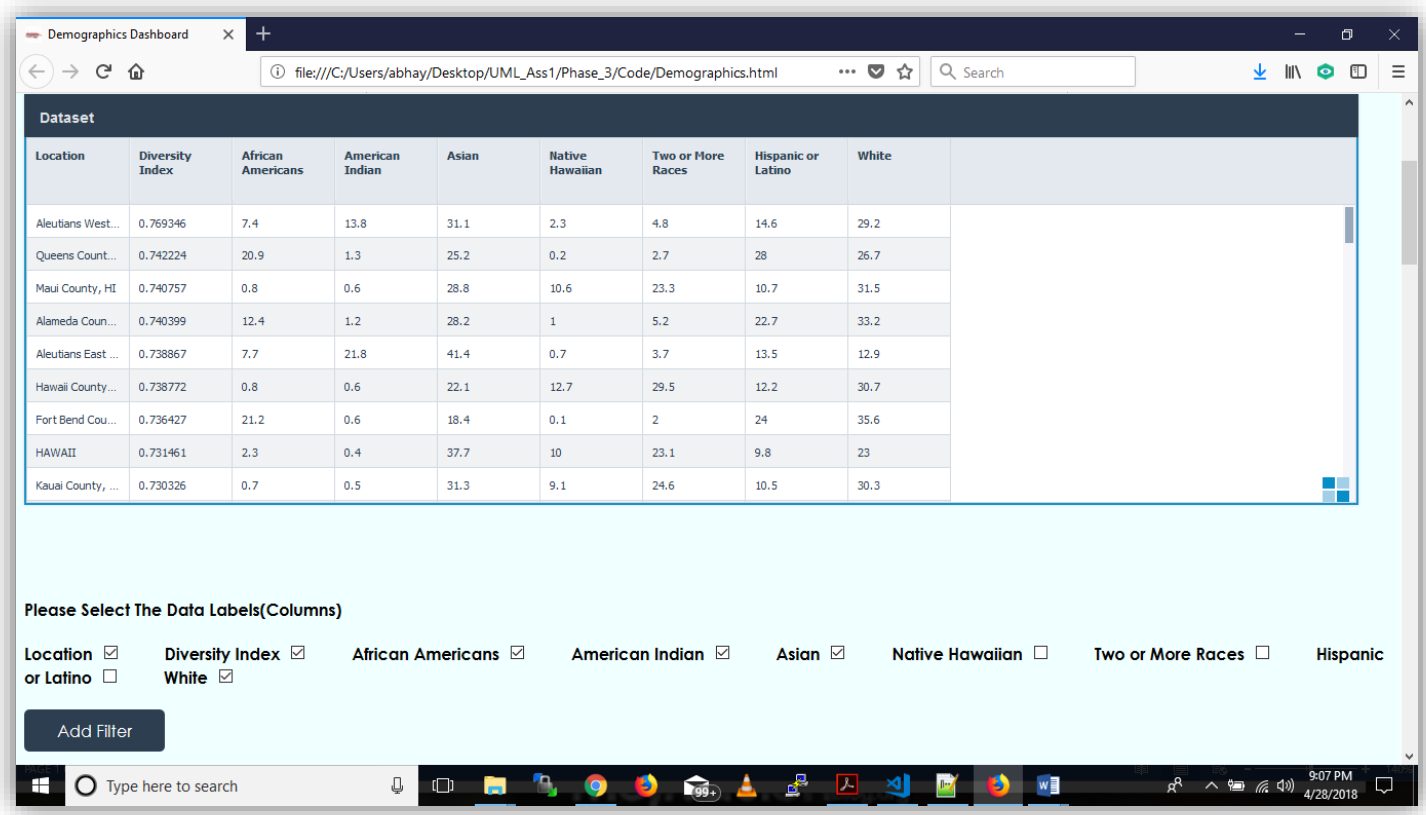


Location	Diversity Index	African Americans	American Indian	Asian	Native Hawaiian	Two or More Races	Hispanic or Latino	White
Aleutians Wes...	0.769346	7.4	13.8	31.1	2.3	4.8	14.6	29.2
Queens Coun...	0.742224	20.9	1.3	25.2	0.2	2.7	28	26.7
Maui County, ...	0.740757	0.8	0.6	28.8	10.6	23.3	10.7	31.5
Alameda Cou...	0.740399	12.4	1.2	28.2	1	5.2	22.7	33.2
Aleutians East ...	0.738867	7.7	21.8	41.4	0.7	3.7	13.5	12.9
Hawaii Count...	0.738772	0.8	0.6	22.1	12.7	29.5	12.2	30.7
Fort Bend Co...	0.736427	21.2	0.6	18.4	0.1	2	24	35.6
HAWAII	0.731461	2.3	0.4	37.7	10	23.1	9.8	23
Kauai County,...	0.730326	0.7	0.5	31.3	9.1	24.6	10.5	30.3

All the columns in the dataset are populated as check boxes. Select the columns whose date you'd want to be used for plotting the charts further.

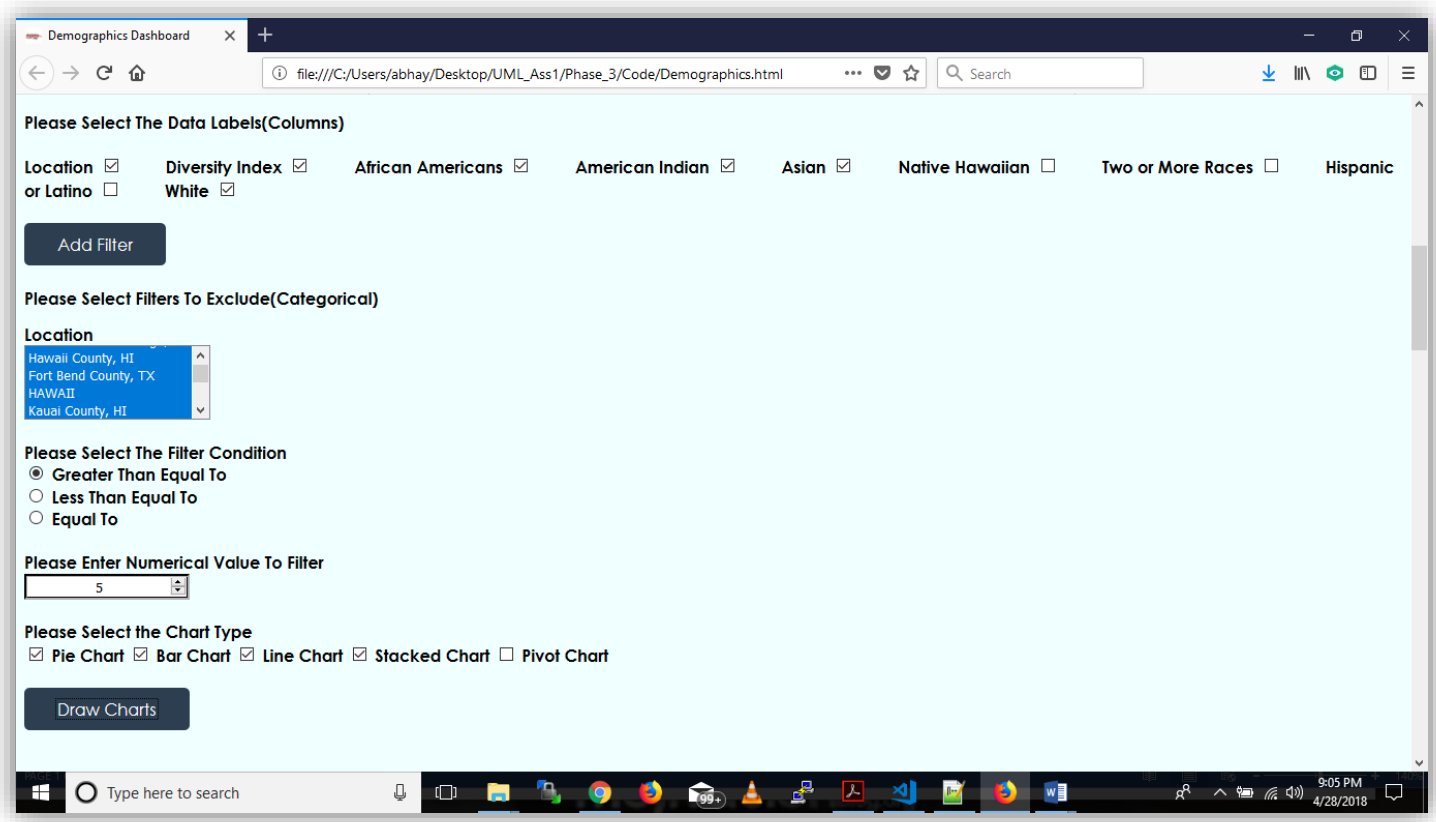
Checked Location, Diversity Index, African American, American Indian, Asian and White columns.

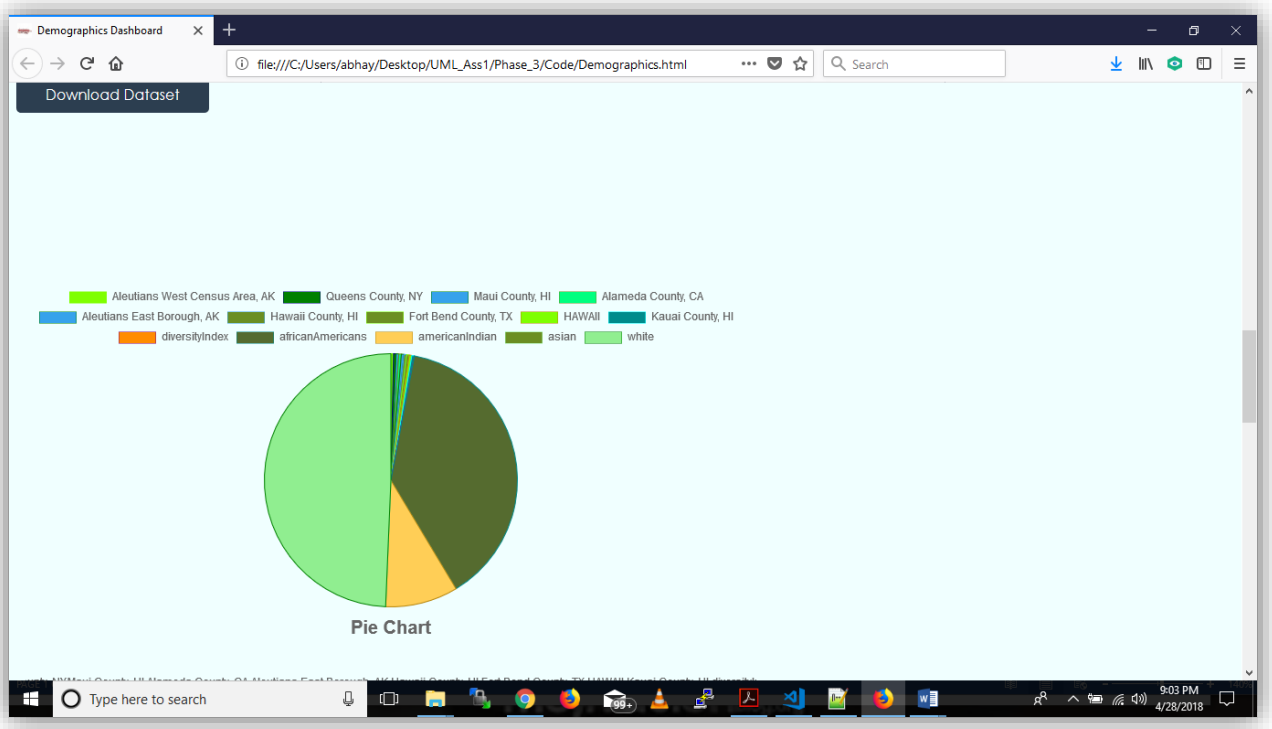
Hit Add filter to see further options.

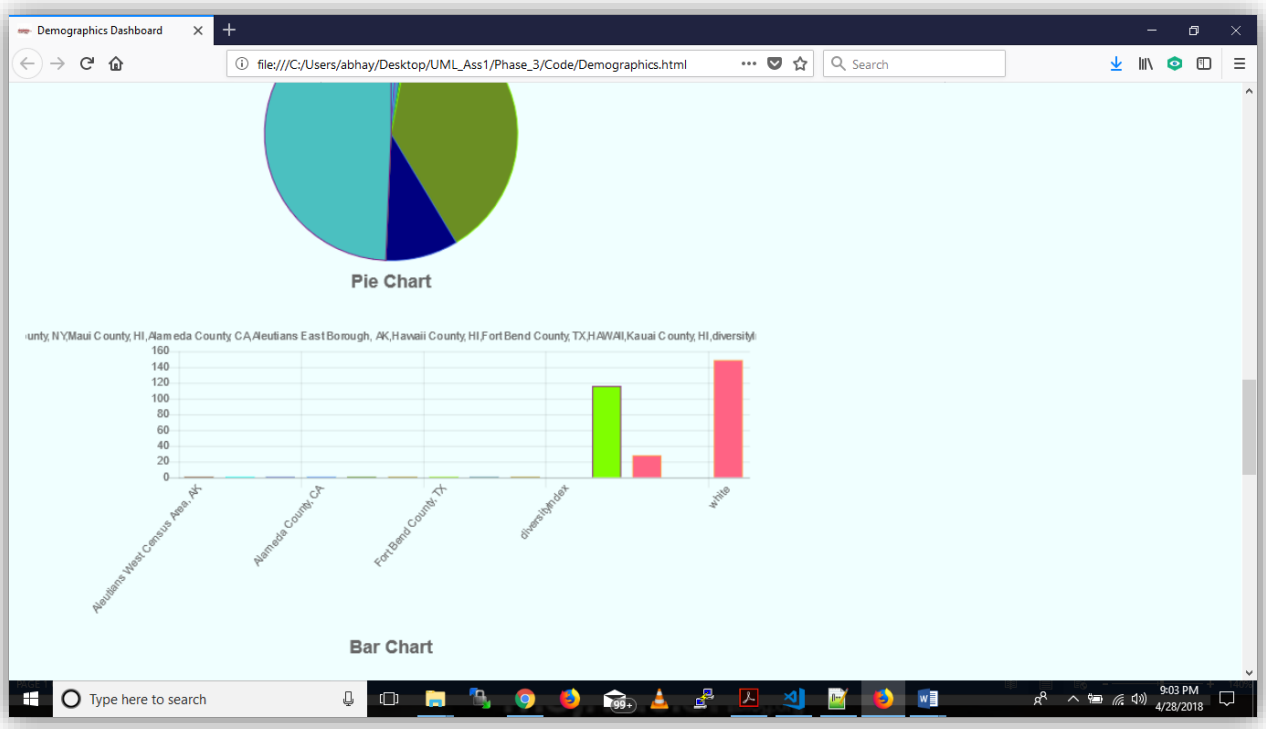


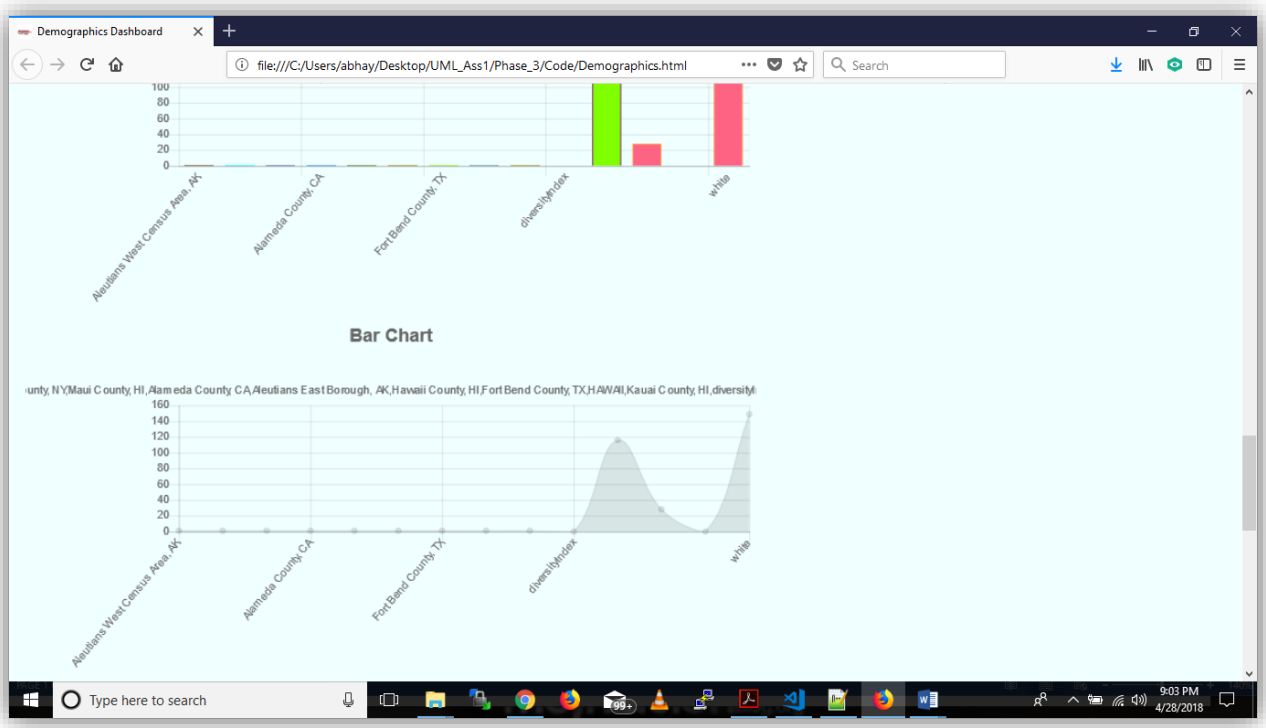
Select filters and add an inequality and enter the numerical value to complete the inequality.

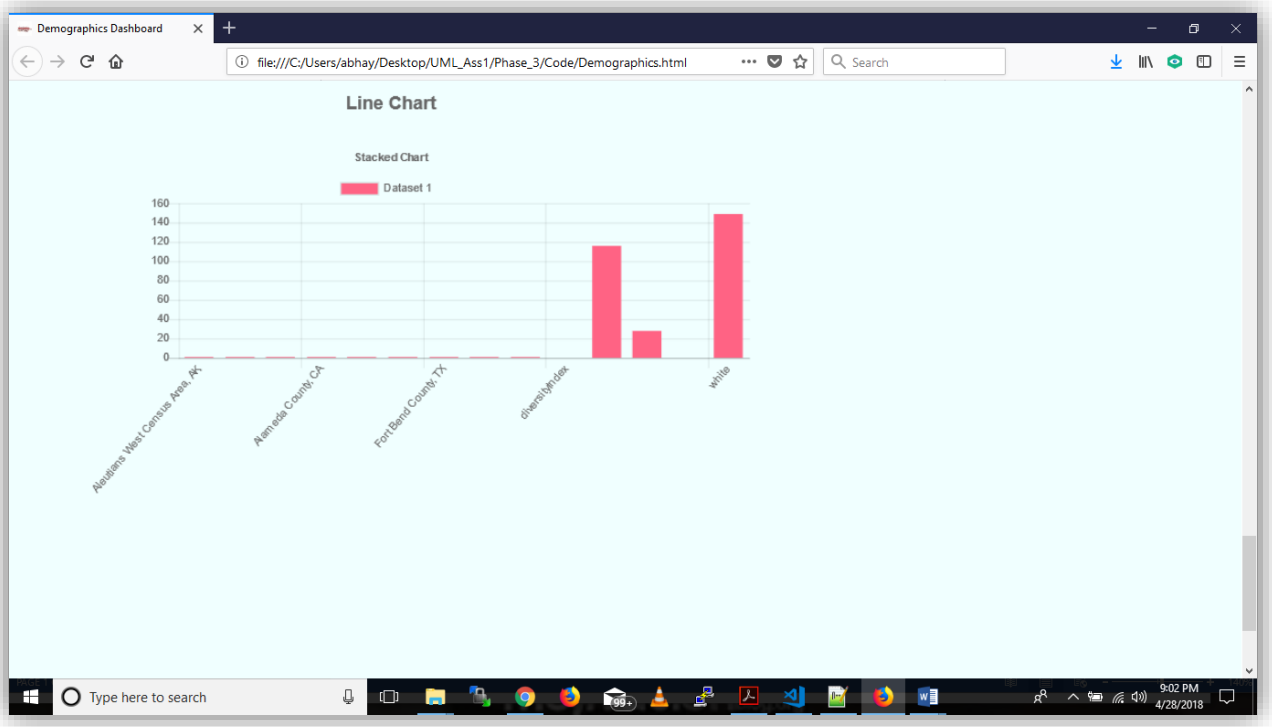
Select the charts you'd want the data to be visualized using. Selected Pie, Bar, Line and Stacked Charts and hit Draw Charts to see the below visuals.



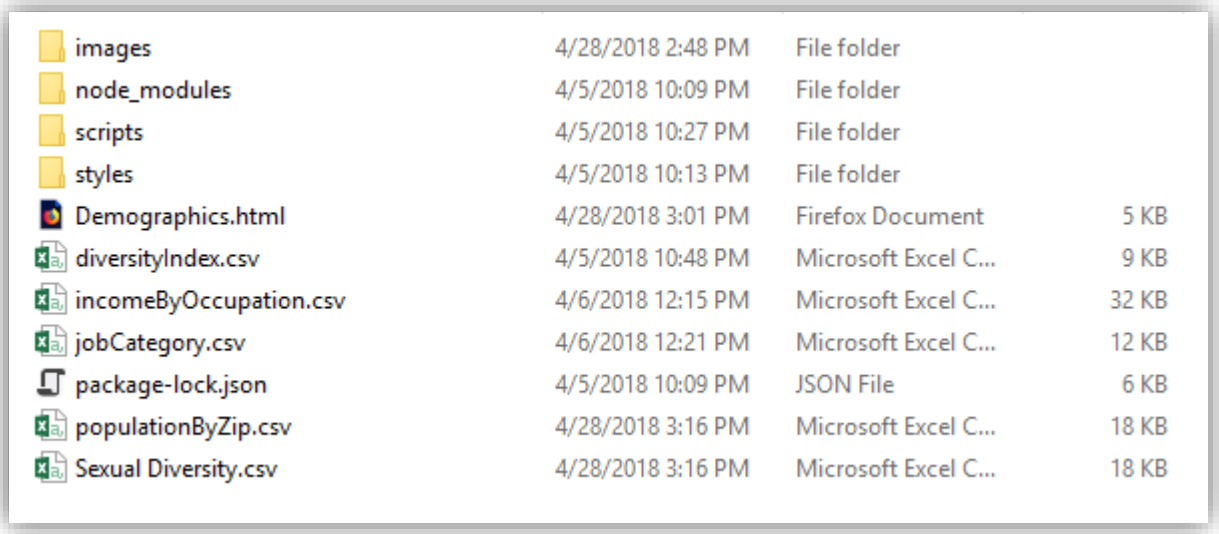

Demographics Dashboard
file:///C:/Users/abhay/Desktop/UML_Ass1/Phase_3/Code/Demographics.html
Search
Download Dataset
Aleutians West Census Area, AK
Queens County, NY
Maui County, HI
Alameda County, CA
Aleutians East Borough, AK
Hawaii County, HI
Fort Bend County, TX
HAWAII
Kauai County, HI
diversityIndex
africanAmericans
americanIndian
asian
white
Pie Chart
Type here to search
9:03 PM
4/28/2018


Demographics Dashboard
file:///C:/Users/abhay/Desktop/UML_Ass1/Phase_3/Code/Demographics.html
Search
Pie Chart
unty, NY,Maui County, HI,Alameda County, CA,Aleutians East Borough, AK,Hawaii County, HI,Fort Bend County, TX,HAWAII,Kauai County, HI,diversityI
160
140
120
100
80
60
40
20
0
Aleutians West Census Area, AK
Alameda County, CA
Fort Bend County, TX
diversityIndex
white
Bar Chart
Type here to search
9:03 PM
4/28/2018


Demographics Dashboard
file:///C:/Users/abhay/Desktop/UML_Ass1/Phase_3/Code/Demographics.html
Search
100
80
60
40
20
0
Aleutians West Census Area, AK
Alameda County, CA
Fort Bend County, TX
diversityIndex
white
Bar Chart
unty, NY,Maui County, HI,Alameda County, CA,Aleutians East Borough, AK,Hawaii County, HI,Fort Bend County, TX,HAWAII,Kauai County, HI,diversityI
160
140
120
100
80
60
40
20
0
Aleutians West Census Area, AK
Alameda County, CA
Fort Bend County, TX
diversityIndex
white
Type here to search
9:03 PM
4/28/2018



The folder schema of the project is as follows



images	4/28/2018 2:48 PM	File folder	
node_modules	4/5/2018 10:09 PM	File folder	
scripts	4/5/2018 10:27 PM	File folder	
styles	4/5/2018 10:13 PM	File folder	
Demographics.html	4/28/2018 3:01 PM	Firefox Document	5 KB
diversityIndex.csv	4/5/2018 10:48 PM	Microsoft Excel C...	9 KB
incomeByOccupation.csv	4/6/2018 12:15 PM	Microsoft Excel C...	32 KB
jobCategory.csv	4/6/2018 12:21 PM	Microsoft Excel C...	12 KB
package-lock.json	4/5/2018 10:09 PM	JSON File	6 KB
populationByZip.csv	4/28/2018 3:16 PM	Microsoft Excel C...	18 KB
Sexual Diversity.csv	4/28/2018 3:16 PM	Microsoft Excel C...	18 KB

The Model View Controller architecture is implemented as follows

Model being the ModeUtility.js

View being the Demographics.js

Controller being the ControllerUtility.js

constants	4/5/2018 10:13 PM	File folder	
ControllerUtility.js	4/6/2018 12:53 PM	JS File	8 KB
Dataset.js	4/28/2018 8:38 PM	JS File	1 KB
Demographics.js	4/28/2018 2:46 PM	JS File	22 KB
ModelUtility.js	4/6/2018 12:25 PM	JS File	2 KB

Code snippets 1

```
class ChartFactory{
    createChart(chartType){
        let chart = null;
        if(chartType == "pie"){
            chart = new PieChart();
        } else if(chartType == "bar"){
            chart = new BarChart();
        } else if(chartType == "line"){
            chart = new LineChart();
        } else if(chartType == "stacked"){
            chart = new StackedChart();
        } else if(chartType == "pivot"){
            chart = new PivotChart();
        }
        return chart;
    }
}

class ChartStore{
    constructor(chartFactory){
        this.chartFactory = new ChartFactory();
    }

    orderChart(chartType){
        var chart = this.chartFactory.createChart(chartType);
        return chart;
    }
}


class BaseChartOperation{

    constructor(chart, name,type, ctx, labels = [], chartData = [], backgroundColor = '', borderColor = '', options = {}) {
        if(chart != undefined || chart != null){
```

Code snippet 2

```
class ChartConfig extends BaseChartOperation{

    constructor(chart) {
        super(chart);
    }
    setLabelAndData(labels, data){
      this.labels = labels;
      this.chartData = data;
    }

    applyBackgroundColor(chartConfig){

    }

    applyBorderColor(chartConfig){

    }

    randomScalingFactor() {
        return (Math.random() > 0.5 ? 1.0 : -1.0) * Math.round(Math.random() * 100);
    }
}

class ChartDecorator extends ChartConfig {

    applyBorderColor(chartConfig){
        chartConfig.borderColor = chartConfig.getborderColor(chartConfig.chartData.length);
    }
}
```

Code snippet 3

```
class Dataset{
    constructor(){
        this.dataFrame = dfjs.DataFrame;
    }

    getDataFrame(){
        return this.dataFrame;
    }

    import_data(){
        var vals = document.getElementById("datasets");
        currentDataSet = vals.options[vals.selectedIndex].value;
        this.dataFrame.fromCSV(currentDataSet).then(df =>
        {
            var data = df.toJSON('SAT.json');
            export_data(data);
        });
    }


    export_data(dataset){
        var blob= new Blob([data],{ type:"text/ApplicationJson;charset:utf-8" });
        dataframeSet = dataset;
        createTableFromJSON();
    }
}
```

Code snippet 4

```
/*
1. Function to import the datasets
*/
function importData(){
    let DataFrame = dfjs.DataFrame;
    var vals = document.getElementById("datasets");
    currentDataSet = vals.options[vals.selectedIndex].value;
    DataFrame.fromCSV(currentDataSet).then(df =>
    {
        data = df.toJSON('SAT.json');
        loadData(data);
    });
}

/*
2. Function to load the datasets on the website for further use
*/
function loadData(dataset) {
    var blob = new Blob([data],{ type:"text/ApplicationJson;charset:utf-8" });
    dataframeSet = dataset;
    createTableFromJSON();
}

/*
3. Function to prepare the table with styles,
   column headers and column data for displaying the data
*/
function createTableFromJSON() {
    var colData = [];
    var div = document.getElementById('table');
```

Code snippet 5

```
/*
9. Function to create object for the Diversity Index dataset
*/
function createDiversityIndexObjects(colHeaderValues) {
    var json = getParsedJson();
    for(var i = 0 ; i < json.length; i++){
        var record = json[i];
        let diversityIndexObject = new DiversityIndexObject(record[colHeaderValues[0]],
                                                            record[colHeaderValues[1]],
                                                            record[colHeaderValues[2]],
                                                            record[colHeaderValues[3]],
                                                            record[colHeaderValues[4]],
                                                            record[colHeaderValues[5]],
                                                            record[colHeaderValues[6]],
                                                            record[colHeaderValues[7]],
                                                            record[colHeaderValues[8]]);

        objArr.push(diversityIndexObject);
    }
}

/*
10. Function to create object for the Income by occupation dataset
*/
function createIncomeOccupationObjects(colHeaderValues) {
    var json = getParsedJson();
    for(var i = 0 ; i < json.length; i++){
        var record = json[i];
        let incomeOccupationObject = new IncomeOccupationObject(record[colHeaderValues[0]],
                                                                record[colHeaderValues[1]],
                                                                record[colHeaderValues[2]],
                                                                record[colHeaderValues[3]],
                                                                record[colHeaderValues[4]],
                                                                record[colHeaderValues[5]]
```

Code snippet 6

```
/*
20. Function to Create Row Filters
*/
function createRowFilters() {
    var divContainer = document.getElementById('categorical-filter-checkbox');
    divContainer.innerHTML = "";
    document.getElementById("filter-row").hidden=false;
    var checkedCategorisedColumnsSet = getCheckedCategorisedColumns();
    for (const value of checkedCategorisedColumnsSet) {
        var selectDiv = document.createElement("div");
        selectDiv.style.marginTop = "14px";
        selectDiv.style.marginRight = "24px";

        var select = document.createElement("select");
        select.id = value;
        select.multiple = true;
        select.style.width = "180px";
        select.style.overflowX = "auto";

        var selectLabel = document.createElement('label')
        selectLabel.htmlFor = "id";
        selectLabel.appendChild(document.createTextNode(value));

        var categoryValueSet = getAllValuesForCategory(value);
        for (const value of categoryValueSet) {
            var option = document.createElement("option");
            option.value = value;
            option.selected ="";
            option.innerHTML = value;
            select.add(option);
        }
        selectLabel.append(document.createElement("br"));
        selectLabel.append(select);
        selectDiv.appendChild(selectLabel);
```

Code snippet 7

```
/*
28. Function to get all filtered conditions
*/
function getAllFilteredConditions() {
    let filteredItem = [];
    let selectedCatColumnValueMap = new Map();
    let selectedNumColumnValueMap = new Map();
    let numericalCheckBox = new Set();
    var filter = document.getElementById('categorical-filter-checkbox');
    for(var i = 0; i < filter.children.length ; i++){
        var childElement = filter.children[i].firstElementChild.childNodes;
        var columnName = convertToCamelCase(childElement[0].data," ");
        var multiSelectDropDowns = childElement[2].options;
        let selectedCatValueSet = new Set();
        for(var j = 0 ; j< multiSelectDropDowns.length; j++){
            if(multiSelectDropDowns[j].selected){
                selectedCatValueSet.add(multiSelectDropDowns[j].value);
            }
        }
        if(selectedCatValueSet.size > 0){
            selectedCatColumnValueMap.set(columnName,selectedCatValueSet);
        }
    }
    filteredItem.push(selectedCatColumnValueMap);
    var numericalFilters = getAllNumericalColumns();
    for (const value of numericalFilters) {
        if(isColumnSelected(value)){
            numericalCheckBox.add(convertToCamelCase(value," "));
        }
    }
    if(numericalCheckBox.size > 0){
```

Code snippet 8

```
/*
38. Function to export the filtered data in CSV format
*/
function saveCSV(objArrayy) {
    var array = typeof objArrayy != 'object' ? JSON.parse(objArrayy) : objArrayy;
    var str = '';
    for (var i = 0; i < array.length; i++) {
        var line = '';
        for (var index in array[i]) {
          if(line != '') line += ','

          line += array[i][index];
        }
        str += line + '\r\n';
    }
    var exportFilename = "Dataset.csv";
    if (navigator.appName != 'Microsoft Internet Explorer') {
        var csvData = new Blob([array], {type: 'text/csv;charset=utf-8;'});
        if (navigator.msSaveBlob) {
            navigator.msSaveBlob(csvData, exportFilename);
        } else {
            var link = document.createElement('a');
            link.href = window.URL.createObjectURL(csvData);
            link.setAttribute('download', exportFilename);
            document.body.appendChild(link);
            link.click();
            document.body.removeChild(link);
        }
    } else {
        var popup = window.open('','csv','');
        popup.document.body.innerHTML = '<pre>' + str + '</pre>';
    }
}
```

Code snippet 9

```
/*
32. Function to apply preprocessed filters
*/
function preProcessFilter(filterCondition) {
    let isCategorical = 0;
    let isNumerical = 0;
    let columnName;
    let categoryValues;
    let columnValue;
    let operand;
    let colNames = [];
    let resultDataMap = new Map();
    for (let categoryKey of filterCondition[0].keys()){
        isCategorical = 1;
        isNumerical = 0;
        columnName = categoryKey;
        categoryValues = filterCondition[0].get(categoryKey);
        for(let columnVal of categoryValues){
            let count = filterData(columnName, columnVal, isCategorical, isNumerical);
            resultDataMap.set(columnVal, count);
        }
    }
    for(let numericalKey of filterCondition[1].keys())
    {
        isNumerical = 1;
        isCategorical = 0;
        if(numericalKey == "selectedNumericalValues"){
            let numericalVals = filterCondition[1].get(numericalKey);

            for(let val of numericalVals.keys())
            {
                colNames.push(val);
```

Code snippet 10

```
class DiversityIndexObject
{
    constructor(location,diversityIndex,africanAmericans,americanIndian,asianNative,hawaiian,twoOrMoreRaces,hispanicOrLatino
    {
        this.location = location;
        this.diversityIndex = diversityIndex;
        this.africanAmericans = africanAmericans;
        this.americanIndian = americanIndian;
        this.asianNative = asianNative;
        this.hawaiian = hawaiian;
        this.twoOrMoreRaces = twoOrMoreRaces;
        this.hispanicOrLatino = hispanicOrLatino;
        this.white = white;
    }
}

class IncomeOccupationObject{
    constructor(occupation,all_workers,all_weekly,m_workers,m_weekly,f_workers,f_weekly)
    {
        this.occupation = occupation;
        this.all_workers = all_workers;
        this.all_weekly = all_weekly;
        this.m_workers = m_workers;
        this.m_weekly = m_weekly;
        this.f_workers = f_workers;
        this.f_weekly = f_weekly;
    }
}

class JobCategoryObject
{
    constructor(company,year,race,gender,job_category,count)
    {
```

THANK YOU