Angular Assignment by Akshit Sanghani

Task A

MODULE (app.module.ts):-

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';


import { NgChartsModule } from 'ng2-charts'
import { HttpClientModule } from '@angular/common/http';

@NgModule({
  declarations: [
    AppComponent],
  imports: [
    BrowserModule,
    AppRoutingModule,
    HttpClientModule,
    NgChartsModule,
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

MODELS (empClass.ts):-

```
export class EmpData {
    Id!:string;
    EmployeeName!:string;
    StarTimeUtc!:Date;
    EndTimeUtc!:Date;
    EntryNotes!:string;
    DeletedOn!:null | boolean
    WorkingHours!:number
  }

  export class EmpWorkingHours{
    EmployeeName!:string;
    TotalWorkingHours!:number;
    HoursRequired!:boolean
  }
```

SERVICE (api.service.ts):-

```
import { Injectable} from '@angular/core';
import { EmpData, EmpWorkingHours } from './models/empClass';
import { HttpClient } from '@angular/common/http';
import{ map } from 'rxjs'

@Injectable({
  providedIn: 'root'
})
export class ApiService{
constructor(private http:HttpClient) { }

getData(){
  const url:string ='https://rc-vault-fap-live-
1.azurewebsites.net/api/gettimeentries?code=vO17RnE8vuzXzPJo5eaLLjXjmRW07law99
QTD90zat9FfOQJKKUcgQ=='
  return this.http.get<EmpData[]>(url)
}
}
```

HTML (app.component.html):-

```
<h1>ANGULAR ASSIGNMENT</h1>
<!-- TABLE -->
<table style="width: auto;">
  <thead>
    <tr>
      <th>Employee</th>
      <th>Hours worked</th>
    </tr>
  </thead>
  <tbody>

    <tr *ngFor="let emp of empWorkingHours; let i= index" id="empTable"
[ngClass]="emp.TotalWorkingHours < 100 ? 'RedBG' : 'WhiteBG'">
      <td>{{emp.EmployeeName}}</td>
      <td>{{emp.TotalWorkingHours}}</td>
    </tr>

  </tbody>
</table>
```

TS (app.component.ts):-

```
import { Component, OnInit } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { EmpData, EmpWorkingHours } from './models/empClass';
import { ApiService } from './api.service';
```

```
import { ChartDataset, ChartType } from 'chart.js';

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

  employees!:EmpData[];
  empWorkingHours: EmpWorkingHours[]=[];
  minRequirement:boolean = true;
  labels:string[] = [];
  values:number[] = [];
  pieChartLabels:any;
  pieChartData:any;

  constructor(private http:HttpClient,private apiService:ApiService){}

  pieChartOptions = {
    responsive: true,
    animationEnabled:true,
    title:{text :"Hours Worked by per employee"}
  }

ngOnInit(){
  // FOR TABLE
  // Adding JSON Values into an array:-
  this.apiService.getData().subscribe(data => {
  this.employees = data;

  // TO GET WORKING HOURS OF EACH EMPLOYEE
  for (let i = 0; i < this.employees.length; i++) {
    const startDate = new Date(this.employees[i].StarTimeUtc);
    const endDate = new Date(this.employees[i].EndTimeUtc);
    const diffMs = endDate.getTime() - startDate.getTime();
    const hours = diffMs / (1000 * 60 * 60);
    const rHours = Math.round(hours * 10)/10
    this.employees[i].WorkingHours = rHours
 }

// FOR CALCULATING AND AGGREGATING EMPLOYEES' TOTAL WORKED HOURS:-

 for(let emp of this.employees){

  var existingEmp = this.empWorkingHours.find(x => x.EmployeeName ==
emp.EmployeeName);
```

```
  if(existingEmp == null)
  {
    let e:EmpWorkingHours = new EmpWorkingHours();
    e.EmployeeName = emp.EmployeeName;
    e.TotalWorkingHours = emp.WorkingHours;
    this.empWorkingHours.push(e);
  }
  else
  {
    var idx = this.empWorkingHours.findIndex(x => x.EmployeeName ==
emp.EmployeeName);
    this.empWorkingHours[idx].TotalWorkingHours += emp.WorkingHours;
  }
}
// FOR CHANGING NAME OF EMPLOYEE WHOSE VALUE IS NULL
var idx = this.empWorkingHours.findIndex(x => x.EmployeeName == null);
if(idx != -1)
  this.empWorkingHours[idx].EmployeeName = "NA";

    // FOR SORTING THE EMPLOYEES
  this.empWorkingHours.sort(
    (a,b)=>{
    const result =  a.TotalWorkingHours - b.TotalWorkingHours
    return result
    })

   //FOR PIE CHART LABELS
  this.empWorkingHours.map(x => {
    if(x.EmployeeName == null)
      this.labels.push("NA");
    else
      this.labels.push(x.EmployeeName)
  })

  //FOR PIE CHART DATA
  this.empWorkingHours.map(x => {
    this.values.push(x.TotalWorkingHours)
  })

  console.log("akshit");
  console.log(this.labels);
  console.log(this.values);

  this.pieChartLabels =  this.labels;
  this.pieChartData = [{
    data:this.values}];

})
```

```
  }
}
```

SCREENSHOT OF THE OUTPUT (TASK A):-

ANGULAR ASSIGNMENT

Employee	Hours worked
NA	30.5
Tamoy Smith	91.19999999999999
Raju Sunuwar	99.40000000000005
Rita Alley	114.09999999999998
Kavvay Verma	163.7
Tim Perkinson	171.60000000000002
Mary Poppins	174.29999999999995
Abhay Singh	197.80000000000004
John Black	203.40000000000003
Stewart Malachi	207.8
Patrick Huthinson	217.99999999999997

TASK B

Note:

- Code for Module, Model, Service, Component TS file are the same as above shown in Task A.
- NPM Package used for generation of pie chart is https://www.npmjs.com/package/ng2-charts

HTML (app.component.html):-

```
<!-- PIE CHART -->
<div style="width: 400px;">
  <canvas
  id="myChart"
      baseChart
      [type]="'pie'"
      [datasets]="pieChartData"
      [labels]="pieChartLabels"
      [options]="pieChartOptions"
      [legend]="true"
      >
  </canvas>
</div>
```

SCREENSHOT OF THE OUTPUT (TASK B):-