Reporting

To install in angular simply run the command:

- pm install -g chart.js
- npm install --save ng2-charts

To install jsPDF & jsPDF-AutoTable you follow the following commands:

- npm install -g jspdf jspdf-autotable –save
- After installing add the following to the “**angular.json**” file under the scripts tag
    - [“node_modules/jspdf/dist/jspdf.min.js”, “node_modules/jspdf-autotable/dist/jspdf.plugin.autotable.js”]
- Import the following in **app.modules.ts**

```
import { ChartsModule } from 'ng2-charts';
```

- In each **component.ts** file you create include the following for the charts

```
import html2canvas from 'html2canvas';
import jsPDF from 'jspdf';
import { Label } from 'ng2-charts';
```

- **App-routing-module** should look like this:

```
import { RadarChartComponent } from
'./radar-chart/radar-chart.component';
import { LineChartComponent } from
'./line-chart/line-chart.component';
import { DoughnutChartComponent } from
'./doughnut-chart/doughnut-chart.component';
import { BarChartComponent } from
'./bar-chart/bar-chart.component';
import { BubbleChartComponent } from
'./bubble-chart/bubble-chart.component';
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { PolarChartComponent } from
'./polar-chart/polar-chart.component';


const routes: Routes = [
{ path: '', pathMatch: 'full', redirectTo: 'bar-chart'},
{ path: 'bar-chart', component: BarChartComponent},
{ path: 'bubble-chart', component: BubbleChartComponent},
{ path: 'doughnut-chart', component:
DoughnutChartComponent},
{ path: 'line-chart', component: LineChartComponent},
{ path: 'radar-chart', component: RadarChartComponent},
{ path: 'polar-chart', component: PolarChartComponent}

];

@NgModule({
  imports: [RouterModule.forRoot(routes,
{relativeLinkResolution: 'legacy'})],
  exports: [RouterModule]
})
export class AppRoutingModule { }
```

- **In Style.css:**

```
@import "~bootstrap/dist/css/bootstrap.css"; */

.navbar {
  margin-bottom: 30px;
```

```
}

.navbar-brand {
  margin: 0 auto;
  font-weight: bold;
}

.list-group-item {
  cursor: pointer;
  font-size: 14px;
}

.list-group-item:hover,
.list-group-item.active {
  color: #fff;
  background-color: #007bff;
  border-color: #007bff;
  outline: 0;
}
```

Bar-chart (Shows frequency count of values for different levels of a category)

Html page

```
<div class="chart-wrapper" id="htmlData">
  <canvas baseChart
  [datasets]="barChartData"
  [labels]="barChartLabels"
  [options]="barChartOptions"
  [plugins]="barChartPlugins"
  [legend]="barChartLegend"
```

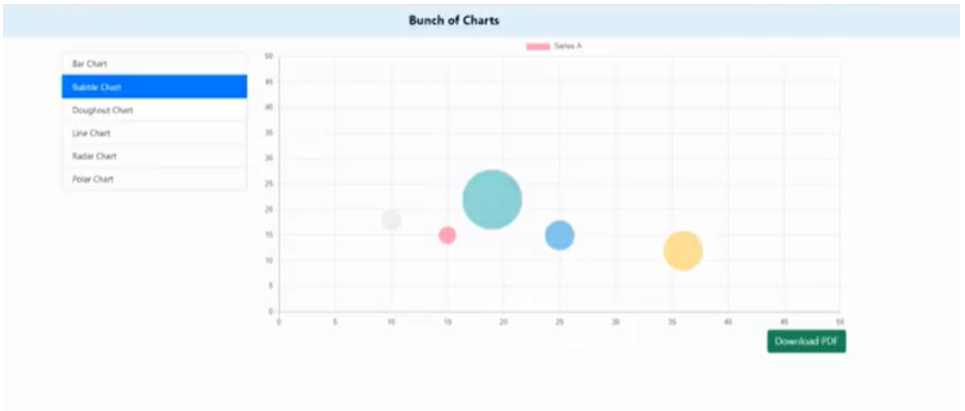
```
  [chartType]="barChartType">
</canvas>
</div>
<div>
  <button style="float: right" class="btn btn-success btn-block"
(click)="openPDF()">Download PDF</button>
</div>
```

.TS File for bar chart

```
import { Component, ElementRef, ViewChild  } from '@angular/core';
import { ChartOptions, ChartDataSets } from 'chart.js';
import { Label } from 'ng2-charts';
import jsPDF from 'jspdf';
import html2canvas from 'html2canvas'

@Component({
  selector: 'app-bar-chart',
  templateUrl: './bar-chart.component.html',
  styleUrls: ['./bar-chart.component.css']
})

export class BarChartComponent {

barChartOptions : ChartOptions = {
  responsive: true
}

barChartLabels: Label[] = ['One', 'Two', 'Three', 'Four', 'Five',
'Six'];
barChartType: Chart.ChartType = 'bar';
barChartLegend = true;
barChartPlugins = [];

barChartData : Chart.ChartDataSets[] = [
  { data: [45, 45, 37, 60, 40, 50], label: 'Series A'},
  { data: [48, 60, 32, 60, 60, 50], label: 'Series B'},
  { data: [70, 90, 33, 60, 50, 50], label: 'Series C'},
]

@ViewChild('htmlData') htmlData:ElementRef | any;

constructor() { }
```

```
// PDF Options
public openPDF(){
  let Data = document.getElementById('htmlData')!;

  html2canvas(Data).then(canvas => {
    let fileWidth = 210;
    let fileHeight = canvas.height * fileWidth / canvas.width;

    const contentDataUrl = canvas.toDataURL('image/png');

    let PDF = new jsPDF({
      orientation: 'p',
      unit: 'mm',
      format: 'a4'
    });

    let topPosition = 10;
    let leftPosition = 0;

    PDF.addImage(contentDataUrl, 'PNG', leftPosition, topPosition,
fileWidth, fileHeight);
    PDF.save('Graph.pdf');
  }

  )
}


// Canvas Options

}
```

Bubble-Chart (Used to determine if at least three numerical variables are related or share some kind of pattern, show trends over time under special circumstances, or compare categorical variables.)



Html page

```
<div class="chart-wrapper" id="htmlData">
  <canvas baseChart
  [datasets]="bubbleChartData"
  [options]="bubbleChartOptions"
  [colors]="bubbleChartColors"
  [legend]="bubbleChartLegend"
  [chartType]="bubbleChartType">
</canvas>
</div>

<div>
  <button style="float: right" class="btn btn-success btn-block"
(click)="openPDF()">Download PDF</button>
</div>
```

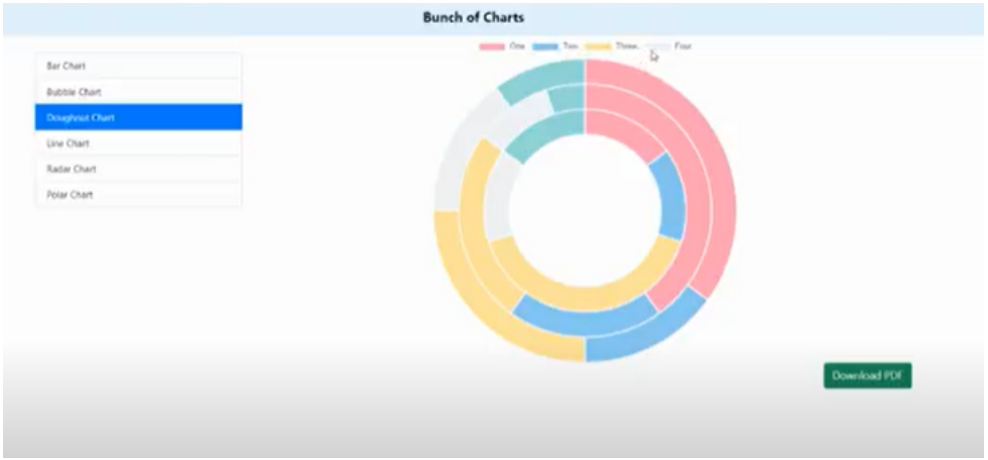
.TS File for Bubble chart

```
import { Component, ElementRef, ViewChild  } from '@angular/core';
import { ChartOptions, ChartDataSets } from 'chart.js';
import { Color } from 'ng2-charts';
import jsPDF from 'jspdf';
import html2canvas from 'html2canvas'

@Component({
  selector: 'app-bubble-chart',
```

```
  templateUrl: './bubble-chart.component.html',
  styleUrls: ['./bubble-chart.component.css']
})

export class BubbleChartComponent {

 public bubbleChartOptions: ChartOptions = {
  responsive: true,
  scales: {
    xAxes: [{
      ticks: {
        min: 0,
        max: 50,
      }
    }],
    yAxes: [{
      ticks: {
        min: 0,
        max: 50,
      }
    }]
  }
 };

 public bubbleChartType : Chart.ChartType = 'bubble';
 public bubbleChartLegend = true;

public bubbleChartData : Chart.ChartDataSets[] = [{
  data: [
    {x: 15, y: 15, r:15},
    {x: 25, y: 15, r:25},
    {x: 35, y: 25, r:33},
    {x: 45, y: 35, r:40},
    {x: 15, y: 20, r:20},
  ],
  label: 'Series A'
},];

public bubbleChartColors: Color[] = [
  {
    backgroundColor: [
      'magenta',
      'cyan',
```

```
      'green',
      'purple',
      'brown',
      'pink',
      'orange'
    ]
  }
];

  @ViewChild('htmlData') htmlData:ElementRef | any;

  constructor() { }

  // PDF Options

  public openPDF():void {
    let Data = document.getElementById('htmlData')!;

  // Canvas Options
    html2canvas(Data).then(canvas => {
        let fileWidth = 210;
        let fileHeight = canvas.height * fileWidth / canvas.width;

        const contentDataURL = canvas.toDataURL('image/png')


        let PDF = new jsPDF({ orientation: 'p', unit: 'mm', format:
'a4',});
        let topPosition = 10;
        let leftPosition = 0;
        PDF.addImage(contentDataURL, 'PNG', leftPosition, topPosition,
fileWidth, fileHeight)
        PDF.save('Graph.pdf');
    });
  }
  }
```

Doughnut chart (Shows the relationship of parts to a whole)



Html page

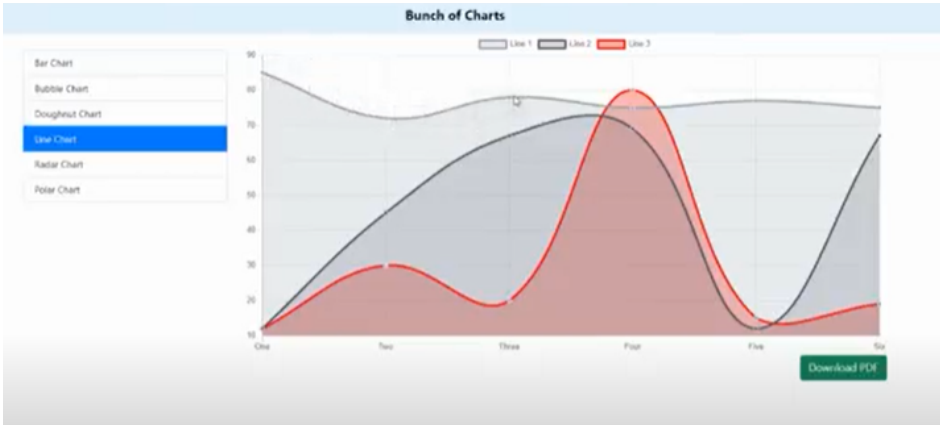
```
<div class="chart-wrapper" id="htmlData">
  <canvas baseChart
  [data]="doughnutChartData"
  [labels]="doughnutChartLabels"
  [chartType]="doughnutChartType">
</canvas>
</div>
<div>
  <button style="float: right" class="btn btn-success btn-block"
(click)="openPDF()">Download PDF</button>
</div>
```

.TS File for Doughnut chart

```
import { Component, OnInit } from '@angular/core';
import { ChartType } from 'chart.js';
import html2canvas from 'html2canvas';
import jsPDF from 'jspdf';
import { MultiDataSet, Label } from 'ng2-charts';

@Component({
  selector: 'app-doughnut-chart',
  templateUrl: './doughnut-chart.component.html',
  styleUrls: ['./doughnut-chart.component.css']
})
export class DoughnutChartComponent implements OnInit {
```

```
  doughnutChartLabels: Label[] = ['One', 'Two', 'Three', 'Four'];
  doughnutChartData: MultiDataSet = [
    [35, 15, 25, 15, 10],
    [40, 20, 25, 10, 5],
    [15, 15, 40, 15, 15]
  ];
  doughnutChartType: ChartType = 'doughnut';

  constructor() { }

  ngOnInit(): void {
  }

// PDF Options

public openPDF():void {
  let Data = document.getElementById('htmlData')!;

// Canvas Options
  html2canvas(Data).then(canvas => {
      let fileWidth = 210;
      let fileHeight = canvas.height * fileWidth / canvas.width;

      const contentDataURL = canvas.toDataURL('image/png')


      let PDF = new jsPDF({ orientation: 'p', unit: 'mm', format:
'a4',});
      let topPosition = 10;
      let leftPosition = 0;
      PDF.addImage(contentDataURL, 'PNG', leftPosition, topPosition,
fileWidth, fileHeight)
      PDF.save('Graph.pdf');
  });
}
}
```

Line Chart (Displays information as a series of points connected by straight line segments)



Html page

```
<div class="chart-wrapper" id="htmlData">
  <canvas baseChart
      [datasets]="lineChartData"
      [labels]="lineChartLabels"
      [options]="lineChartOptions"
      [colors]="lineChartColors"
      [legend]="lineChartLegend"
      [chartType]="lineChartType"
      [plugins]="lineChartPlugins">
  </canvas>
</div>
<div>
  <button style="float: right" class="btn btn-success btn-block"
(click)="openPDF()">Download PDF</button>
</div>
```

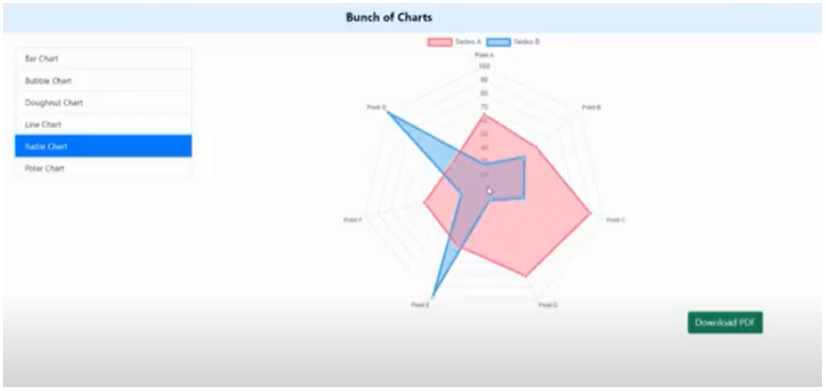
.TS File for Line chart

```
import { Component, OnInit } from '@angular/core';
import { ChartType } from 'chart.js';
import html2canvas from 'html2canvas';
import jsPDF from 'jspdf';
import { Color, Label } from 'ng2-charts';

@Component({
  selector: 'app-line-chart',
```

```
  templateUrl: './line-chart.component.html',
  styleUrls: ['./line-chart.component.css']
})
export class LineChartComponent implements OnInit {


  lineChartData: Chart.ChartDataSets[] = [
    { data: [12, 45, 67, 69, 12, 67], label: 'Line 1' },
    { data: [12, 45, 67, 69, 12, 67], label: 'Line 2' },
    { data: [15, 38, 20, 80, 15, 19], label: 'Line 3' },
    { data: [15, 30, 29, 80, 13, 19], label: 'Line 4' },
    { data: [16, 39, 20, 80, 15, 19], label: 'Line 5' },
  ];

  lineChartLabels: Label[] = ['One', 'Two', 'Three', 'Four', 'Five',
'Six'];

  lineChartOptions = {
    responsive: true,
  };

  public lineChartColors: Color[] = [
    { // grey
      backgroundColor: 'rgba(148,159,177,0.2)',
      borderColor: 'rgba(148,159,177,1)',
      pointBackgroundColor: 'rgba(148,159,177,1)',
      pointBorderColor: '#fff',
      pointHoverBackgroundColor: '#fff',
      pointHoverBorderColor: 'rgba(148,159,177,0.8)'
    },
    { // dark grey
      backgroundColor: 'rgba(77,83,96,0.2)',
      borderColor: 'rgba(77,83,96,1)',
      pointBackgroundColor: 'rgba(77,83,96,1)',
      pointBorderColor: '#fff',
      pointHoverBackgroundColor: '#fff',
      pointHoverBorderColor: 'rgba(77,83,96,1)'
    },
    { // red
      backgroundColor: 'rgba(255,0,0,0.3)',
      borderColor: 'red',
      pointBackgroundColor: 'rgba(148,159,177,1)',
      pointBorderColor: '#fff',
```

```
      pointHoverBackgroundColor: '#fff',
      pointHoverBorderColor: 'rgba(148,159,177,0.8)'
    }
  ];

  lineChartLegend = true;
  lineChartPlugins = [];
  lineChartType: ChartType = 'line';

  constructor() { }
  ngOnInit(): void {

  }

// PDF Options

public openPDF():void {
  let Data = document.getElementById('htmlData')!;

// Canvas Options
  html2canvas(Data).then(canvas => {
      let fileWidth = 210;
      let fileHeight = canvas.height * fileWidth / canvas.width;

      const contentDataURL = canvas.toDataURL('image/png')


      let PDF = new jsPDF({ orientation: 'p', unit: 'mm', format:
'a4',});
      let topPosition = 10;
      let leftPosition = 0;
      PDF.addImage(contentDataURL, 'PNG', leftPosition, topPosition,
fileWidth, fileHeight)
      PDF.save('Graph.pdf');
  });
}
}
```

Radar Chart (Used to compare two or more items or groups of various features or characteristics)



Html page

```
<div class="chart-wrapper" id="htmlData">
  <canvas baseChart
  [datasets]="radarChartData"
  [options]="radarChartOptions"
  [labels]="radarChartLabels"
  [chartType]="radarChartType">
</canvas>
</div>
<div>
  <button style="float: right" class="btn btn-success btn-block"
(click)="openPDF()">Download PDF</button>
</div>
```

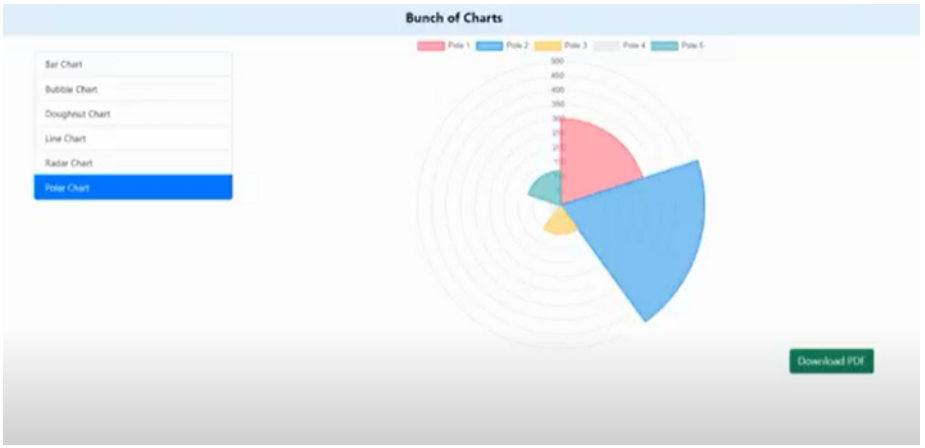
.TS File for Radar chart

```
import { Component, OnInit } from '@angular/core';
import { ChartDataSets, ChartType, RadialChartOptions } from
'chart.js';
import html2canvas from 'html2canvas';
import jsPDF from 'jspdf';
import { Label } from 'ng2-charts';

@Component({
  selector: 'app-radar-chart',
  templateUrl: './radar-chart.component.html',
  styleUrls: ['./radar-chart.component.css']
```

```
})
export class RadarChartComponent implements OnInit {

  public radarChartOptions: RadialChartOptions = {
    responsive: true,
  };
  public radarChartLabels: Label[] = ['Point A', 'Point B', 'Point C',
    'Point D', 'Point E', 'Point F', 'Point G'];

  public radarChartData: ChartDataSets[] = [
    { data: [65, 59, 90, 81, 56, 55, 40], label: 'Series A' },
    { data: [28, 48, 40, 19, 96, 27, 100], label: 'Series B' }
  ];
  public radarChartType: ChartType = 'radar';


  constructor() { }

  ngOnInit(): void {
  }

// PDF Options

public openPDF():void {
  let Data = document.getElementById('htmlData')!;

// Canvas Options
  html2canvas(Data).then(canvas => {
      let fileWidth = 210;
      let fileHeight = canvas.height * fileWidth / canvas.width;

      const contentDataURL = canvas.toDataURL('image/png')


      let PDF = new jsPDF({ orientation: 'p', unit: 'mm', format:
'a4',});
      let topPosition = 10;
      let leftPosition = 0;
      PDF.addImage(contentDataURL, 'PNG', leftPosition, topPosition,
fileWidth, fileHeight)
      PDF.save('Graph.pdf');
  });
}
```

```
}
```

Polar Chart (Use values and angels to show scientific data, can specify default measure)



Html page

```
<div class="chart-wrapper" id="htmlData">
  <canvas baseChart
  [data]="polarAreaChartData"
  [labels]="polarAreaChartLabels"
  [legend]="polarAreaLegend"
  [chartType]="polarAreaChartType">
</canvas>
</div>
<div>
  <button style="float: right" class="btn btn-success btn-block"
(click)="openPDF()">Download PDF</button>
</div>
```

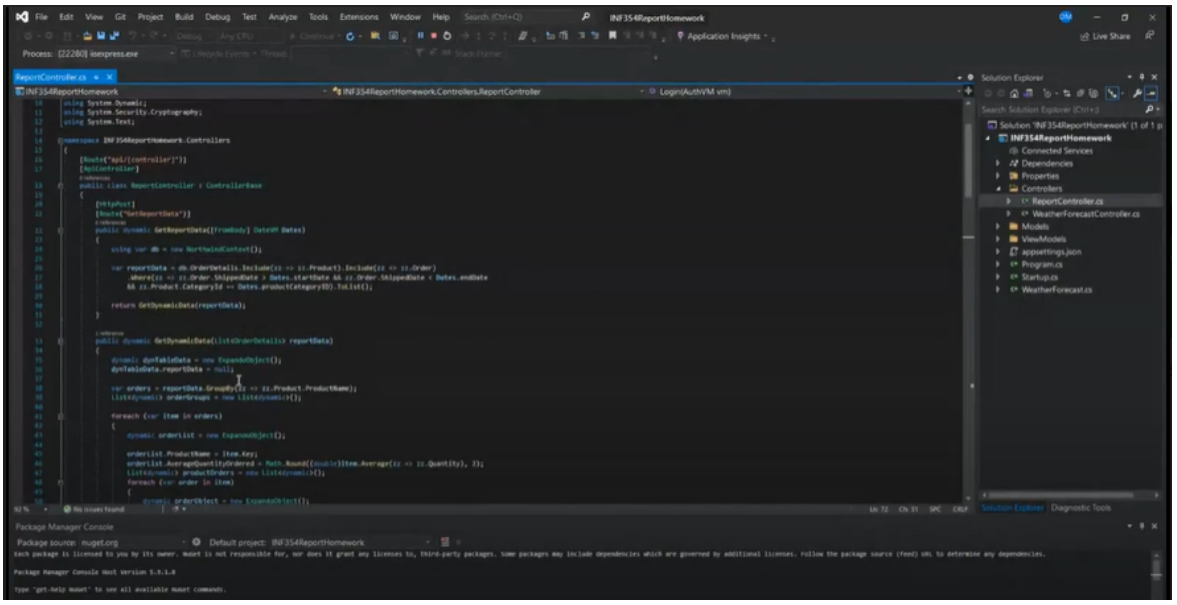
.TS File for Polar chart

```
import { Component, OnInit } from '@angular/core';
import { ChartType } from 'chart.js';
import html2canvas from 'html2canvas';
import jsPDF from 'jspdf';
import { Label, SingleDataSet } from 'ng2-charts';
```

```
@Component({
  selector: 'app-polar-chart',
  templateUrl: './polar-chart.component.html',
  styleUrls: ['./polar-chart.component.css']
})
export class PolarChartComponent implements OnInit {
  public polarAreaChartLabels: Label[] = ['Pole 1', 'Pole 2', 'Pole 3',
'Pole 4', 'Pole 5'];
  public polarAreaChartData: SingleDataSet = [300, 500, 100, 40, 120];
  public polarAreaLegend = true;

  public polarAreaChartType: ChartType = 'polarArea';
  constructor() { }

  ngOnInit(): void {
  }
// PDF Options

public openPDF():void {
  let Data = document.getElementById('htmlData')!;

// Canvas Options
  html2canvas(Data).then(canvas => {
      let fileWidth = 210;
      let fileHeight = canvas.height * fileWidth / canvas.width;

      const contentDataURL = canvas.toDataURL('image/png')


      let PDF = new jsPDF({ orientation: 'p', unit: 'mm', format:
'a4',});
      let topPosition = 10;
      let leftPosition = 0;
      PDF.addImage(contentDataURL, 'PNG', leftPosition, topPosition,
fileWidth, fileHeight)
      PDF.save('Graph.pdf');
  });
}
}
```

If Api or Json file is included

1) Get report data



Pie Chart Html

```
<p></p>
<h3>Pie Chart (with options)</h3>
<p></p>
<table>
  <tr>
      <td>
          <ngx-charts-pie-chart
          [results]="singleDataOption"
          [view]= "[700, 370]"
          [scheme]="colorScheme"
          [gradient]="gradient"
          [legend]="showLegend"
          [legendPosition] ="legendlocation"
          [labels]="showLabels"
          [doughnut]="isDoughnut"
                (select)="onSelect($event)"
                (activate)="onActivate($event)"
                (deactivate)="onDeactivate($event)">
          </ngx-charts-pie-chart>
      </td>
  </tr>
```

```
  <tr>
      <td>
          <ngx-charts-pie-grid
          [results]="singleDataOption"
          [view]= "[700, 370]"
          [scheme]="colorScheme"
                (select)="onSelect($event)"
                (activate)="onActivate($event)"
                (deactivate)="onDeactivate($event)">
          </ngx-charts-pie-grid>
      </td>
  </tr>
  <tr>
      <td>
          <ngx-charts-advanced-pie-chart
          [results]="singleDataOption"
          [view]= "[700, 370]"
          [scheme]="colorScheme"
          [gradient]="gradient"
                (select)="onSelect($event)"
                (activate)="onActivate($event)"
                (deactivate)="onDeactivate($event)">
          </ngx-charts-advanced-pie-chart>
      </td>
  </tr>
</table>
```

Pie Chart.ts file

```
import { Component, OnInit } from '@angular/core';
import { LegendPosition } from '@swimlane/ngx-charts';
import { bubbleData } from '../_data/bubbleData';
import { singleDataOption } from '../_data/singleData';
import { multiDataOptions } from '../_data/multiData'

@Component({
  selector: 'app-pie-chart',
  templateUrl: './pie-chart.component.html',
  styleUrls: ['./pie-chart.component.css']
})
export class PieChartComponent implements OnInit {

  singleDataOption: any[] = [];
```

```
  multiDataOptions: any[] = [];

  // options
  showLegend: boolean = true;
  showLabels: boolean = true;

  gradient: boolean = false;
  isDoughnut: boolean = true;

  legendlocation = LegendPosition.Below;

  colorScheme = {
    domain: ['aqua', 'blue', 'chartreuse', 'crimson', 'fuchsia',
'gray',
            'green', 'indigo', 'lime', 'magenta', 'navy', 'maroon',
'olive',
            'purple', 'red', 'silver', 'teal', 'yellow']
  };

  constructor() { Object.assign(this, { singleDataOption,
multiDataOptions }) }

  ngOnInit(): void {
  }

  onActivate(data: any): void {
    console.log('Activate', JSON.parse(JSON.stringify(data)));
  }

  onDeactivate(data: any): void {
    console.log('Deactivate', JSON.parse(JSON.stringify(data)));
  }

  onSelect(data: any): void {
    console.log('Item clicked', JSON.parse(JSON.stringify(data)));
  }
}
```

Bar Chart Html:

```
<p></p>
<h3>Bar Chart (with options)</h3>
<p></p>
<table>
```

```
<tr>
    <td>
        <ngx-charts-bar-vertical
            [results]="singleDataOption"
            [view] = "[700, 370]"
            [scheme]="colorScheme"
            [gradient]="gradient"
            [xAxis]="xAxis"
            [yAxis]="yAxis"
            [legend]="legend"
            [showXAxisLabel]="showXAxisLabel"
            [showYAxisLabel]="showYAxisLabel"
            [xAxisLabel]="xAxisLabel"
            [yAxisLabel]="xAxisLabel"
            [legendTitle]="legendTitle"
            [legendPosition]="legendlocation"
            [animations]="animations"
            [showGridLines]="showGridLines"
            [trimXAxisTicks]="trimXAxisTicks"
            [trimYAxisTicks]="trimYAxisTicks"
            [rotateXAxisTicks]="rotateXAxisTicks"
            [maxXAxisTickLength]="maxXAxisTickLength"
            [maxYAxisTickLength]="maxYAxisTickLength"
            [xAxisTickFormatting]="formatString"
            [yAxisTickFormatting]="formatNumber"
            [yAxisTicks]="yAxisTicks"
            [showDataLabel]="showDataLabel"
            [barPadding]="barPadding"
            [tooltipDisabled]="tooltipDisabled"
            [roundEdges]="roundEdges">
        </ngx-charts-bar-vertical>
    </td>
</tr>
<p></p>
<tr>
    <td>
        <ngx-charts-bar-horizontal
        [results]="singleDataOption"
        [view]= "[700, 370]"
        [scheme]="colorScheme"
        [gradient]="gradient"
        [xAxis]="xAxis"
        [yAxis]="yAxis"
```

```
            [legend]="legend"
            [showXAxisLabel]="showXAxisLabel"
            [showYAxisLabel]="showYAxisLabel"
            [xAxisLabel]="yAxisLabel"
            [yAxisLabel]="xAxisLabel"
            [legendTitle]="legendTitle"
            [legendPosition]="legendlocation"
            [animations]="animations"
            [showGridLines]="showGridLines"
            [trimXAxisTicks]="trimXAxisTicks"
            [trimYAxisTicks]="trimYAxisTicks"
            [rotateXAxisTicks]="rotateXAxisTicks"
            [maxXAxisTickLength]="maxXAxisTickLength"
            [maxYAxisTickLength]="maxYAxisTickLength"
            [xAxisTickFormatting]="formatNumber"
            [yAxisTickFormatting]="formatString"
            [xAxisTicks]="yAxisTicks"
            [showDataLabel]="showDataLabel"
            [barPadding]="barPadding"
            [tooltipDisabled]="tooltipDisabled"
            [roundEdges]="roundEdges">
            </ngx-charts-bar-horizontal>
        </td>
    </tr>
    <p></p>
    <tr>
        <td>
            <ngx-charts-bar-vertical-stacked
            [results]="multiDataOptions"
            [view]= "[700, 370]"
            [scheme]="colorScheme"
            [gradient]="gradient"
            [xAxis]="xAxis"
            [yAxis]="yAxis"
            [legend]="legend"
            [showXAxisLabel]="showXAxisLabel"
            [showYAxisLabel]="showYAxisLabel"
            [xAxisLabel]="xAxisLabel"
            [yAxisLabel]="yAxisLabel"
            [legendTitle]="legendTitleMulti"
            [legendPosition]="legendlocation"
            [animations]="animations"
            [showGridLines]="showGridLines"
```

```
            [showDataLabel]="showDataLabel"
            [barPadding]="barPadding"
            [tooltipDisabled]="tooltipDisabled">
            </ngx-charts-bar-vertical-stacked>
        </td>
    </tr>
    <p></p>
    <tr>
        <td>
            <ngx-charts-bar-horizontal-stacked
            [results]="multiDataOptions"
            [view]= "[700, 370]"
            [scheme]="colorScheme"
            [gradient]="gradient"
            [xAxis]="xAxis"
            [yAxis]="yAxis"
            [legend]="legend"
            [showXAxisLabel]="showXAxisLabel"
            [showYAxisLabel]="showYAxisLabel"
            [xAxisLabel]="yAxisLabel"
            [yAxisLabel]="xAxisLabel"
            [legendTitle]="legendTitleMulti"
            [legendPosition]="legendlocation"
            [animations]="animations"
            [showGridLines]="showGridLines"
            [showDataLabel]="showDataLabel"
            [barPadding]="barPadding"
            [tooltipDisabled]="tooltipDisabled">
            </ngx-charts-bar-horizontal-stacked>
        </td>
    </tr>
    <p></p>
    <tr>
        <td>
            <ngx-charts-bar-vertical-normalized
            [results]="multiDataOptions"
            [view]= "[700, 370]"
            [scheme]="colorScheme"
            [gradient]="gradient"
            [xAxis]="xAxis"
            [yAxis]="yAxis"
            [legend]="legend"
            [showXAxisLabel]="showXAxisLabel"
```

```
            [showYAxisLabel]="showYAxisLabel"
            [xAxisLabel]="xAxisLabel"
            [yAxisLabel]="yAxisLabel"
            [legendTitle]="legendTitleMulti"
            [legendPosition]="legendlocation"
            [animations]="animations"
            [showGridLines]="showGridLines"
            [barPadding]="barPadding"
            [tooltipDisabled]="tooltipDisabled">
          </ngx-charts-bar-vertical-normalized>
        </td>
    </tr>
    <p></p>
    <tr>
        <td>
            <ngx-charts-bar-horizontal-normalized
            [results]="multiDataOptions"
            [view]= "[700, 370]"
            [scheme]="colorScheme"
            [gradient]="gradient"
            [xAxis]="xAxis"
            [yAxis]="yAxis"
            [legend]="legend"
            [showXAxisLabel]="showXAxisLabel"
            [showYAxisLabel]="showYAxisLabel"
            [xAxisLabel]="yAxisLabel"
            [yAxisLabel]="xAxisLabel"
            [legendTitle]="legendTitleMulti"
            [legendPosition]="legendlocation"
            [animations]="animations"
            [showGridLines]="showGridLines"
            [barPadding]="barPadding"
            [tooltipDisabled]="tooltipDisabled">
          </ngx-charts-bar-horizontal-normalized>
        </td>
    </tr>
    <p></p>
    <tr>
        <td>
            <ngx-charts-bar-vertical-2d
            [results]="multiDataOptions"
            [view]= "[700, 370]"
            [scheme]="colorScheme"
```

```
            [gradient]="gradient"
            [xAxis]="xAxis"
            [yAxis]="yAxis"
            [legend]="legend"
            [showXAxisLabel]="showXAxisLabel"
            [showYAxisLabel]="showYAxisLabel"
            [xAxisLabel]="xAxisLabel"
            [yAxisLabel]="yAxisLabel"
            [legendTitle]="legendTitleMulti"
            [legendPosition]="legendlocation"
            [animations]="animations"
            [showGridLines]="showGridLines"
            [barPadding]="barPadding"
            [tooltipDisabled]="tooltipDisabled">
          </ngx-charts-bar-vertical-2d>
        </td>
    </tr>
    <p></p>
    <tr>
        <td>
            <ngx-charts-bar-horizontal-2d
            [results]="multiDataOptions"
            [view]= "[700, 370]"
            [scheme]="colorScheme"
            [gradient]="gradient"
            [xAxis]="xAxis"
            [yAxis]="yAxis"
            [legend]="legend"
            [showXAxisLabel]="showXAxisLabel"
            [showYAxisLabel]="showYAxisLabel"
            [xAxisLabel]="yAxisLabel"
            [yAxisLabel]="xAxisLabel"
            [legendTitle]="legendTitleMulti"
            [legendPosition]="legendlocation"
            [animations]="animations"
            [showGridLines]="showGridLines"
            [barPadding]="barPadding"
            [tooltipDisabled]="tooltipDisabled">
          </ngx-charts-bar-horizontal-2d>
        </td>
    </tr>
</table>
```

Bar Chart .ts File:

```
import { Component, OnInit } from '@angular/core';
import { LegendPosition } from '@swimlane/ngx-charts';
import { bubbleData } from '../_data/bubbleData';
import { singleDataOption } from '../_data/singleData';
import { multiDataOptions } from '../_data/multiData'

@Component({
  selector: 'app-bar-chart',
  templateUrl: './bar-chart.component.html',
  styleUrls: ['./bar-chart.component.css']
})

export class BarChartComponent implements OnInit {

  singleDataOption: any[] = [];
  multiDataOptions: any[] = [];

  // options
  legendTitle: string = 'Legend Title';
  legendTitleMulti: string = 'Time Marker';
  legend: boolean = true;
  legendlocation = LegendPosition.Right;
  xAxis: boolean = true;
  yAxis: boolean = true;
  yAxisLabel: string = 'Axis Label';
  xAxisLabel: string = 'Axis Label';
  showXAxisLabel: boolean = true;
  showYAxisLabel: boolean = true;
  maxXAxisTickLength: number = 30;
  maxYAxisTickLength: number = 30;
  trimXAxisTicks: boolean = false;
  trimYAxisTicks: boolean = false;
  rotateXAxisTicks: boolean = false;

  xAxisTicks: any[] = ['Item 1', 'Item 2', 'Item 3', 'Item 4', 'Item
5']
  yAxisTicks: any[] = [100, 500, 1000, 2000, 3000]

  animations: boolean = true; // animations on load

  showGridLines: boolean = true; // grid lines
```

```
  showDataLabel: boolean = true; // numbers on bars

  gradient: boolean = true;
  colorScheme = {
    domain: ['aqua', 'blue', 'chartreuse', 'crimson', 'fuchsia',
'gray',
           'green', 'indigo', 'lime', 'magenta', 'navy', 'maroon',
'olive',
           'purple', 'red', 'silver', 'teal', 'yellow']
  };

  activeEntries: any[] = []
  barPadding: number = 5
  tooltipDisabled: boolean = false;

  yScaleMax: number = 4000;

  roundEdges: boolean = false;

  constructor() { Object.assign(this, { singleDataOption,
multiDataOptions }); }

  ngOnInit(): void {
  }

  onSelect(event: any) {
    console.log(event);
  }

  onActivate(data: any): void {
    console.log('Activate', JSON.parse(JSON.stringify(data)));
  }

  onDeactivate(data: any): void {
    console.log('Deactivate', JSON.parse(JSON.stringify(data)));
  }

  formatString(input: string): string {
    return input.toUpperCase()
  }

  formatNumber(input: number): number {
    return input
```

```
  }
}
```

Bubble Chart Html:

```
<p></p>
<h3>Bubble Chart (with options)</h3>
<p></p>
<table>
  <tr>
      <td>
        <ngx-charts-bubble-chart
        [view]= "[700, 370]"
        [scheme]="colorScheme"
        [results]="bubbleData"
        [xAxis]="showXAxis"
        [yAxis]="showYAxis"
        [legend]="showLegend"
        [legendPosition] = "legendlocation"
        [showXAxisLabel]="showXAxisLabel"
        [showYAxisLabel]="showYAxisLabel"
        [xAxisLabel]="xAxisLabel"
        [yAxisLabel]="yAxisLabel"
        [yScaleMin]="yScaleMin"
        [yScaleMax]="yScaleMax"
        [minRadius]="minRadius"
        [maxRadius]="maxRadius"
            (select)="onSelect($event)"
            (activate)="onActivate($event)"
            (deactivate)="onDeactivate($event)">
      </ngx-charts-bubble-chart>
      </td>
  </tr>
</table>
```

Bubble Chart.ts File:

```
import { Component, OnInit } from '@angular/core';
import { LegendPosition } from '@swimlane/ngx-charts';
import { bubbleData } from '../_data/bubbleData';
import { singleDataOption } from '../_data/singleData';
import { multiDataOptions } from '../_data/multiData'
```

```
@Component({
  selector: 'app-bubble-chart',
  templateUrl: './bubble-chart.component.html',
  styleUrls: ['./bubble-chart.component.css']
})
export class BubbleChartComponent implements OnInit {

  bubbleData: any[] = [];

  // options
  showXAxis: boolean = true;
  showYAxis: boolean = true;
  showLegend: boolean = true;
  legendlocation = LegendPosition.Below;
  showXAxisLabel: boolean = true;
  showYAxisLabel: boolean = true;
  xAxisLabel: string = 'Axis Label';
  yAxisLabel: string = 'Axis Label';
  maxRadius: number = 20;
  minRadius: number = 5;
  yScaleMin: number = 70;
  yScaleMax: number = 85;

  gradient: boolean = false;
  colorScheme = {
    domain: ['aqua', 'blue', 'chartreuse', 'crimson', 'fuchsia',
'gray',
            'green', 'indigo', 'lime', 'magenta', 'navy', 'maroon',
'olive',
            'purple', 'red', 'silver', 'teal', 'yellow']
  };

  constructor() {
    Object.assign(this, { bubbleData });
  }

  onSelect(data: any): void {
    console.log('Item clicked', JSON.parse(JSON.stringify(data)));
  }

  onActivate(data: any): void {
    console.log('Activate', JSON.parse(JSON.stringify(data)));
```

```
  }

  onDeactivate(data: any): void {
    console.log('Deactivate', JSON.parse(JSON.stringify(data)));
  }

  ngOnInit(): void {
  }

}
```

Guage Chart Html:

```
<p></p>
<h3>Gauge Chart (with options)</h3>
<p></p>

<table>
  <tr>
      <td>
        <ngx-charts-gauge
        [view]= "[700, 370]"
        [scheme]="colorScheme"
        [results]="singleDataOption"
        [legend]="legend"
        [legendPosition]="legendlocation"
            (select)="onSelect($event)"
            (activate)="onActivate($event)"
            (deactivate)="onDeactivate($event)">
      </ngx-charts-gauge>
      </td>
  </tr>
</table>
```

Guage Chart.ts File:

```
import { Component, OnInit } from '@angular/core';
import { LegendPosition } from '@swimlane/ngx-charts';
import { bubbleData } from '../_data/bubbleData';
import { singleDataOption } from '../_data/singleData';
import { multiDataOptions } from '../_data/multiData'
```

```
@Component({
  selector: 'app-gauge-chart',
  templateUrl: './gauge-chart.component.html',
  styleUrls: ['./gauge-chart.component.css']
})
export class GaugeChartComponent implements OnInit {

  singleDataOption: any[] = [];

  legend: boolean = true;
  legendlocation = LegendPosition.Below;

  colorScheme = {
    domain: ['aqua', 'blue', 'chartreuse', 'crimson', 'fuchsia',
'gray',
            'green', 'indigo', 'lime', 'magenta', 'navy', 'maroon',
'olive',
            'purple', 'red', 'silver', 'teal', 'yellow']
  };

  constructor() {
    Object.assign(this, { singleDataOption });
  }

  onSelect(data: any): void {
    console.log('Item clicked', JSON.parse(JSON.stringify(data)));
  }

  onActivate(data: any): void {
    console.log('Activate', JSON.parse(JSON.stringify(data)));
  }

  onDeactivate(data: any): void {
    console.log('Deactivate', JSON.parse(JSON.stringify(data)));
  }

  ngOnInit(): void {
  }

}
```

Number Card Chart Html:

```
<p></p>
<h3>Number Cards (with options)</h3>
<p></p>
<table>
  <tr>
      <td>
          <ngx-charts-number-card
            [results]="singleDataOption"
            [view] = "[700, 370]"
            [scheme]="colorScheme"
            [cardColor]="cardColor"
                (select)="onSelect($event)">
          </ngx-charts-number-card>
      </td>
  </tr>
</table>
```

Number Card Chart.ts File:

```
import { Component, OnInit } from '@angular/core';
import { bubbleData } from '../_data/bubbleData';
import { singleDataOption } from '../_data/singleData';
import { multiDataOptions } from '../_data/multiData'

@Component({
  selector: 'app-number-card-chart',
  templateUrl: './number-card-chart.component.html',
  styleUrls: ['./number-card-chart.component.css']
})
export class NumberCardChartComponent implements OnInit {

  singleDataOption: any[] = [];
    colorScheme = {
    domain: ['aqua', 'blue', 'chartreuse', 'crimson', 'fuchsia',
'gray',
            'green', 'indigo', 'lime', 'magenta', 'navy', 'maroon',
'olive',
            'purple', 'red', 'silver', 'teal', 'yellow']
  };
  cardColor: string = 'silver';
```

```
  constructor() {
    Object.assign(this, { singleDataOption });
  }

  onSelect(event: any) {
    console.log(event);
  }

  ngOnInit(): void {
  }

}
```

Polar rader Chart Html:

```
<p></p>
<h3>Polar Radar Chart (with options)</h3>
<p></p>
<table>
  <tr>
      <td>
          <ngx-charts-polar-chart
              [results]="multiDataOptions"
              [view] = "[700, 370]"
              [scheme]="colorScheme"
              [legend]="legend"
              [legendPosition] ="legendlocation"
              [showXAxisLabel]="showXAxisLabel"
              [showYAxisLabel]="showYAxisLabel"
              [xAxis]="xAxis" [yAxis]="yAxis"
              [xAxisLabel]="xAxisLabel"
              [yAxisLabel]="yAxisLabel"
                    (select)="onSelect($event)">
          </ngx-charts-polar-chart>
      </td>
  </tr>
</table>
```

Polar rader Chart.ts File:

```
import { Component, OnInit } from '@angular/core';
import { LegendPosition } from '@swimlane/ngx-charts';
```

```
import { bubbleData } from '../_data/bubbleData';
import { singleDataOption } from '../_data/singleData';
import { multiDataOptions } from '../_data/multiData'

@Component({
  selector: 'app-polar-radar-chart',
  templateUrl: './polar-radar-chart.component.html',
  styleUrls: ['./polar-radar-chart.component.css']
})
export class PolarRadarChartComponent implements OnInit {

  multiDataOptions: any[] = [];

  // options
  legend: boolean = true;

  showLabels: boolean = true;
  legendlocation = LegendPosition.Below;
  animations: boolean = true;
  xAxis: boolean = true;
  yAxis: boolean = true;
  showYAxisLabel: boolean = true;
  showXAxisLabel: boolean = true;
  xAxisLabel: string = 'Axis Label';
  yAxisLabel: string = 'Axis Label';

  colorScheme = {
    domain: ['aqua', 'blue', 'chartreuse', 'crimson', 'fuchsia',
'gray',
           'green', 'indigo', 'lime', 'magenta', 'navy', 'maroon',
'olive',
           'purple', 'red', 'silver', 'teal', 'yellow']
  };

  constructor() {
    Object.assign(this, { multiDataOptions });
  }

  onSelect(event: any) {
    console.log(event);
  }

  ngOnInit(): void {
```

```
  }

}
```

Tree-map Chart Html:

```
<p></p>
<h3>Tree Map (with options)</h3>
<p></p>
<table>
  <tr>
      <td>
          <ngx-charts-tree-map
              [view] = "[700, 370]"
              [scheme]="colorScheme"
              [results]="singleDataOption"
              [gradient]="gradient"
              [animations]="animations"
              [labelFormatting]="labelFormatting"
                  (select)="onSelect($event)">
          </ngx-charts-tree-map>
      </td>
  </tr>
</table>
```

Tree-map.ts File:

```
import { Component, OnInit } from '@angular/core';
import { bubbleData } from '../_data/bubbleData';
import { singleDataOption } from '../_data/singleData';
import { multiDataOptions } from '../_data/multiData'

@Component({
  selector: 'app-tree-map-chart',
  templateUrl: './tree-map-chart.component.html',
  styleUrls: ['./tree-map-chart.component.css']
})
export class TreeMapChartComponent implements OnInit {

  singleDataOption: any[] = [];
```

```
  // options
  gradient: boolean = false;
  animations: boolean = true;

  colorScheme = {
    domain: ['aqua', 'blue', 'chartreuse', 'crimson', 'fuchsia',
'gray',
            'green', 'indigo', 'lime', 'magenta', 'navy', 'maroon',
'olive',
            'purple', 'red', 'silver', 'teal', 'yellow']
  };

  constructor() {
    Object.assign(this, { singleDataOption });
  }

  onSelect(event: any) {
    console.log(event);
  }

  labelFormatting(c: { label: any; }) {
    return `${(c.label)} Label`;
  }

  ngOnInit(): void {
  }

}
```

Area Line Chart Html:

```
<p></p>
<h3>Area Line Chart (with options)</h3>
<p></p>
<table>
  <tr>
      <td>
          <ngx-charts-line-chart
          [results]="multiDataOptions"
          [view] = "[700, 370]"
          [scheme]="colorScheme"
          [gradient]="gradient"
```

```
        [legend]="legend"
        [legendPosition] = "legendlocation"
        [showXAxisLabel]="showXAxisLabel"
        [showYAxisLabel]="showYAxisLabel"
        [xAxis]="xAxis"
        [yAxis]="yAxis"
        [xAxisLabel]="xAxisLabel"
        [yAxisLabel]="yAxisLabel"
        [timeline]="timeline">
        </ngx-charts-line-chart>
    </td>
</tr>
<tr>
    <td>
        <ngx-charts-area-chart
        [results]="multiDataOptions"
        [view] = "[700, 370]"
        [scheme]="colorScheme"
        [gradient]="gradient"
        [legend]="legend"
        [legendPosition] = "legendlocation"
        [showXAxisLabel]="showXAxisLabel"
        [showYAxisLabel]="showYAxisLabel"
        [xAxis]="xAxis"
        [yAxis]="yAxis"
        [xAxisLabel]="xAxisLabel"
        [yAxisLabel]="yAxisLabel"
        [timeline]="timeline">
        </ngx-charts-area-chart>
    </td>
</tr>
<tr>
    <td>
        <ngx-charts-area-chart-stacked
        [results]="multiDataOptions"
        [view] = "[700, 370]"
        [scheme]="colorScheme"
        [gradient]="gradient"
        [legend]="legend"
        [legendPosition] = "legendlocation"
        [showXAxisLabel]="showXAxisLabel"
        [showYAxisLabel]="showYAxisLabel"
        [xAxis]="xAxis"
```

```
          [yAxis]="yAxis"
          [xAxisLabel]="xAxisLabel"
          [yAxisLabel]="yAxisLabel"
          [timeline]="timeline">
          </ngx-charts-area-chart-stacked>
      </td>
  </tr>
  <tr>
      <td>
          <ngx-charts-area-chart-normalized
          [results]="multiDataOptions"
          [view] = "[700, 370]"
          [scheme]="colorScheme"
          [gradient]="gradient"
          [legend]="legend"
          [legendPosition] = "legendlocation"
          [showXAxisLabel]="showXAxisLabel"
          [showYAxisLabel]="showYAxisLabel"
          [xAxis]="xAxis"
          [yAxis]="yAxis"
          [xAxisLabel]="xAxisLabel"
          [yAxisLabel]="yAxisLabel"
          [timeline]="timeline">
          </ngx-charts-area-chart-normalized>
      </td>
  </tr>
</table>
```

Area Line Chart.ts File:

```
import { Component, OnInit } from '@angular/core';
import { LegendPosition } from '@swimlane/ngx-charts';
import { bubbleData } from '../_data/bubbleData';
import { singleDataOption } from '../_data/singleData';
import { multiDataOptions } from '../_data/multiData'

@Component({
  selector: 'app-area-line-chart',
  templateUrl: './area-line-chart.component.html',
  styleUrls: ['./area-line-chart.component.css']
})
export class AreaLineChartComponent implements OnInit {
```

```
  constructor() {
    Object.assign(this, { multiDataOptions });
  }

  multiDataOptions: any[] = [];

  // options
  legend: boolean = true;
  legendlocation = LegendPosition.Right;
  showLabels: boolean = true;
  animations: boolean = true;
  xAxis: boolean = true;
  yAxis: boolean = true;
  showXAxisLabel: boolean = true;
  showYAxisLabel: boolean = true;
  xAxisLabel: string = 'Axis Label';
  yAxisLabel: string = 'Axis Label';
  timeline: boolean = true;

  gradient: boolean = true;
  colorScheme = {
    domain: ['aqua', 'blue', 'chartreuse', 'crimson', 'fuchsia',
'gray',
            'green', 'indigo', 'lime', 'magenta', 'navy', 'maroon',
'olive',
            'purple', 'red', 'silver', 'teal', 'yellow']
  };

  onSelect(event: any) {
    console.log(event);
  }

  onActivate(data: any): void {
    console.log('Activate', JSON.parse(JSON.stringify(data)));
  }

  onDeactivate(data: any): void {
    console.log('Deactivate', JSON.parse(JSON.stringify(data)));
  }

  ngOnInit(): void {
  }
```

```
}
```

Heat-map Html:

```
<p></p>
<h3>Heat Map (with options)</h3>
<p></p>
<table>
  <tr>
      <td>
        <ngx-charts-heat-map
        [results]="multiDataOptions"
        [view] = "[700, 370]"
        [scheme]="colorScheme"
        [gradient]="gradient"
        [legend]="legend"
        [legendPosition] = "legendlocation"
        [showXAxisLabel]="showXAxisLabel"
        [showYAxisLabel]="showYAxisLabel"
        [xAxis]="xAxis"
        [yAxis]="yAxis"
        [xAxisLabel]="xAxisLabel"
        [yAxisLabel]="yAxisLabel"
            (select)="onSelect($event)"
            (activate)="onActivate($event)"
            (deactivate)="onDeactivate($event)">
        </ngx-charts-heat-map>
      </td>
  </tr>
</table>
```

Heat-map.ts File:

```
import { Component, OnInit } from '@angular/core';
import { LegendPosition } from '@swimlane/ngx-charts';
import { bubbleData } from '../_data/bubbleData';
import { singleDataOption } from '../_data/singleData';
import { multiDataOptions } from '../_data/multiData'

@Component({
  selector: 'app-heat-map',
```

```
  templateUrl: './heat-map.component.html',
  styleUrls: ['./heat-map.component.css']
})
export class HeatMapComponent implements OnInit {

  multiDataOptions: any[] = [];

  // options
  legend: boolean = true;
  legendlocation = LegendPosition.Below;
  showLabels: boolean = true;
  animations: boolean = true;
  xAxis: boolean = true;
  yAxis: boolean = true;
  showYAxisLabel: boolean = true;
  showXAxisLabel: boolean = true;
  xAxisLabel: string = 'Axis Label';
  yAxisLabel: string = 'Axis Label';

  gradient: boolean = true;
  colorScheme = {

    // https://html-color.codes/

    domain: ['deepskyblue', 'steelblue', 'mediumblue', 'darkblue',
'midnightblue', 'navy']

    // domain: ['aqua', 'blue', 'chartreuse', 'crimson', 'fuchsia',
'gray',
    //         'green', 'indigo', 'lime', 'magenta', 'navy', 'maroon',
'olive',
    //         'purple', 'red', 'silver', 'teal', 'yellow']

  };

  constructor() {
    Object.assign(this, { multiDataOptions });
  }

  onSelect(data: any): void {
    console.log('Item clicked', JSON.parse(JSON.stringify(data)));
  }
```

```
  onActivate(data: any): void {
    console.log('Activate', JSON.parse(JSON.stringify(data)));
  }

  onDeactivate(data: any): void {
    console.log('Deactivate', JSON.parse(JSON.stringify(data)));
  }

  ngOnInit(): void {
  }

}
```

Data Tables Html:

```
<p></p>
<!-- ---------------------------------------- -->
<div>
<h1>Single Data Options</h1>
<table class="table table-bordered table-hover table-sm">
  <thead class="table-primary">
      <tr>
        <th>Item Name</th>
        <th>Quantity</th>
      </tr>
  </thead>
  <tbody>
    <tr *ngFor = "let singleData of singleDataOption">
      <td>{{ singleData.name }}</td>
      <td>{{ singleData.value }}</td>
    </tr>
  </tbody>
</table>
</div>
<!-- ---------------------------------------- -->
<p></p>
<div>
  <h1>Multiple Chart Data</h1>

  <table class="table table-bordered table-hover table-sm">
   <thead class="table-primary">
     <tr>
```

```
      <th>Item Name</th>
      <th>Multiple Value Data</th>
     </tr>
   </thead>
   <tbody>
      <tr *ngFor = "let multiData of multiDataOptions">
        <td>{{ multiData.name }}</td>
        <td>
          <table class="table table-bordered table-hover table-sm">
            <thead class="table-primary">
              <tr>
                <th>Name</th>
                <th>Value</th>

               </tr>
            </thead>
            <tbody>
              <tr *ngFor = "let nestedMultiData of multiData?.series">
                <td>{{ nestedMultiData.name }}</td>
                <td>{{ nestedMultiData.value }}</td>
              </tr>
            </tbody>
          </table>
        </td>
      </tr>
    </tbody>
  </table>
  </div>
<!-- ---------------------------------------- -->
<p></p>
<div>
  <h1>Bubble Chart Data</h1>

  <table class="table table-bordered table-hover table-sm">
   <thead class="table-primary">
     <tr>
      <th>Item Name</th>
      <th>Bubble Data</th>
     </tr>
   </thead>
   <tbody>
      <tr *ngFor = "let simpleBubbleData of bubbleData">
        <td>{{ simpleBubbleData.name }}</td>
```

```
        <td>
          <table class="table table-bordered table-hover table-sm">
            <thead class="table-primary">
              <tr>
                <th>Month</th>
                <th>Name</th>
                <th>Total</th>
                <th>Volume</th>
               </tr>
            </thead>
            <tbody>
              <tr *ngFor = "let nestedBubbleData of
simpleBubbleData?.series">
                <td>{{ nestedBubbleData.name }}</td>
                <td>{{ nestedBubbleData.x }}</td>
                <td>{{ nestedBubbleData.y }}</td>
                <td>{{ nestedBubbleData.r }}</td>
              </tr>
            </tbody>
          </table>
        </td>
      </tr>
    </tbody>
  </table>
  </div>
  <!-- ---------------------------------------- -->
<router-outlet></router-outlet>
```

Data Tables.ts File:

```
import { Component, OnInit } from '@angular/core';
import { bubbleData } from '../_data/bubbleData';
import { singleDataOption } from '../_data/singleData';
import { multiDataOptions } from '../_data/multiData'

@Component({
  selector: 'app-data-tables',
  templateUrl: './data-tables.component.html',
  styleUrls: ['./data-tables.component.css']
})
export class DataTablesComponent implements OnInit {
```

```
  title = 'Data Options';
  public bubbleData: [] | any = bubbleData;
  public singleDataOption: [] | any = singleDataOption;
  public multiDataOptions: [] | any = multiDataOptions

  constructor() { }
  ngOnInit(): void {
  }
}
```

What the json file called singleData.ts looks like:

```
export const singleDataOption = [
  {
    "name": "Pepsi Max",
    "value": 1345
  }, {
    "name": "Coke Zero",
    "value": 3553
  }, {
    "name": "Sprite",
    "value": 2345
  }, {
    "name": "Creme Soda",
    "value": 5345
  }, {
    "name": "Fanta Orange",
    "value": 4333
  }
];
```