**home.component**

**html**

<body>

  <nav class="navbar navbar-expand-lg navbar-light bg-light shadow fixed-top">

    <button class="navbar-toggler navbar-toggler-right" type="button" data-toggle="collapse"

      data-target="#navbarTogglerDemo02" aria-controls="navbarTogglerDemo02" aria-expanded="false"

      aria-label="Toggle navigation">

      <span class="navbar-toggler-icon"></span>

    </button>

    <a class="navbar-brand"><img src="assets/Voice.ai-logo.png" alt="logo"></a>

    <div class="collapse navbar-collapse custom-navbar" id="navbarTogglerDemo02">

      <ul class="navbar-nav ml-auto right-nav">

        <li class="nav-item">

          <a class="nav-link"><img class="logohome" src="assets/Home.jpg" alt="logo-home"></a>

        </li>

        <li class="nav-item dropdown">

          <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-toggle="dropdown"

            aria-haspopup="true" aria-expanded="false">

            Johan deo

          </a>

          <div class="dropdown-menu" aria-labelledby="navbarDropdown">

            <a class="dropdown-item" routerLink="/login">logout</a>

          </div>

        </li>

        <li class="nav-item">

          <a class="nav-link" href="#"><img src="assets/user.png" alt="arrow" class="user-img"></a>

        </li>

      </ul>

    </div>

  </nav>

  <section class="bg-image-home">

    <div class="container ">

      <div class="row">

        <div class="col-md-12 first-container first-image">

          <div class="row">

            <div class="col-md-6 col-12 order-sm-1 order-2 text-col">

              <h1>CRM - Call Matedata Analysis</h1>

              <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et

                dolore magna aliqua.</p>

              <button type="button" class="btn btn-primary" routerLink="/crmoverview">View Dashboard</button>

            </div>

            <div class="col-md-6 col-sm-12 order-1"><img src="assets/CRM-analysis-single.jpg" alt="crm-analysis"

                class="row-img"></div>

          </div>

        </div>

      </div>

    </div>

    <div class="container mt-4">

      <div class="row">

        <div class="col-md-12 second-container second-image">

          <div class="row">

            <div class="col-md-6 col-sm-12 col-12"><img src="assets/call-ingestion-single.jpg"

                alt="conversional-analysis" class="row-img"></div>

            <div class="col-md-6 col-sm-12 col-12 text-col">

              <h1>Conversational Analytics</h1>

              <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et

                dolore magna aliqua.</p>

              <button type="button" class="btn btn-primary" routerLink="/cioverview">View Dashboard</button>

            </div>

          </div>

        </div>

      </div>

    </div>

  </section>

</body>

**Css**

body {

    background: linear-gradient(to bottom, #0f75bc 50%, #052942 100%);

    background-repeat: no-repeat;

    background-size: cover;

    height: auto;

}

li.nav-item.dropdown {

    margin-top: 10px;

}

img.logohome {

    margin-top: 5px;

}

.first-image, .second-image {

    background-color: #ffffff;

}

img.row-img {

    width: 100%;

    border-radius: 10px;

    height: 100%;

}

.text-col {

    padding: 4%;

}

    /\* .first-image{

    background-image: url(../code/assets/home-screen/CRM-analysis.jpg);

    background-repeat: no-repeat;

    background-size:cover;

    padding: 5%;

    border-radius: 12px;

    }

    .second-image{

    background-image: url(../code/assets/home-screen/call-ingestion.jpg);

    background-repeat: no-repeat;

    background-size:cover;

    padding: 5%;

    border-radius: 12px;

    } \*/

.first-container {

    margin-top: 10%;

    padding-right: 0px;

    padding-left: 0px;

    border-radius: 10px;

}

.second-container {

    padding-right: 0px;

    padding-left: 0px;

    border-radius: 10px;

}

.bg-image-home h1 {

    font-size: 35px;

    font-family: 'Open Sans', sans-serif;

    color: #2e384d;

}

.bg-image-home p {

    font-family: 'Open Sans', sans-serif;

    color: #b0bac9;

    font-size: 16px;

}

button.btn-primary {

    background-color: #0f75bc;

    padding: 10px 85px;

    color: #f7f7f2;

    margin-top: 2%;

}

.second-image h1 {

    text-align: right;

}

.second-image p {

    text-align: right;

}

.second-image button.btn-primary {

    float: right;

}

img.user-img {

    width: 40px;

}

section.bg-image-home {

    padding: 4% 0%;

}

/\*\*\*\*\*MOBILE MEDIA QUERY\*\*\*\*\*/

@media only screen and (max-width: 768px) {

    .first-container {

        margin-top: 30%;

        text-align: center;

    }

    .bg-image-home h1 {

        font-size: 24px;

    }

    .second-container {

        text-align: center !important;

    }

    .second-image h1 {

        text-align: center;

    }

    .second-image p {

        text-align: center;

    }

    .second-image button.btn-primary {

        float: none;

    }

    .bg-image-home {

        padding: 4%;

    }

    .bg-image-home {

        height: 100%;

    }

    .text-col {

        padding: 8%;

    }

    section.bg-image-home {

        padding: 4% 4%;

    }

}

/\*\*\*\*\*IPAD MEDIA QUERY\*\*\*\*\*/

@media only screen

and (min-device-width : 768px)

and (max-device-width : 1024px) {

    .first-container {

        margin-top: 20%;

        text-align: center;

    }

    section.bg-image-home {

        padding: 8% 0% 50% 0%;

    }

    .bg-image-home {

        height: 100%;

    }

    .second-container {

        text-align: center !important;

    }

    .second-image h1 {

        text-align: center;

    }

    .second-image p {

        text-align: center;

    }

    .second-image button.btn-primary {

        float: none;

    }

}

**Ts**

import { Component, OnInit } from '@angular/core';

import { Router } from '@angular/router';

@Component({

  selector: 'app-home',

  templateUrl: './home.component.html',

  styleUrls: ['./home.component.css']

})

export class HomeComponent implements OnInit {

  constructor(private router: Router) { }

  ngOnInit() {

  }

  onLogout(){

    this.router.navigate(

      ['login']);

  }

}

**login Component**

**html**

<nav class="navbar navbar-expand-sm navbar-dark shadow ">

    <!-- Brand/logo -->

    <a class="navbar-brand" >

        <img src="assets/Voice.ai-logo.png" alt="logo" class="logo">

    </a>

    <!-- Links -->

    <ul class="navbar-nav ml-auto">

        <li class="nav-item">

            <img src="assets/Eclerx-logo.jpg" alt="logo" class="logo">

        </li>

    </ul>

</nav>

<section class="bg-img">

    <div class="container">

        <div class="row">

            <div class="col-md-3"></div>

            <div class="col-md-6">

                <div class="card">

                    <div class="card-body">

                        <h4 class="card-title">Login to continue</h4>

                        <form [formGroup]="loginForm" (ngSubmit)="onSubmit()">

                            <div class="form-group">

                                <label>USERNAME</label>

                                <input [class.is-invalid]="emails.invalid && emails.touched" formControlName="emails"

                                    type="text" class="form-control">

                                <div \*ngIf="emails.invalid && emails.touched">

                                    <small \*ngIf="emails.errors ?.required" class="text-danger">username is

                                        requied</small>

                                </div>

                            </div>

                            <div class="form-group">

                                <label>PASSWORD</label>

                                <input [class.is-invalid]="password.invalid && password.touched"

                                    formControlName="password" type="password" class="form-control">

                                <div \*ngIf="password.invalid && password.touched">

                                    <small \*ngIf="password.errors ?.required" class="text-danger">password is

                                        required</small>

                                </div>

                            </div>

                            <button [disabled]='!loginForm.valid' class="btn btn-primary" type="submit">Login</button>

                        </form>

                    </div>

                </div>

            </div>

            <div class="col-md-3"></div>

        </div>

    </div>

</section>

**Css**

.bg-img{

    background: url('src/assets/BG.jpg');

    height: auto !important;

    background-size: cover !important;

    background-repeat: no-repeat !important;

}

.card{

    margin:17vh 0;

    border-radius: 12px;

}

.card-body {

    padding: 3.25rem;

}

.card-title {

    margin-bottom: 1.75rem;

    font-size: 30px;

    font-family: 'Open Sans', sans-serif;

}

.btn-block {

    margin-top: 6vh;

}

label {

    font-size: 14px;

    font-weight: 500;

    color: #cecece;

    font-family: 'Open Sans', sans-serif;

}

.form-control {

    background-color: #ececec47;

    border: 1px solid #ececec;

    font-family: 'Open Sans', sans-serif;

}

nav.navbar.navbar-expand-sm.navbar-dark.shadow {

    background-color: white;

}

button.btn.btn-primary {

    width: 100%;

}

**ts**

import { Component, OnInit } from '@angular/core';

import { FormBuilder, FormGroup, Validators } from '@angular/forms';

import { Router } from '@angular/router';

import { LoginServiceService } from '../services/login-service.service';

import { CookieService } from 'ngx-cookie-service';

import { AuthService } from '../auth.service';

@Component({

  selector: 'app-login',

  templateUrl: './login.component.html',

  styleUrls: ['./login.component.css']

})

export class LoginComponent implements OnInit {

  loginForm: FormGroup;

  get emails() {

    return this.loginForm.get('emails');

  }

  get password() {

    return this.loginForm.get('password');

  }

  constructor(private fb: FormBuilder,

     private loginService: LoginServiceService,

     private router: Router,

     private cookie: CookieService,

     private authService:AuthService) { }

  ngOnInit() {

    this.loginForm = this.fb.group({

      emails: ['', [Validators.required]],

      password: ['', [Validators.required]]

    });

  }

  onSubmit() {

    // this.loginService.loginServiceFun(this.loginForm.value).subscribe(data =>{

    //   if(data.success){

    //     console.log("login agent value ", this.loginForm.value);

    //     alert("you are successfully logged in");

    //     this.router.navigateByUrl('/home');

    //   }

    // })

    if (this.loginForm.invalid) {

      this.router.navigate(

        ['signup']);

      return;

    } else {

      this.cookie.set("emails", this.loginForm.value.emails);

      this.cookie.set("password", this.loginForm.value.password);

      this.authService.CreateCookie();

      // alert("Successfully login");

      this.router.navigate(

        ['home']);

    }

    // this.router.navigateByUrl('/home');

  }

}

**app component**

**html**

<router-outlet></router-outlet>

**App.module.ts**

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

import { NgModule } from '@angular/core';

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

import { AppComponent } from './app.component';

import { HomeComponent } from './home/home.component';

import { LoginComponent } from './login/login.component';

import { ReactiveFormsModule, FormsModule } from '@angular/forms';

import { HttpClientModule } from '@angular/common/http';

import { CRMOverviewComponent } from './crm-overview/crm-overview.component';

import { CRMCallMonitoringComponent } from './crm-call-monitoring/crm-call-monitoring.component';

import { CRMNavBarComponent } from './crm-nav-bar/crm-nav-bar.component';

import { CIOverviewComponent } from './ci-overview/ci-overview.component';

import { CIAgentComplianceMonitoringComponent } from './ci-agent-compliance-monitoring/ci-agent-compliance-monitoring.component';

import { CINavBarComponent } from './ci-nav-bar/ci-nav-bar.component';

import { CookieService } from 'ngx-cookie-service';

import { Daterangepicker } from 'ng2-daterangepicker';

import { HashLocationStrategy, LocationStrategy  } from '@angular/common';

import { BrowserAnimationsModule } from '@angular/platform-browser/animations';

import { AngularMultiSelectModule } from 'angular2-multiselect-dropdown';

import { NgxPaginationModule } from 'ngx-pagination';

@NgModule({

  declarations: [

    AppComponent,

    HomeComponent,

    LoginComponent,

    CRMOverviewComponent,

    CRMCallMonitoringComponent,

    CRMNavBarComponent,

    CIOverviewComponent,

    CIAgentComplianceMonitoringComponent,

    CINavBarComponent,

  ],

  imports: [

    BrowserModule,

    AppRoutingModule,

    ReactiveFormsModule,

    FormsModule,

    HttpClientModule,

    Daterangepicker,

    BrowserAnimationsModule,

    AngularMultiSelectModule,

    NgxPaginationModule

  ],

  providers: [ CookieService, {provide : LocationStrategy , useClass: HashLocationStrategy} ],

  bootstrap: [AppComponent]

})

export class AppModule { }

**app.routing.module.ts**

import { NgModule } from '@angular/core';

import { Routes, RouterModule } from '@angular/router';

import { LoginComponent } from './login/login.component';

import { HomeComponent } from './home/home.component';

import { CRMOverviewComponent } from './crm-overview/crm-overview.component';

import { CRMCallMonitoringComponent } from './crm-call-monitoring/crm-call-monitoring.component';

import { CIOverviewComponent } from './ci-overview/ci-overview.component';

import { CIAgentComplianceMonitoringComponent } from './ci-agent-compliance-monitoring/ci-agent-compliance-monitoring.component';

import { AuthGuard } from './auth.guard';

const routes: Routes = [

  { path: '', redirectTo: '/login', pathMatch: 'full'},

  { path: 'login', component: LoginComponent },

  { path: 'home' , component: HomeComponent },

  { path: 'crmoverview' , component: CRMOverviewComponent },

  { path: 'crmcall' , component: CRMCallMonitoringComponent },

  { path: 'cioverview' , component: CIOverviewComponent  },

  { path: 'ciagent' , component: CIAgentComplianceMonitoringComponent },

];

@NgModule({

  imports: [RouterModule.forRoot(routes)],

  exports: [RouterModule]

})

export class AppRoutingModule { }

**ci-overview**

**html**

<app-ci-nav-bar></app-ci-nav-bar>

<body class="sb-nav-fixed">

    <div id="layoutSidenav">

        <div id="layoutSidenav\_nav">

            <p class="top-text"><span style="font-weight: 600;">FILTERS</span></p>

            <nav class="sb-sidenav accordion sb-sidenav-light toggled" id="sidenavAccordion">

                <div class="sb-sidenav-menu">

                    <div class="nav">

                        <div class="nav-ul-item pt-3 pl-3 pr-3 sidenav-menus">

                            <div class="row">

                                <div class="col-md-4 col">

                                    <p class="side-heading">FOLDER</p>

                                </div>

                                <div class="col-md-8 col text-right">

                                    <!-- Button to Open the Modal -->

                                    <button type="button" class="modal-btn" data-toggle="modal"

                                        data-target="#myModal">+ADD

                                        FOLDER</button>

                                </div>

                            </div>

                            <div class="row sidebar-heading">

                                <form [formGroup]="searchFolderForm" class="form-horizontal">

                                    <div class="form-group has-feedback">

                                        <input type="text" class="form-control custom" id="inputSuccess"

                                            placeholder="NAME" formControlName="searchFolder">

                                        <div class="text-right"><img type="submit" class="search-img icon-search-folder"

                                                src="assets/search-icon.png"

                                                (click)="onSearchFolderSubmit(searchFolderForm.value)">

                                        </div>

                                    </div>

                                </form>

                            </div>

                            <div class="row">

                                <div class="col-md-12 col-scroll">

                                    <form class="form-radio-btns">

                                        <div \*ngFor="let info of information[2]?.folder\_names\_data" class="row">

                                            <div class="col-md-8 col">

                                                <div class="custom-control custom-radio custom-control-inline">

                                                    <input type="radio" class="custom-control-input" name="example1"

                                                        [ngModel]="folderRadiobtn"

                                                        (ngModelChange)="folderRadiobtnEvent($event)" value="{{info}}">

                                                    <label class="custom-control-label">{{info}}</label>

                                                </div>

                                            </div>

                                            <div class="col-md-4 col text-right"><label for="{{info}}"><img

                                                        src="assets/upload.png"></label><input id="{{info}}" type="file"

                                                    style="display: none;" /></div>

                                        </div>

                                    </form>

                                </div>

                            </div>

                        </div>

                        <div class="h-divider"></div>

                        <div class="nav-ul-item p-3 sidenav-menus">

                            <div class="row">

                                <div class="col">

                                    <p class="side-heading">START/END DATE</p>

                                    <input type="text" name="daterangeInput" daterangepicker [options]="options"

                                        (selected)="selectedDate($event, daterange)" id="datepick" />

                                    <div class="text-right"><img class="search-img icon-calendar"

                                            src="assets/Calendar Icon.png">

                                    </div>

                                </div>

                            </div>

                        </div>

                        <div class="h-divider"></div>

                        <div class="nav-ul-item p-3 sidenav-menus">

                            <div class="nav-ul-item pt-3 sidenav-menus">

                                <div class="row">

                                    <div class="col-sm-4 col">

                                        <p class="side-heading">QUERIES</p>

                                    </div>

                                    <div class="col-sm-8 col text-right">

                                        <!-- Button to Open the Modal -->

                                        <button type="button" class="modal-btn" data-toggle="modal"

                                            data-target="#myModal1">+ADD

                                            QUERY</button>

                                    </div>

                                </div>

                                <div class="row sidebar-heading">

                                    <form [formGroup]="searchQueryForm" class="form-horizontal">

                                        <div class="form-group has-feedback">

                                            <input type="text" class="form-control custom" placeholder="NAME"

                                                formControlName="searchQuery">

                                            <div class="text-right"><img class="search-img icon-search-query"

                                                    src="assets/search-icon.png"

                                                    (click)="onSearchQuerySubmit(searchQueryForm.value)">

                                            </div>

                                        </div>

                                    </form>

                                </div>

                                <div class="row">

                                    <div class="col-md-12 col-scroll">

                                        <form class="form-radio-btns">

                                            <div class="custom-control custom-radio custom-control-inline">

                                                <input type="radio" class="custom-control-input" id="customRadio6"

                                                    name="example1" [ngModel]="queryRadiobtn"

                                                    (ngModelChange)="queryRadiobtnEvent($event)" value="1radio">

                                                <label class="custom-control-label" for="customRadio6">Lorem ipsum dolor

                                                    sit</label>

                                            </div>

                                            <div class="custom-control custom-radio custom-control-inline">

                                                <input type="radio" class="custom-control-input" id="customRadio7"

                                                    name="example1" [ngModel]="queryRadiobtn"

                                                    (ngModelChange)="queryRadiobtnEvent($event)" value="2radio">

                                                <label class="custom-control-label" for="customRadio7">consectetur

                                                    adipiscing</label>

                                            </div>

                                            <div class="custom-control custom-radio custom-control-inline">

                                                <input type="radio" class="custom-control-input" id="customRadio8"

                                                    name="example1" [ngModel]="queryRadiobtn"

                                                    (ngModelChange)="queryRadiobtnEvent($event)" value="3radio">

                                                <label class="custom-control-label" for="customRadio8">sed do eiusmod

                                                    tempor</label>

                                            </div>

                                            <div class="custom-control custom-radio custom-control-inline">

                                                <input type="radio" class="custom-control-input" id="customRadio9"

                                                    name="example1" [ngModel]="queryRadiobtn"

                                                    (ngModelChange)="queryRadiobtnEvent($event)" value="4radio">

                                                <label class="custom-control-label" for="customRadio9">incididunt ut

                                                    labore</label>

                                            </div>

                                            <div class="custom-control custom-radio custom-control-inline pb-3">

                                                <input type="radio" class="custom-control-input" id="customRadio10"

                                                    name="example1" [ngModel]="queryRadiobtn"

                                                    (ngModelChange)="queryRadiobtnEvent($event)" value="5radio">

                                                <label class="custom-control-label" for="customRadio10">et dolore magna

                                                    aliqua</label>

                                            </div>

                                        </form>

                                    </div>

                                </div>

                            </div>

                        </div>

                        <div class="h-divider"></div>

                        <div class="nav-ul-item p-3 sidenav-menus">

                            <div class="row">

                                <div class="col-md-12">

                                    <p class="side-heading">CALL ID</p>

                                    <div class="form-group has-search justify-content-center">

                                        <form [formGroup]="searchCallIDForm">

                                            <input type="text" class="form-control bg-text-box"

                                                placeholder="Search Call ID" formControlName="searchCallID">

                                            <div class="text-right"><img class="search-img icon-search-callid"

                                                    src="assets/search-icon.png"

                                                    (click)="onSearchCallIDSubmit(searchCallIDForm.value)">

                                            </div>

                                        </form>

                                    </div>

                                </div>

                            </div>

                        </div>

                        <!-- changes i have done -->

                        <div class="nav-ul-item p-3 sidenav-menus">

                            <!------Call Duration------>

                            <div class="row">

                                <div class="col-md-6">

                                    <label class="sidenav-label">Call Duration <= </label>

                                            <input class="form-control bg-text-box" type="text" placeholder="00:00:00"

                                                [ngModel]="callDurationLessThan"

                                                (ngModelChange)="callDurLessThanEvent($event)">

                                </div>

                                <div class="col-md-6">

                                    <label class="sidenav-label">Call Duration >=</label>

                                    <input class="form-control bg-text-box" type="text" placeholder="00:00:00"

                                        [ngModel]="callDurationGreaterThan"

                                        (ngModelChange)="callDurGreaterThanEvent($event)">

                                </div>

                            </div>

                            <div class="range-item mt-3">

                                <label class="sidenav-label">Overtalk</label>

                                <div class="row">

                                    <div class="col-md-7 mt-2">

                                        <input type="range" class="form-control-range" [ngModel]="overtalkRange"

                                            (ngModelChange)="overtalkRangeChangeEvent($event)">

                                    </div>

                                    <div class="col-md-5">

                                        <input class="form-control bg-text-box" type="text" placeholder="10.00"

                                            [ngModel]="overtalkRange"

                                            (ngModelChange)="overtalkRangeChangeEvent($event)">

                                    </div>

                                </div>

                            </div>

                            <div class="range-item">

                                <label class="sidenav-label">Call silence time</label>

                                <div class="row">

                                    <div class="col-md-7 mt-2">

                                        <input type="range" class="form-control-range" [ngModel]="callSilenceRange"

                                            (ngModelChange)="callSilenceRangeChangeEvent($event)">

                                    </div>

                                    <div class="col-md-5">

                                        <input class="form-control bg-text-box" type="text" placeholder="10.00"

                                            [ngModel]="callSilenceRange"

                                            (ngModelChange)="callSilenceRangeChangeEvent($event)">

                                    </div>

                                </div>

                            </div>

                            <!-- adding buttons -->

                            <!-- <div class="row">

                                <div class="col-md-6">

                                    <button type="reset" id="callIdResetbtn"

                                        class="btn btn-block btn-secondary">Reset</button>

                                </div>

                                <div class="col-md-6">

                                    <button type="submit" id="callIdSubmitbtn"

                                        class="btn btn-block btn-primary">Search</button>

                                </div>

                            </div> -->

                        </div>

                        <div class="h-divider"></div>

                    </div>

                </div>

            </nav>

        </div>

        <div id="layoutSidenav\_content">

            <main>

                <div class="container-fluid">

                    <p class="top-text"><span style="font-weight: 600;">DATE RANGE</span> {{dateRangeTextStart}} to

                        {{dateRangeTextEnd}}</p>

                    <div class="row pb-3">

                        <div class="col-md-12">

                            <!-------Tabs content Started-------->

                            <div class="card">

                                <div class="card-body tabs-card">

                                    <ul class="nav custom-tab-btn pl-4" role="tablist">

                                        <li class="nav-item">

                                            <a class="nav-link " href="#Call-Volume" role="tab" data-toggle="tab">Call

                                                Volume</a>

                                        </li>

                                        <li class="nav-item">

                                            <a class="nav-link" href="#Call-Duration" role="tab" data-toggle="tab">Call

                                                Duration</a>

                                        </li>

                                        <li class="nav-item">

                                            <a class="nav-link" href="#Sentiment" role="tab"

                                                data-toggle="tab">Sentiment</a>

                                        </li>

                                        <li class="nav-item">

                                            <a class="nav-link" href="#Talktime" role="tab"

                                                data-toggle="tab">Talktime</a>

                                        </li>

                                    </ul>

                                    <div class="tab-content pl-4">

                                        <br>

                                        <div role="tabpanel" class="tab-pane active" id="Call-Volume">

                                            <div class="row">

                                                <div class="col-md-2">

                                                    <p class="side-heading">CALL VOLUME</p>

                                                </div>

                                                <div class="col-md-8 text-center">

                                                    <p class="center-heading">Overall Call Volume - 123</p>

                                                </div>

                                            </div>

                                            <div id="chartdiv" style="height: 300px; width: 100%;"></div>

                                            <!-- <div id="callVolChartId" style="height: 180px; width: 100%;"></div> -->

                                        </div>

                                        <div role="tabpanel" class="tab-pane" id="Call-Duration">

                                            <div class="row">

                                                <div class="col-md-2">

                                                    <p class="side-heading">CALL Duration</p>

                                                </div>

                                                <div class="col-md-8 text-center">

                                                    <p class="center-heading">Overall Call Duration - 123</p>

                                                </div>

                                            </div>

                                            <div class="row">

                                                <div class="col-md-12">

                                                    <div id="chartdiv2" style="height: 300px; width: 100%;"></div>

                                                    <!-- <div id="callDurChartId" style="height: 180px; width: 100%;"></div> -->

                                                </div>

                                            </div>

                                        </div>

                                        <div role="tabpanel" class="tab-pane" id="Sentiment">

                                            <div class="row">

                                                <div class="col-md-2">

                                                    <p class="side-heading">Sentiment</p>

                                                </div>

                                                <div class="col-md-8 text-center">

                                                    <p class="center-heading">Overall Sentiments</p>

                                                </div>

                                            </div>

                                            <div id="chartdiv3" style="height: 300px; width: 100%;"></div>

                                            <!-- <div id="sentimentChartId" style="height: 180px; width: 100%;"></div> -->

                                        </div>

                                        <div role="tabpanel" class="tab-pane" id="Talktime">

                                            <div class="row">

                                                <div class="col-md-2">

                                                    <p class="side-heading">Talktime</p>

                                                </div>

                                                <div class="col-md-8 text-center">

                                                    <p class="center-heading">Overall Talktime- 123</p>

                                                </div>

                                            </div>

                                            <div id="chartdiv4" style="height: 300px; width: 100%;"></div>

                                            <!-- <div id="talktimeChartId" style="height: 180px; width: 100%;"></div> -->

                                        </div>

                                    </div>

                                </div>

                            </div>

                            <!------Tabs content Closed------>

                        </div>

                    </div>

                    <div class="row pb-3">

                        <div class="col-md-12">

                            <div class="card">

                                <div class="card-body">

                                    <div>

                                        <p class="side-heading">SEARCH PHRASE</p>

                                    </div>

                                    <form [formGroup]="searchPhrase" (ngSubmit)="onSearchPhrase()" class="form">

                                        <div class="row">

                                            <div class="col-md-6">

                                                <div class="form-group">

                                                    <input class="form-control bg-text-box" type="text"

                                                        placeholder="Enter text here to search"

                                                        formControlName="searchText">

                                                </div>

                                            </div>

                                            <div class="col-md-6">

                                                <select class="form-control select-custom"

                                                    formControlName="searchByOptions">

                                                    <option [ngValue]="null">Select Category</option>

                                                    <option value="agent">Agent</option>

                                                    <option value="customer">Customer</option>

                                                    <option value="both">Both</option>

                                                </select>

                                                <img class="select-icon-search-opt" style="width: 10px;"

                                                    src="assets/Dropdown-arrow.png">

                                            </div>

                                        </div>

                                        <div class="row">

                                            <div class="col-md-12">

                                                <div class="form-group">

                                                    <div formArrayName="ContactNos" class="mt-2">

                                                        <!-- <div class="formarrayscroll"> -->

                                                        <div \*ngFor="let contact of searchPhrase.get('ContactNos')['controls'];

                                                let ind = index">

                                                            <input type="text" class="form-control bg-text-box"

                                                                [formControlName]="ind" placeholder="PHRASE">

                                                            <button type="button" id="formarrayclose" class="close"

                                                                data-dismiss="modal"

                                                                (click)="remove(ind)">&times;</button>

                                                        </div>

                                                        <!-- </div> -->

                                                        <button class="add-phrase" (click)="addContactNo()">+ADD

                                                            PHRASE</button>

                                                    </div>

                                                </div>

                                            </div>

                                        </div>

                                        <div class="row">

                                            <div class="col-md-4 searchfields-align">

                                                <div class="form-group">

                                                    <select class="form-control select-custom"

                                                        formControlName="searchFiles">

                                                        <option [ngValue]="null">Find files that contain any of the

                                                            search terms</option>

                                                        <option value="2"> 2 </option>

                                                        <option value="3"> 3 </option>

                                                        <option value="4"> 4 </option>

                                                    </select>

                                                </div>

                                                <img class="select-icon-search1" style="width: 10px;"

                                                    src="assets/Dropdown-arrow.png">

                                            </div>

                                            <div class="col-md-3 mt-1">

                                                <label class="sidenav-label">Audio Time basis</label>

                                                <div class="form-group">

                                                    <select class="form-control select-custom"

                                                        formControlName="searchAudio">

                                                        <option [ngValue]="null">Anywhere in the file </option>

                                                        <option value="2"> 2 </option>

                                                        <option value="3"> 3 </option>

                                                        <option value="4"> 4 </option>

                                                    </select>

                                                </div>

                                                <img class="select-icon-search2" style="width: 10px;"

                                                    src="assets/Dropdown-arrow.png">

                                            </div>

                                            <div class="col-md-5 searchfields-align">

                                                <div class="d-flex  justify-content-between">

                                                    <div class="d-inputs">

                                                        <div class="input-group">

                                                            <div class="form-group">

                                                                <input type="text" class="form-control"

                                                                    formControlName="searchTime">

                                                            </div>

                                                            <div class="form-group">

                                                                <select class="form-control select-custom"

                                                                    formControlName="timeInType">

                                                                    <option [ngValue]="null">Select</option>

                                                                    <option value="secs">Sec</option>

                                                                    <option value="mins">Minutes</option>

                                                                    <option value="hours">Hours</option>

                                                                </select>

                                                                <img class="sec-icon" style="width: 10px;"

                                                                    src="assets/Dropdown-arrow.png">

                                                            </div>

                                                        </div>

                                                    </div>

                                                    <div class="d-inputs"><button type="submit"

                                                            class="btn btn-block btn-primary btn-search-row mb-2">Search</button>

                                                    </div>

                                                    <div class="d-inputs"><button type="reset"

                                                            class="btn btn-block btn-secondary btn-search-row mb-2">Reset</button>

                                                    </div>

                                                </div>

                                            </div>

                                        </div>

                                    </form>

                                </div>

                            </div>

                        </div>

                    </div>

                    <div class="row pb-3">

                        <div class="col-md-12">

                            <div class="card">

                                <div class="card-body tabs-card">

                                    <ul class="nav custom-tab-btn pl-4" role="tablist">

                                        <li class="nav-item">

                                            <a class="nav-link active" href="#tag" role="tab" data-toggle="tab">TAG</a>

                                        </li>

                                        <li class="nav-item">

                                            <a class="nav-link" href="#transcript" role="tab"

                                                data-toggle="tab">TRANSCRIPT</a>

                                        </li>

                                        <li class="nav-item">

                                            <a class="nav-link" href="#export" role="tab" data-toggle="tab">EXPORT</a>

                                        </li>

                                    </ul>

                                    <div class="tab-content pl-4">

                                        <br>

                                        <div role="tabpanel" class="tab-pane active" id="tag">

                                            <div id="forScroll">

                                                <table id="example" class="table table-striped table-bordered"

                                                    style="width:100%">

                                                    <thead>

                                                        <tr>

                                                            <th>CALL ID</th>

                                                            <th>CALL TRANSCRIPT ID</th>

                                                            <th>AGENT NAME</th>

                                                        </tr>

                                                    </thead>

                                                    <tbody

                                                        \*ngFor="let info of information[1]?.transcript\_data | paginate: { itemsPerPage: 50, currentPage: p }">

                                                        <tr>

                                                            <td>{{info.ticket\_no}}</td>

                                                            <td>System Architect</td>

                                                            <td>{{info.agent\_name}}</td>

                                                        </tr>

                                                    </tbody>

                                                </table>

                                            </div>

                                            <br>

                                            <pagination-controls (pageChange)="p = $event"></pagination-controls>

                                        </div>

                                        <div role="tabpanel" class="tab-pane" id="transcript">

                                            <h4>Pro</h4>

                                            <p>

                                                2. Tabs are helpful to hide or collapse some addtional content.

                                            </p>

                                        </div>

                                        <div role="tabpanel" class="tab-pane" id="export">

                                            <h4>Messages</h4>

                                            <p>

                                                3. You can really put whatever you want into the tab pane.

                                            </p>

                                        </div>

                                    </div>

                                </div>

                            </div>

                        </div>

                    </div>

                    <div class="row pb-3">

                        <div class="col-md-12">

                            <div class="card">

                                <div class="card-body">

                                    <p>No Item Selected</p>

                                    <div class="row">

                                        <!-- <div class="col-md-4">

                                            <img src="assets/Skip.png">

                                        </div> -->

                                        <div class="col-md-12">

                                            <audio controls="controls" ID="Player">

                                                <source

                                                    src="https://p.scdn.co/mp3-preview/6d00206e32194d15df329d4770e4fa1f2ced3f57#t=,1" />

                                            </audio>

                                        </div>

                                        <!-- <div class="col-md-4 text-right">

                                            <img src="assets/Speaker.png">

                                        </div> -->

                                    </div>

                                    <div class="row">

                                        <div class="col-md-12 text-center">

                                            <img class="mr-3" src="assets/back-10.png" onclick="PreviousAudio()">

                                            <img class="mr-3" src="assets/Play.png" onclick="stopsong()">

                                            <img class="mr-3" src="assets/jump-10.png" onclick="forwardAudio()">

                                        </div>

                                    </div>

                                </div>

                            </div>

                        </div>

                    </div>

                </div>

            </main>

        </div>

    </div>

    <!-- The Modal Add Folder -->

    <div class="modal" id="myModal">

        <div class="modal-dialog">

            <div class="modal-content">

                <!-- Modal Header -->

                <div class="modal-header">

                    <button type="button" class="close" data-dismiss="modal">&times;</button>

                    <div class="row pt-2 col">

                        <div class="col-md-6 col">

                            <p class="side-heading">CREATE NEW FOLDER</p>

                        </div>

                        <div class="col-md-6 col text-right">

                            <p class="modal-created">Created by John Doe</p>

                        </div>

                    </div>

                </div>

                <!-- Modal body -->

                <div class="modal-body">

                    <form [formGroup]="addNewFolder" (ngSubmit)="onSubmit()">

                        <div class="form-group">

                            <label for="formGroupExampleInput">Folder name</label>

                            <input type="text" class="form-control" placeholder="Click to enter"

                                formControlName="folderName">

                        </div>

                        <div class="form-group">

                            <label for="formGroupExampleInput2">Folder description</label>

                            <input type="text" class="form-control" placeholder="Click to enter"

                                formControlName="folderDescription">

                        </div>

                        <div class="form-group">

                            <label for="formGroupExampleInput3">Select a single audio file or a zip file containing

                                audio files</label>

                            <input type="file" class="form-control-file" formControlName="chooseFileForAudio">

                        </div>

                        <div class="form-group">

                            <label for="formGroupExampleInput3">Upload Meta Data</label>

                            <input type="file" class="form-control-file" formControlName="chooseFileForMetaData">

                        </div>

                        <div class="form-group">

                            <div class="form-check">

                                <input class="form-check-input" type="checkbox" formControlName="uploadLater" value=""

                                    id="uploadLater">

                                <label class="form-check-label" for="UploadLater">

                                    Upload Later

                                </label>

                            </div>

                        </div>

                        <div class="row mt-3">

                            <div class="col-md-4 col">

                                <div class="form-check">

                                    <input class="form-check-input" formControlName="textMasking" type="checkbox"

                                        value="" id="usepurifytext">

                                    <label class="form-check-label" for="testmasking">

                                        Use Masking text

                                    </label>

                                </div>

                            </div>

                            <div class="col-md-4 col">

                                <div class="form-check">

                                    <input class="form-check-input" formControlName="audioMasking" type="checkbox"

                                        value="" id="usepurifyaudio">

                                    <label class="form-check-label" for="audiomasking">

                                        Use Masking audio

                                    </label>

                                </div>

                            </div>

                            <div class="col-md-4">

                                <button type="submit" class="btn btn-success">Create</button>

                            </div>

                        </div>

                    </form>

                </div>

            </div>

        </div>

    </div>

    <!-- The Modal Add Query -->

    <div class="modal" id="myModal1">

        <div class="modal-dialog">

            <div class="modal-content">

                <!-- Modal Header -->

                <div class="modal-header">

                    <button type="button" class="close" data-dismiss="modal">&times;</button>

                    <div class="row pt-2 col">

                        <div class="col-md-6 col">

                            <p class="side-heading">CREATE NEW APPLICATION</p>

                        </div>

                        <div class="col-md-6 col text-right">

                            <p class="modal-created">Created by John Doe</p>

                        </div>

                    </div>

                </div>

                <!-- Modal body -->

                <div class="modal-body">

                    <form>

                        <div class="form-group">

                            <div class="row">

                                <div class="col-md-6">

                                    <label for="formGroupExampleInput">Query name</label>

                                </div>

                                <div class="col-md-6 text-right">

                                    <a href="#" class="view-query">View all queries</a>

                                </div>

                            </div>

                            <input type="text" class="form-control" id="formGroupExampleInput"

                                placeholder="Click to enter">

                        </div>

                        <div class="form-group">

                            <label for="formGroupExampleInput2">Query description</label>

                            <textarea class="form-control" rows="5" id="formGroupExampleInput2" name="text"

                                placeholder="Click to enter"></textarea>

                        </div>

                        <div class="form-group">

                            <label for="formGroupExampleInput3">Query phrase</label>

                            <input type="text" class="form-control" id="formGroupExampleInput3"

                                placeholder="Click to enter">

                        </div>

                        <div class="text-right">

                            <button type="button" class="btn btn-success">Create</button>

                        </div>

                    </form>

                </div>

            </div>

        </div>

    </div>

</body>

**Css**

\* {

    -webkit-box-sizing: border-box;

    -moz-box-sizing: border-box;

    box-sizing: border-box

  }

  .loading {

    width: 200px;

    height: 50px;

    margin: 20px auto;

    position: relative;

  }

  .loading span {

    display: block;

    position: absolute;

    bottom: 0;

    height: 5px;

    width: 9px;

    background-color: #9b59B6;

    -webkit-animation: waving 1.5s infinite ease-in-out;

    -moz-animation: waving 1.5s infinite ease-in-out;

    -o-animation: waving 1.5s infinite ease-in-out;

    animation: waving 1.5s infinite ease-in-out

  }

  .loading span:nth-of-type(2) {

    left: 11px;

    -webkit-animation-delay: .2s;

    -moz-animation-delay: .2s;

    -o-animation-delay: .2s;

    animation-delay: .2s

  }

  .loading span:nth-of-type(3) {

    left: 22px;

    -webkit-animation-delay: .4s;

    -moz-animation-delay: .4s;

    -o-animation-delay: .4s;

    animation-delay: .4s

  }

  .loading span:nth-of-type(4) {

    left: 33px;

    -webkit-animation-delay: .6s;

    -moz-animation-delay: .6s;

    -o-animation-delay: .6s;

    animation-delay: .6s

  }

  .loading span:last-of-type {

    left: 44px;

    -webkit-animation-delay: .8s;

    -moz-animation-delay: .8s;

    -o-animation-delay: .8s;

    animation-delay: .8s

  }

  @-webkit-keyframes waving {

    0%, 50%, 100% {

      height: 5px;

      -webkit-transform: translateY(0);

      -moz-transform: translateY(0);

      -o-transform: translateY(0);

      transform: translateY(0);

      background-color: #9B59B6

    }

    25% {

      height: 30px;

      -webkit-transform: translateY(15px);

      -moz-transform: translateY(15px);

      -o-transform: translateY(15px);

      transform: translateY(15px);

      background-color: #3498D6

    }

  }

  @-moz-keyframes waving {

    0%, 50%, 100% {

      height: 5px;

      -webkit-transform: translateY(0);

      -moz-transform: translateY(0);

      -o-transform: translateY(0);

      transform: translateY(0);

      background-color: #9B59B6

    }

    25% {

      height: 30px;

      -webkit-transform: translateY(15px);

      -moz-transform: translateY(15px);

      -o-transform: translateY(15px);

      transform: translateY(15px);

      background-color: #3498D6

    }

  }

  @-o-keyframes waving {

    0%, 50%, 100% {

      height: 5px;

      -webkit-transform: translateY(0);

      -moz-transform: translateY(0);

      -o-transform: translateY(0);

      transform: translateY(0);

      background-color: #9B59B6

    }

    25% {

      height: 30px;

      -webkit-transform: translateY(15px);

      -moz-transform: translateY(15px);

      -o-transform: translateY(15px);

      transform: translateY(15px);

      background-color: #3498D6

    }

  }

  @keyframes waving {

    0%, 50%, 100% {

      height: 5px;

      -webkit-transform: translateY(0);

      -moz-transform: translateY(0);

      -o-transform: translateY(0);

      transform: translateY(0);

      background-color: #9B59B6

    }

    25% {

      height: 30px;

      -webkit-transform: translateY(15px);

      -moz-transform: translateY(15px);

      -o-transform: translateY(15px);

      transform: translateY(15px);

      background-color: #3498D6

    }

  }

button#formarrayclose {

    margin-top: -42px;

}

.formarrayscroll {

    height: 82px;

    overflow-y: hidden;

}

audio#Player {

    width: 100%;

}

.col-scroll {

    height: 150px;

    position: relative;

    top: 0px;

    bottom: 0;

    left: 0;

    width: 100%;

    overflow-y: scroll;

}

div#forScroll {

    height: 400px;

    overflow-y: scroll;

}

#layoutSidenav {

    display: flex;

    background: #f2f2f2;

}

input#datepick {

    width: 100%;

    padding: 6px 4px;

    background-color: #f9faff;

    border: 1px solid #e0e7ff;

    border-radius: 3px;

    font-size: 13px;

    height: 38px;

}

button.add-phrase {

    background: none;

    border: none;

    font-size: 13px;

    margin-top: 10px;

    font-weight: 700;

    color: #0f75bc;

    margin-bottom: 0px;

}

img.search-img.icon-search-query {

    margin-top: 0px;

}

#layoutSidenav #layoutSidenav\_nav {

    flex-basis: 275px;

    flex-shrink: 0;

    transition: transform 0.15s ease-in-out;

  /\*z-index: 1038;\*/

    transform: translateX(-225px);

}

#layoutSidenav #layoutSidenav\_content {

    position: relative;

    display: flex;

    flex-direction: column;

    justify-content: space-between;

    min-width: 0;

    flex-grow: 1;

    min-height: calc(100vh - 56px);

    margin-left: -225px;

}

.sb-sidenav-toggled #layoutSidenav #layoutSidenav\_nav {

    transform: translateX(0);

}

.sb-sidenav-toggled #layoutSidenav #layoutSidenav\_content:before {

    content: "";

    display: block;

    position: absolute;

    top: 0;

    left: 0;

    width: 100%;

    height: 100%;

    background: #000;

    z-index: 1037;

    opacity: 0.5;

    transition: opacity 0.3s ease-in-out;

}

@media (min-width: 992px) {

    #layoutSidenav #layoutSidenav\_nav {

        transform: translateX(0);

    }

    #layoutSidenav #layoutSidenav\_content {

        margin-left: 0;

        transition: margin 0.15s ease-in-out;

    }

    .sb-sidenav-toggled #layoutSidenav #layoutSidenav\_nav {

        transform: translateX(-225px);

    }

    .sb-sidenav-toggled #layoutSidenav #layoutSidenav\_content {

        margin-left: -225px;

    }

    .sb-sidenav-toggled #layoutSidenav #layoutSidenav\_content:before {

        display: none;

    }

}

.sb-nav-fixed .sb-topnav {

    z-index: 1039;

}

.sb-nav-fixed #layoutSidenav #layoutSidenav\_nav {

    width: 285px;

 /\* height: 100vh;

  z-index: 1038;\*/;

}

.sb-nav-fixed #layoutSidenav #layoutSidenav\_nav .sb-sidenav {

    padding-top: 0px;

}

.sb-nav-fixed #layoutSidenav #layoutSidenav\_nav .sb-sidenav .sb-sidenav-menu {

    overflow-y: auto;

}

.sb-nav-fixed #layoutSidenav #layoutSidenav\_content {

    padding-left: 0px;

    top: 0px;

}

.sb-sidenav {

    display: flex;

    flex-direction: column;

    height: 95%;

    flex-wrap: nowrap;

}

.sb-sidenav .sb-sidenav-menu {

    flex-grow: 1;

}

.sb-sidenav .sb-sidenav-menu .nav {

    flex-direction: column;

    flex-wrap: nowrap;

}

.sb-sidenav .sb-sidenav-menu .nav .sb-sidenav-menu-heading {

    padding: 1.75rem 1rem 0.75rem;

    font-size: 0.75rem;

    font-weight: bold;

    text-transform: uppercase;

}

.sb-sidenav .sb-sidenav-menu .nav .nav-link {

    display: flex;

    align-items: center;

    padding-top: 0.75rem;

    padding-bottom: 0.75rem;

    position: relative;

}

.sb-sidenav .sb-sidenav-menu .nav .nav-link .sb-nav-link-icon {

    font-size: 0.9rem;

}

.sb-sidenav .sb-sidenav-menu .nav .nav-link .sb-sidenav-collapse-arrow {

    display: inline-block;

    margin-left: auto;

    transition: transform 0.15s ease;

}

.sb-sidenav .sb-sidenav-menu .nav .nav-link.collapsed .sb-sidenav-collapse-arrow {

    transform: rotate(-90deg);

}

.sb-sidenav .sb-sidenav-menu .nav .sb-sidenav-menu-nested {

    margin-left: 1.5rem;

    flex-direction: column;

}

.sb-sidenav .sb-sidenav-footer {

    padding: 0.75rem;

    flex-shrink: 0;

}

.sb-sidenav-dark {

    background-color: #000000;

    color: rgba(255, 255, 255, 0.5);

}

.sb-sidenav-dark .sb-sidenav-menu .sb-sidenav-menu-heading {

    color: #000000;

}

.sb-sidenav-dark .sb-sidenav-menu .nav-link {

    color: rgba(255, 255, 255, 0.5);

}

.sb-sidenav-dark .sb-sidenav-menu .nav-link .sb-nav-link-icon {

    color: rgba(255, 255, 255, 0.25);

}

.sb-sidenav-dark .sb-sidenav-menu .nav-link .sb-sidenav-collapse-arrow {

    color: rgba(255, 255, 255, 0.25);

}

.sb-sidenav-dark .sb-sidenav-menu .nav-link:hover {

    color: #333333;

}

.sb-sidenav-dark .sb-sidenav-menu .nav-link.active {

    color: #000;

}

.sb-sidenav-dark .sb-sidenav-menu .nav-link.active .sb-nav-link-icon {

    color: #000;

}

.sb-sidenav-dark .sb-sidenav-footer {

    background-color: #343a40;

}

.sb-sidenav-light {

    background-color: #ffffff;

    color: #212529;

}

.sb-sidenav-light .sb-sidenav-menu .sb-sidenav-menu-heading {

    color: #adb5bd;

}

.sb-sidenav-light .sb-sidenav-menu .nav-link {

    color: #212529;

}

.sb-sidenav-light .sb-sidenav-menu .nav-link .sb-nav-link-icon {

    color: #adb5bd;

}

.sb-sidenav-light .sb-sidenav-menu .nav-link .sb-sidenav-collapse-arrow {

    color: #adb5bd;

}

.sb-sidenav-light .sb-sidenav-menu .nav-link:hover {

    color: #007bff;

}

.sb-sidenav-light .sb-sidenav-menu .nav-link.active {

    color: #007bff;

}

.sb-sidenav-light .sb-sidenav-menu .nav-link.active .sb-nav-link-icon {

    color: #007bff;

}

.sb-sidenav-light .sb-sidenav-footer {

    background-color: #e9ecef;

}

/\*\*\*\*\*\*\*\*\*13-March-2020\*\*\*\*\*\*\*\*\*\*/

body {

    background: #f2f2f2;

    background-repeat: no-repeat;

    background-size: cover;

    background-attachment: fixed;

    height: 100%;

    font-size: 13px;

    font-family: 'Open Sans', sans-serif;

}

li.nav-item.mt-2 {

    font-family: 'Open Sans', sans-serif;

    padding: 0px 25px;

    font-weight: 600;

}

.nav-custom-link.active {

    color: #1d5574 !important;

}

a.nav-custom-link {

    color: #b0bac9 !important;

}

a.dropdown-item {

    font-size: 13px;

}

a#navbarDropdown {

    margin-top: 8px;

}

img.user-img {

    width: 40px;

}

.sticky-top {

    padding-right: 50px;

    padding-left: 50px;

    background-color: #ffffff !important;

}

div#layoutSidenav\_nav {

    padding: 0px 0px 0px 12px;

}

.d-inputs {

    width: 30%;

}

.sb-sidenav {

    border-radius: 4px;

}

.form-control:focus {

    color: #495057;

    background-color: #fff;

    border-color: none !important;

    outline: 0;

    box-shadow: none !important;

}

/\*.row.sidebar-heading {

    border-top: 1px solid #b0bac9;

    border-bottom: 1px solid #b0bac9;

    padding-top: 6px;

}\*/

.has-search .form-control-feedback {

    position: absolute;

    z-index: 2;

    display: block;

    width: 2.375rem;

    height: 2.375rem;

    line-height: 2.375rem;

    text-align: center;

    pointer-events: none;

    color: #aaa;

}

.nav-ul-item P {

    font-size: 12px;

    font-weight: bold;

    margin-bottom: 6px;

}

.h-divider {

    margin-top: 5px;

    margin-bottom: 5px;

    height: 1px;

    width: 100%;

    border-top: 1px solid #e0e7ff;

}

.datepicker {

    font-size: 0.875em;

}

.datepicker td, .datepicker th {

    width: 1.5em;

    height: 1.5em;

}

h6.side-heading {

    font-size: 12px;

    font-weight: 700;

}

p.side-heading {

    font-size: 12px;

    font-weight: 800;

    margin-top: 4px;

}

.search-img {

    margin-top: -5px;

}

p.top-text {

    margin: 1rem 0 10px 0;

    color: #000000;

    font-weight: 500;

    font-size: 12px;

}

button.btn-primary {

    background-color: #0f75bc!important;

}

.custom-tab-btn {

    font-size: 14px;

    font-weight: 700;

    /\*text-decoration: underline;\*/

    border-bottom: 1px solid #b0bac9;

}

.custom-tab-btn a.nav-link {

    color: #0f75bc;

    padding-left: 5px;

}

button#sidebarToggle {

    display: none;

}

span.navbar-toggler-icon {

    display: none;

}

.icon-search-folder {

    position: absolute;

    top: 100px;

    right: 16px;

}

.icon-search-query {

    position: absolute;

    top: 468px;

    right: 16px;

}

form.form-horizontal {

    width: 100%;

}

input#inputSuccess {

    border-left: 0px;

    border-right: 0px;

    border-radius: 0px;

}

p.add-phrase {

    font-size: 13px;

    font-weight: 700;

    color: #0f75bc;

    margin-bottom: 0px;

}

p.form-sec {

    font-size: 16px;

    font-weight: 600;

    margin-top: 6px;

}

.sticky-top {

    z-index: 99999 !important;

}

.modal {

    top: 26px !important;

}

button.modal-btn {

    background: none;

    border: none;

    font-size: 12px;

    padding: 0px;

    margin-top: 4px;

    font-weight: 600;

    margin-bottom: 6px;

    color: #0f75bc;

}

.modal-created{

    font-size: 13px;

    font-weight: 600;

}

.modal-header {

    display: inline-block !important;

    padding: 18px 1rem 0px !important;

}

.modal-body .form-control {

    background-color: #f9fafe;

    font-size: 14px;

    height: 38px;

}

.modal-body .form-control-file {

    background-color: #f9fafe;

    border: 1px solid #ced4da;

    border-radius: .25rem;

    height: 38px;

    padding-top: 5px;

}

.modal-body button.btn-success {

    background-color: #0f75bd;

    border-color: #0f75bd;

}

.modal-body .btn-success:hover {

    color: #fff;

    background-color: #0f75bd;

    border-color: #0f75bd;

}

.modal-header p.side-heading {

    font-size: 13px;

    font-weight: 700;

    margin-top: 4px;

}

.modal-body label {

    font-weight: 500;

    font-size: 14px;

    margin-top: 10px;

}

.modal-body .form-check-input {

    margin-top: 15px;

}

.modal-body {

    padding: 1rem 2rem 2rem !important;

}

.searchfields-align {

    margin-top: 32px;

}

.tabs-card {

    padding: 1.25rem 0px;

}

a.view-query {

    color: #0f75bc;

    font-weight: 700;

    font-size: 14px;

}

.bg-text-box{

    background-color: #f9faff;

    border: 1px solid #e0e7ff;

    border-radius: 3px;

    font-size: 13px;

    height: 38px;

}

img.icon-calendar {

    position: absolute;

    top: 45px;

    right: 25px;

}

.input-date{

    width: 100%;

    padding: 6px 4px;

}

div#example\_length {

    display: none;

}

div#example\_filter {

    display: none;

}

th.sorting\_asc {

    border: none;

    font-size: 12px;

    font-weight: 800;

}

table#example {

    border: none;

}

th.sorting {

    border: none;

    font-size: 12px;

    font-weight: 800;

}

td.sorting\_1 {

    border: none;

}

tr.odd {

    background-color: #ffffff !important;

}

.table-bordered td, .table-bordered th {

    border: none;

}

.table td {

    padding: 5px;

}

.table th {

    padding: 10px 5px;

}

.table thead th {

    border-bottom: none;

}

.table-bordered {

    border: none;

}

input.custom {

    font-size: 12px;

    font-weight: 700;

    color: #000000 !important;

    border-color: #e0e7ff;

    padding: 4px;

}

label.sidenav-label {

    font-size: 12px;

    font-weight: 700;

}

p.center-heading {

    background-color: #c6f5e5;

    width: 30%;

    position: absolute;

    left: 255px;

    font-weight: 600;

    padding: 3px 0px;

}

.form-group {

    margin-bottom: 10px;

}

.custom-control-label {

    position: relative;

    margin-bottom: 5px;

    vertical-align: top;

    margin-top: 5px;

}

.select-custom{

    font-size: 13px;

    height: 38px;

}

.btn-custom{

    border: 1px solid #e0e7ff;

    font-size: 13px;

}

.btn-search-row {

    font-size: 15px;

    height: 38px;

    font-weight: 500;

}

.btn-outline-secondary:hover {

    background-color: #0f75bc;

    border-color: #0f75bc;

}

.has-feedback{

    margin-left: 12px;

    margin-right: 12px;

}

.card{

    border: none;

}

select.cll-volume-select {

    font-size: 13px;

    border: none;

}

.dropdown-toggle::after {

    display: none;

}

select {

  -webkit-appearance: none;

  -moz-appearance: none;

}

img.select-icon-volume {

    position: relative;

    top: -26px;

    right: 40px;

}

img.select-icon-search1 {

    position: relative;

    top: -38px;

    right: -180px;

}

img.select-icon-search2 {

    position: relative;

    top: -38px;

    right: -122px;

}

img.sec-icon {

    position: absolute;

    right: 10px;

    top: 65px;

}

img.user-icon {

    position: relative;

    right: -4px;

}

.icon-search-callid {

    position: absolute;

    top: 45px;

    right: 28px;

}

.tabs-card li.nav-item {

    margin-right: 18px;

}

div#chartContainer, div#sentiment, div#talktime {

    margin-top: 35px;

}

p.center-heading-tabs {

    background-color: #c6f5e5;

    width: 30%;

    position: absolute;

    left: 355px;

    font-weight: 600;

    padding: 3px 0px;

}

/\*\*\*\*Mobile Media Query\*\*\*\*/

@media (max-width:768px) {

    button#sidebarToggle {

        display: block;

    }

    .sb-nav-fixed #layoutSidenav #layoutSidenav\_nav {

        z-index: 1038;

    }

    span.navbar-toggler-icon {

        display: block;

    }

    #layoutSidenav #layoutSidenav\_nav {

        flex-basis: 268px;

    }

    .icon-search-folder {

        top: 96px;

    }

    .icon-search-query {

        top: 512px;

    }

    .layoutSidenav\_nav\_mob {

    z-index: 0;

    }

    #layoutSidenav #layoutSidenav\_content {

    margin-left: -222px;

    }

    .modal-header p.side-heading {

    font-size: 11px;

    }

    .modal-created {

    font-size: 12px;

    }

    .modal-body button.btn-success {

    margin-top: 12px;

    }

    .modal-header .close {

    padding: 5px 1rem;

    }

    .modal-header {

    padding: 18px 1rem 0px !important;

    }

    .d-inputs {

    width: 30%;

    }

    .sec-d-input {

    width: 12%;

    }

}

img.select-icon-search-opt {

    position: absolute;

    top: 18px;

    right: 30px;

}

**Ts**

import { Component, OnInit, NgZone } from '@angular/core';

import \* as CanvasJS from 'src/assets/canvasjs.min.js';

import { FormGroup, FormBuilder, FormArray, FormControl } from '@angular/forms';

import \* as am4core from "@amcharts/amcharts4/core";

import \* as am4charts from "@amcharts/amcharts4/charts";

import am4themes\_animated from "@amcharts/amcharts4/themes/animated";

import am4themes\_material from "@amcharts/amcharts4/themes/material";

import { CiOverviewServiceService } from '../services/ci-overview-service.service';

am4core.useTheme(am4themes\_animated);

@Component({

  selector: 'app-ci-overview',

  templateUrl: './ci-overview.component.html',

  styleUrls: ['./ci-overview.component.css']

})

export class CIOverviewComponent implements OnInit {

  p: number = 1;

  private chart: am4charts.XYChart;

  constructor(private formBuilder: FormBuilder, private zone: NgZone, private cioverviewser: CiOverviewServiceService) { }

  addNewFolder: FormGroup;

  submitted = false;

  searchPhrase: FormGroup;

  searchFolderForm: FormGroup;

  searchQueryForm: FormGroup;

  searchCallIDForm: FormGroup;

  dateRangeTextStart: any;

  dateRangeTextEnd: any

  public information: any = [];

  public callDuration: Array<any> = [];

  public callVolume: Array<any> = [];

  public callTalktime: Array<any> = [];

  // date range picker start

  public daterange: any = {};

  public options: any = {

    locale: { format: 'DD-MM-YYYY' },

    alwaysShowCalendars: false,

  };

  public selectedDate(value: any, datepicker?: any) {

    // console.log(value);

    datepicker.start = value.start;

    datepicker.end = value.end;

    this.daterange.start = value.start;

    this.daterange.end = value.end;

    this.daterange.label = value.label;

    this.dateRangeTextStart = value.start.format('DD-MM-YYYY');

    this.dateRangeTextEnd = value.end.format('DD-MM-YYYY');

    console.log("start date range :-  ", value.start.format('DD-MM-YYYY'))

    console.log("end date range :-  ", value.end.format('DD-MM-YYYY'))

  }

  // date range picker end

  ngOnInit() {

    // addNewFolder form

    this.addNewFolder = this.formBuilder.group({

      folderName: [],

      folderDescription: [],

      chooseFileForAudio: [],

      chooseFileForMetaData: [],

      uploadLater: [],

      textMasking: [],

      audioMasking: []

    });

    // searchPharse form

    this.searchPhrase = this.formBuilder.group({

      ContactNos: this.formBuilder.array([]),

      searchText: [],

      searchByOptions: [],

      searchFiles: [],

      searchAudio: [],

      searchTime: [],

      timeInType: []

    });

    // searchFolder form

    this.searchFolderForm = this.formBuilder.group({

      searchFolder: []

    });

    // searchQuery form

    this.searchQueryForm = this.formBuilder.group({

      searchQuery: []

    });

    // searchCallID form

    this.searchCallIDForm = this.formBuilder.group({

      searchCallID: []

    });

    this.cioverviewser.initalApiCall().subscribe( data  => {

      // console.log(data[0].call\_graph\_data.call\_duration\_data);

      this.information = data;

      console.log(this.information);

      this.callDuration = data[0].call\_graph\_data.call\_duration\_data;

      console.log(this.callDuration);

      this.callVolume = data[0].call\_graph\_data.call\_volume\_data;

      console.log(this.callVolume);

      this.callTalktime = data[0].call\_graph\_data.call\_talktime\_data;

      console.log(this.callTalktime);

    })

  }

  ngAfterViewInit() {

    this.callVolumeChartFun();

    this.callDurationChartFun();

    this.sentimentChartFun();

    this.talktimeChartFun();

  }

  ngOnDestroy() {

    this.zone.runOutsideAngular(() => {

      if (this.chart) {

        this.chart.dispose();

      }

    });

  }

  // addnewfolder form function submit start

  onSubmit() {

    this.submitted = true;

    console.log(this.addNewFolder.value);

    this.cioverviewser.addfolderfun(this.addNewFolder.value);

  }

  // addNewFolder form function submit end

  // searchPharse form function reset and submit start

  get fControl(): FormControl {

    return new FormControl()

  }

  addContactNo() {

    (this.searchPhrase.get("ContactNos") as FormArray).push(new FormControl);

  }

  onSearchPhrase() {

    console.log(this.searchPhrase.value);

  }

  remove(index) {

    (this.searchPhrase.get("ContactNos") as FormArray).removeAt(index);

  }

  // searchPharse form function reset and submit end

  // search folder

  onSearchFolderSubmit(value) {

    console.log("search Folder input :-  ", value);

    this.cioverviewser.searchfolderfun(value)

  }

  onSearchQuerySubmit(value) {

    console.log("search Queryr input :-  ", value);

  }

  onSearchCallIDSubmit(value) {

    console.log("search CallID input :-  ", value);

    this.cioverviewser.callIDsearchfun(value)

  }

  //-------------------------------------filter events--------------------------------//

  private callSilenceRange;

  // private callSilenceInput;

  callSilenceRangeChangeEvent(event) {

    this.callSilenceRange = event;

    console.log('Call Silence Range :-  ', this.callSilenceRange);

    this.cioverviewser.callsilenceserfun(this.callSilenceRange)

  }

  callSilenceInputChangeEvent(event) {

    this.callSilenceRange = event;

    console.log('Call Silence Input :-  ', this.callSilenceRange);

    this.cioverviewser.callsilenceserfun(this.callSilenceRange)

  }

  private overtalkRange;

  // private overtalkInput;

  overtalkRangeChangeEvent(event) {

    this.overtalkRange = event;

    console.log('Overtalk Range :-  ', this.overtalkRange);

    this.cioverviewser.overtalkserfun(this.overtalkRange)

  }

  overtalkInputChangeEvent(event) {

    this.overtalkRange = event;

    console.log('Overtalk Input :-  ', this.overtalkRange);

    this.cioverviewser.overtalkserfun(this.overtalkRange)

  }

  private callDurationLessThan;

  private callDurationGreaterThan;

  callDurLessThanEvent(event) {

    this.callDurationLessThan = event;

    console.log('Call Duration Less than :-  ', this.callDurationLessThan);

    this.cioverviewser.callDurLesfun(this.callDurationLessThan)

  }

  callDurGreaterThanEvent(event) {

    this.callDurationGreaterThan = event;

    console.log('Call Duration Greater than :-  ', this.callDurationGreaterThan);

    this.cioverviewser.callDurGreatfun(this.callDurationGreaterThan)

  }

  private folderRadiobtn;

  folderRadiobtnEvent(event) {

    this.folderRadiobtn = event;

    console.log('Folder radio button select :-  ', this.folderRadiobtn);

    this.cioverviewser.selectfolderfun(this.folderRadiobtn)

  }

  private queryRadiobtn;

  queryRadiobtnEvent(event) {

    this.queryRadiobtn = event;

    console.log('Query radio button select :-  ', this.queryRadiobtn);

  }

  //------------------------------------graphs function-----------------------------//

  callVolumeChartFun() {

    var chart = am4core.create("chartdiv", am4charts.XYChart);

    // Add data

    chart.data = [

      {

        "\_id": "2020-01-22",

        "call\_volume": 423

      },

      {

        "\_id": "2020-01-23",

        "call\_volume": 118

      },

      {

        "\_id": "2020-01-24",

        "call\_volume": 18

      },

      {

        "\_id": "2020-01-27",

        "call\_volume": 2

      }

    ]

    // Create axes

    var categoryAxis = chart.xAxes.push(new am4charts.CategoryAxis());

    categoryAxis.dataFields.category = "\_id";

    categoryAxis.renderer.grid.template.location = 0;

    categoryAxis.renderer.minGridDistance = 30;

    var valueAxis = chart.yAxes.push(new am4charts.ValueAxis());

    // Create series

    var series = chart.series.push(new am4charts.ColumnSeries());

    series.dataFields.valueY = "call\_volume";

    series.dataFields.categoryX = "\_id";

    series.name = "call\_volume";

    // color change

    series.stroke = am4core.color("#0f75bc");

    series.columns.template.fill = am4core.color("#0f75bc");

    series.columns.template.tooltipText = "{categoryX}: [bold]{valueY}[/]";

    series.columns.template.fillOpacity = .8;

    series.columns.template.width = 5;

    var columnTemplate = series.columns.template;

    columnTemplate.strokeWidth = 2;

    columnTemplate.strokeOpacity = 1;

  }

  callDurationChartFun() {

    var chart = am4core.create("chartdiv2", am4charts.XYChart);

    // Add data

    chart.data = [{

      "\_id": "2020-01-22",

      "call\_duration": 46953

    },

    {

      "\_id": "2020-01-23",

      "call\_duration": 13098

    },

    {

      "\_id": "2020-01-24",

      "call\_duration": 1998

    },

    {

      "\_id": "2020-01-27",

      "call\_duration": 222

    }];

    // Create axes

    var categoryAxis = chart.xAxes.push(new am4charts.CategoryAxis());

    categoryAxis.dataFields.category = "\_id";

    categoryAxis.renderer.grid.template.location = 0;

    categoryAxis.renderer.minGridDistance = 30;

    var valueAxis = chart.yAxes.push(new am4charts.ValueAxis());

    // Create series

    var series = chart.series.push(new am4charts.ColumnSeries());

    series.dataFields.valueY = "call\_duration";

    series.dataFields.categoryX = "\_id";

    series.name = "call\_duration";

    series.columns.template.tooltipText = "{categoryX}: [bold]{valueY}[/]";

    series.columns.template.fillOpacity = .8;

    series.stroke = am4core.color("#0f75bc");

    series.columns.template.fill = am4core.color("#0f75bc");

    series.columns.template.tooltipText = "{categoryX}: [bold]{valueY}[/]";

    series.columns.template.fillOpacity = .8;

    series.columns.template.width = 5;

    var columnTemplate = series.columns.template;

    columnTemplate.strokeWidth = 2;

    columnTemplate.strokeOpacity = 1;

  }

  sentimentChartFun() {

    var chart = am4core.create("chartdiv3", am4charts.XYChart);

    // Add data

    chart.data = [{

      "age": "85+",

      "positive": 0.3

    }, {

      "age": "80-54",

      "negative": -0.2,

      "positive": 0.3

    }, {

      "age": "75-79",

      "negative": -0.3,

    }, {

      "age": "70-74",

      "negative": -0.5,

    }, {

      "age": "65-69",

      "positive": 1.0

    }, {

      "age": "60-64",

      "negative": -1.1,

      "positive": 1.3

    }, {

      "age": "55-59",

      "negative": -1.7,

      "positive": 1.9

    }, {

      "age": "50-54",

      "negative": -2.2,

      "positive": 2.5

    }, {

      "age": "45-49",

      "positive": 3.0

    }];

    // Use only absolute numbers

    chart.numberFormatter.numberFormat = "#.#s";

    // Create axes

    var categoryAxis = chart.yAxes.push(new am4charts.CategoryAxis());

    categoryAxis.dataFields.category = "age";

    categoryAxis.renderer.grid.template.location = 0;

    categoryAxis.renderer.inversed = true;

    var valueAxis = chart.xAxes.push(new am4charts.ValueAxis());

    valueAxis.extraMin = 0.1;

    valueAxis.extraMax = 0.1;

    valueAxis.renderer.minGridDistance = 40;

    valueAxis.renderer.ticks.template.length = 5;

    valueAxis.renderer.ticks.template.disabled = false;

    valueAxis.renderer.ticks.template.strokeOpacity = 0.4;

    valueAxis.renderer.labels.template.adapter.add("text", function (text) {

      return text == "negative" || text == "positive" ? text : text + "%";

    })

    // Create series

    var negative = chart.series.push(new am4charts.ColumnSeries());

    negative.dataFields.valueX = "negative";

    negative.dataFields.categoryY = "age";

    negative.clustered = false;

    negative.stroke = am4core.color("#e20d2a");

    negative.columns.template.fill = am4core.color("#e20d2a");

    negative.columns.template.width = 5;

    var negativeLabel = negative.bullets.push(new am4charts.LabelBullet());

    negativeLabel.label.text = "{valueX}%";

    negativeLabel.label.hideOversized = false;

    negativeLabel.label.truncate = false;

    negativeLabel.label.horizontalCenter = "right";

    negativeLabel.label.dx = -10;

    var positive = chart.series.push(new am4charts.ColumnSeries());

    positive.dataFields.valueX = "positive";

    positive.dataFields.categoryY = "age";

    positive.clustered = false;

    positive.stroke = am4core.color("#15ab65");

    positive.columns.template.fill = am4core.color("#15ab65");

    positive.columns.template.width = 5;

    var positiveLabel = positive.bullets.push(new am4charts.LabelBullet());

    positiveLabel.label.text = "{valueX}%";

    positiveLabel.label.hideOversized = false;

    positiveLabel.label.truncate = false;

    positiveLabel.label.horizontalCenter = "left";

    positiveLabel.label.dx = 10;

    var negativeRange = valueAxis.axisRanges.create();

    negativeRange.value = -10;

    negativeRange.endValue = 0;

    negativeRange.label.text = "negative";

    negativeRange.label.fill = chart.colors.list[0];

    negativeRange.label.dy = 20;

    negativeRange.label.fontWeight = '600';

    negativeRange.grid.strokeOpacity = 1;

    negativeRange.grid.stroke = negative.stroke;

    var positiveRange = valueAxis.axisRanges.create();

    positiveRange.value = 0;

    positiveRange.endValue = 10;

    positiveRange.label.text = "positive";

    positiveRange.label.fill = chart.colors.list[1];

    positiveRange.label.dy = 20;

    positiveRange.label.fontWeight = '600';

    positiveRange.grid.strokeOpacity = 1;

    positiveRange.grid.stroke = positive.stroke;

  }

  talktimeChartFun() {

    var chart = am4core.create("chartdiv4", am4charts.XYChart);

    // Add data

    chart.data = [{

      "\_id": "2020-01-22",

      "call\_talktime": 21573

    },

    {

      "\_id": "2020-01-23",

      "call\_talktime": 6018

    },

    {

      "\_id": "2020-01-24",

      "call\_talktime": 918

    },

    {

      "\_id": "2020-01-27",

      "call\_talktime": 102

    }];

    // Create axes

    var categoryAxis = chart.xAxes.push(new am4charts.CategoryAxis());

    categoryAxis.dataFields.category = "\_id";

    categoryAxis.renderer.grid.template.location = 0;

    categoryAxis.renderer.minGridDistance = 30;

    var valueAxis = chart.yAxes.push(new am4charts.ValueAxis());

    // Create series

    var series = chart.series.push(new am4charts.ColumnSeries());

    series.dataFields.valueY = "call\_talktime";

    series.dataFields.categoryX = "\_id";

    series.name = "call\_talktime";

    series.columns.template.tooltipText = "{categoryX}: [bold]{valueY}[/]";

    series.columns.template.fillOpacity = .8;

    series.stroke = am4core.color("#0f75bc");

    series.columns.template.fill = am4core.color("#0f75bc");

    series.columns.template.tooltipText = "{categoryX}: [bold]{valueY}[/]";

    series.columns.template.fillOpacity = .8;

    series.columns.template.width = 5;

    var columnTemplate = series.columns.template;

    columnTemplate.strokeWidth = 2;

    columnTemplate.strokeOpacity = 1;

  }

}

**ci-agent-compliance-monitoring**

**html**

<app-ci-nav-bar></app-ci-nav-bar>

<body>

    <div id="main-div">

        <section>

            <div class="container-fluid">

                <div class="box">

                    <div class="form-group pt-3 ">

                        <div class="row">

                            <div class="col-md-10">

                                <div class="row">

                                    <div class="col-md-4">

                                        <label class="date-block-ACM ACM-label">Date Range</label>

                                        <input type="text" placeholder="choose date range" name="daterangeInput"

                                            daterangepicker [options]="options"

                                            (selected)="selectedDate($event, daterange)" class="form-control" />

                                    </div>

                                    <div class="col-md-3">

                                        <label class="ACM-label">Select Country</label>

                                          <angular2-multiselect [data]="dropdownList" [(ngModel)]="selectedItems"

                                          [settings]="dropdownSettings"

                                          (onSelect)="onItemSelect($event)"

                                          (onDeSelect)="OnItemDeSelect($event)"

                                          (onSelectAll)="onSelectAll($event)"

                                          (onDeSelectAll)="onDeSelectAll($event)"></angular2-multiselect>

                                    </div>

                                    <div class="col-md-5">

                                        <label class="date-block-ACM ACM-label">Percentage %</label>

                                        <input class="percentage-min" type="number" value="Min" min="0" max="100"

                                            placeholder="min" [ngModel]="minValue"

                                            (ngModelChange)="minChangeEvent($event)">

                                        <input class="percentage-max" type="number" value="Max" min="0" max="100"

                                            placeholder="max" [ngModel]="maxValue"

                                            (ngModelChange)="maxChangeEvent($event)">

                                    </div>

                                </div>

                            </div>

                            <div class="col-md-2">

                               <button  (click)="formdata()">Submit</button>

                            </div>

                        </div>

                    </div>

                    <div class="row">

                        <div class="col-md-4 col-scroll">

                            <div class="card">

                                <div class="card-body">

                                    <p>Call Opening Compliance %</p>

                                    <div id="callOpeningCompliance" style="height:200px; width: 100%;"></div>

                                </div>

                            </div>

                        </div>

                        <div class="col-md-4 col-scroll">

                            <div class="card">

                                <div class="card-body">

                                    <p>Call Recording DisClosure %</p>

                                    <div id="callRecordingDisClosure" style="height: 200px; width: 100%;"></div>

                                </div>

                            </div>

                        </div>

                        <div class="col-md-4 col-scroll">

                            <div class="card">

                                <div class="card-body">

                                    <p>Other LOB Check %</p>

                                    <div id="otherLOBCheck" style="height: 200px; width: 100%;"></div>

                                </div>

                            </div>

                        </div>

                    </div>

                    <div class="row pt-4">

                        <div class="col-md-4 col-scroll">

                            <div class="card">

                                <div class="card-body">

                                    <p>Self Help Option %</p>

                                    <div id="selfHelpOption" style="height: 200px; width: 100%;"></div>

                                </div>

                            </div>

                        </div>

                        <div class="col-md-4 col-scroll">

                            <div class="card">

                                <div class="card-body">

                                    <p>Cancel Tech Visit %</p>

                                    <div id="cancelTechVisit" style="height: 200px; width: 100%;"></div>

                                </div>

                            </div>

                        </div>

                        <div class="col-md-4 col-scroll">

                            <div class="card">

                                <div class="card-body">

                                    <p>Call Closing Compliance %</p>

                                    <div id="callClosingCompliance" style="height: 200px; width: 100%;"></div>

                                </div>

                            </div>

                        </div>

                    </div>

                    <div class="row pt-4 pb-5">

                        <div class="col-md-9">

                            <div class="card">

                                <div class="card-body">

                                    <p>Empathy / Sympathy %</p>

                                    <div id="empathySympathy" style="height: 300px; width: 100%;"></div>

                                </div>

                            </div>

                        </div>

                        <div class="col-md-3">

                            <div class="card">

                                <div class="card-body">

                                    <p>Overall Behavioural Sentiments %</p>

                                    <div id="overallBehaviouralSentiments" style="height: 275px; width: 100%;">

                                    </div>

                                </div>

                            </div>

                        </div>

                    </div>

                </div>

            </div>

        </section>

    </div>

</body>

**css**

body {

    background: #f2f2f2;

    background-repeat: no-repeat;

    background-size: cover;

    background-attachment: fixed;

    height: 100%;

}

.cuppa-dropdown {

    position: absolute;

    background: white;

    border-radius: 4px;

    border: 1px solid gainsboro;

}

angular2-multiselect.myclass.custom-class.ng-untouched.ng-pristine.ng-valid {

    position: absolute;

    margin-top: 27px;

    background: white;

    border-radius: 4px;

    border: 1px solid gainsboro;

    margin-left: -86px;

    width: inherit;

    height: 37px;

}

.selected-list {

    position: absolute;

    margin-top: -3px;

    background: white;

    border-radius: 4px;

    border: 1px solid gainsboro;

    /\* margin-left: 0px; \*/

    width: 187px;

    height: 37px;

}

angular2-multiselect.myclass.custom-class.ng-valid.ng-touched.ng-dirty {

    position: absolute;

    margin-top: 27px;

    background: white;

    border-radius: 4px;

    border: 1px solid gainsboro;

    margin-left: -86px;

    width: inherit;

    height: 37px;

}

.cuppa-dropdown {

    position: absolute !important;

    margin-top: -3px !important;

    background: white !important;

    border-radius: 4px !important;

    border: 1px solid gainsboro !important;

    /\* margin-left: 0px; \*/

    width: 187px !important;

    height: 37px !important;

}

a#navbarDropdown {

    margin-top: 8px;

}

img.user-img {

    width: 40px;

}

.sticky-top {

    padding-right: 50px;

    padding-left: 50px;

    background-color: #ffffff !important;

}

.box {

    width: 95%;

    margin: auto;

}

.card {

    background-color: #ffffff;

}

.date-form {

    border: none;

    width: 100%;

    padding: 6px;

}

button.btn-primary {

    background-color: #0f75bc!important;

}

.col-scroll .card {

    height: 280px;

    position: relative;

    top: 0px;

    bottom: 0;

    left: 0;

    width: 100%;

    overflow-y: scroll;

}

label.date-block-ACM {

    display: block;

}

label.ACM-label {

    font-size: 13px;

    font-weight: 700;

}

input.percentage-min {

    border: 1px solid #ced4da;

    border-radius: .25rem;

    padding: 5px 5px;

    width: 20%;

}

input.percentage-max {

    border: 1px solid #ced4da;

    border-radius: .25rem;

    padding: 5px 5px;

    width: 20%;

}

/\*\*\*\*\*MOBILE MEDIA QUERY\*\*\*\*\*/

@media only screen and (max-width: 768px) {

    .form {

        width: 60%;

    }

    .right-nav {

        text-align: center;

    }

}

div#main-div {

    background: #f2f2f2;

}

select.form-control {

    margin-top: -3px;

}

**ts**

import { Component, OnInit } from '@angular/core';

import \* as CanvasJS from 'src/assets/canvasjs.min.js';

import { FormGroup, FormBuilder } from '@angular/forms';

import { CiAgentComServiceService } from '../services/ci-agent-com-service.service';

@Component({

  selector: 'app-ci-agent-compliance-monitoring',

  templateUrl: './ci-agent-compliance-monitoring.component.html',

  styleUrls: ['./ci-agent-compliance-monitoring.component.css']

})

export class CIAgentComplianceMonitoringComponent implements OnInit {

  constructor(private fb: FormBuilder, private ciAgentSer: CiAgentComServiceService) { }

  dropdownList = [];

  selectedItems = [];

  dropdownSettings = {};

  formval = [];

  ngOnInit() {

    this.selectCountryFunction();

    this.callOpening();

    this.LOBCheck();

    this.callRecording();

    this.cancelTech();

    this.selfHelp();

    this.callClosing();

    this.overRollBehave();

    this.empathySympathy();

  }

  selectCountryFunction() {

    this.dropdownList = [

      { "id": 1, "itemName": "Florida" },

      { "id": 2, "itemName": "XHS" },

      { "id": 3, "itemName": "GBR" },

      { "id": 4, "itemName": "WNE" },

      { "id": 5, "itemName": "Freedom" },

      { "id": 6, "itemName": "TwinCity" },

      { "id": 7, "itemName": "Beltway" },

      { "id": 8, "itemName": "Keystone" },

      { "id": 9, "itemName": "Portland" },

      { "id": 10, "itemName": "Houston" },

      { "id": 7, "itemName": "Big South" },

      { "id": 8, "itemName": "Mountain West" },

      { "id": 9, "itemName": "California" },

      { "id": 10, "itemName": "Seattle" }

    ];

    this.dropdownSettings = {

      singleSelection: false,

      text: "Select Countries",

      selectAllText: 'Select All',

      unSelectAllText: 'UnSelect All',

      enableSearchFilter: true,

      classes: "myclass custom-class"

    };

  }

  onItemSelect(item: any) {

    console.log(item);

    console.log(this.selectedItems);

  }

  OnItemDeSelect(item: any) {

    console.log(item);

    console.log(this.selectedItems);

  }

  onSelectAll(items: any) {

    console.log(items);

  }

  onDeSelectAll(items: any) {

    console.log(items);

  }

  public daterange: any = {};

  public options: any = {

    locale: { format: 'DD-MM-YYYY' },

    alwaysShowCalendars: false,

  };

  public selectedDate(value: any, datepicker?: any) {

    datepicker.start = value.start;

    datepicker.end = value.end;

    this.daterange.start = value.start;

    this.daterange.end = value.end;

    this.daterange.label = value.label;

    console.log("start date range :-  ", value.start.format('DD-MM-YYYY'))

    console.log("end date range :-  ", value.end.format('DD-MM-YYYY'))

  }

// ---------------------------percentage filter-----------------------------//

  private minValue;

  private maxValue;

  private minChangeEvent(event) {

    if (this.minValue > 100) {

      this.minValue = 100;

      console.log('Min Value New :-  ', this.minValue);

      this.minValue = event;

      this.ciAgentSer.minPercentFun(this.minValue).subscribe(data => {

        console.log("Min Value New :- ", this.minValue);

      })

    }

    else if (this.minValue < 0) {

      this.minValue = 0;

      console.log('Min Value New :-  ', this.minValue);

      this.minValue = event;

      this.ciAgentSer.minPercentFun(this.minValue).subscribe(data => {

        console.log("Min Value New :- ", this.minValue);

      })

    }

    else {

      console.log('Min Value New :-  ', this.minValue);

      this.minValue = event;

      this.ciAgentSer.minPercentFun(this.minValue).subscribe(data => {

        console.log("Min Value New :- ", this.minValue);

      })

    }

  }

  private maxChangeEvent(event) {

    if (this.maxValue > 100) {

      this.maxValue = 100;

      console.log('Max Value New :-  ', this.maxValue);

      this.maxValue = event;

      this.ciAgentSer.minPercentFun(this.maxValue).subscribe(data => {

        console.log("Max Value New :- ", this.maxValue);

      })

    }

    else if (this.maxValue < 0) {

      this.maxValue = 0;

      console.log('Max Value New :-  ', this.maxValue);

      this.maxValue = event;

      this.ciAgentSer.minPercentFun(this.maxValue).subscribe(data => {

        console.log("Max Value New :- ", this.maxValue);

      })

    }

    else {

      console.log('Max Value New :-  ', this.maxValue);

      this.maxValue = event;

      this.ciAgentSer.minPercentFun(this.maxValue).subscribe(data => {

        console.log("Max Value New :- ", this.maxValue);

      })

    }

  }

  formdata(){

    this.formval.push(this.minValue);

    console.log("araaayyy        ", this.formval);

  }

  // ---------------------------graphs functions-----------------------------//

  callOpening() {

    let chart = new CanvasJS.Chart("callOpeningCompliance", {

      animationEnabled: true,

      axisX: {

        interval: 1

      },

      axisY: {

        interval: 100

      },

      axisY2: {

        interlacedColor: "rgba(1,77,101,.2)",

        gridColor: "rgba(1,77,101,.1)",

        title: "Number of Companies"

      },

      dataPointWidth: 7,

      data: [{

        type: "bar",

        name: "companies",

        axisYType: "primary",

        color: "#0165",

        dataPoints: [

          { y: 25, label: "Country", color: "#ff8c91" },

          { y: 75, label: "Country", color: "#649dd2" },

          { y: 81, label: "Country", color: "#649dd2" },

          { y: 99, label: "Country", color: "#1075bd" },

          { y: 99, label: "Country", color: "#1075bd" },

        ]

      }]

    });

    chart.render();

  }

  callRecording() {

    let chart = new CanvasJS.Chart("callRecordingDisClosure", {

      animationEnabled: true,

      axisX: {

        interval: 1

      },

      axisY: {

        interval: 100

      },

      axisY2: {

        interlacedColor: "rgba(1,77,101,.2)",

        gridColor: "rgba(1,77,101,.1)",

        title: "Number of Companies"

      },

      dataPointWidth: 7,

      data: [{

        type: "bar",

        name: "companies",

        axisYType: "primary",

        color: "#0165",

        dataPoints: [

          { y: 25, label: "Country", color: "#ff8c91" },

          { y: 75, label: "Country", color: "#649dd2" },

          { y: 81, label: "Country", color: "#649dd2" },

          { y: 99, label: "Country", color: "#1075bd" },

          { y: 99, label: "Country", color: "#1075bd" },

        ]

      }]

    });

    chart.render();

  }

  LOBCheck() {

    let chart = new CanvasJS.Chart("otherLOBCheck", {

      animationEnabled: true,

      axisX: {

        interval: 1

      },

      axisY: {

        interval: 100

      },

      axisY2: {

        interlacedColor: "rgba(1,77,101,.2)",

        gridColor: "rgba(1,77,101,.1)",

        title: "Number of Companies"

      },

      dataPointWidth: 7,

      data: [{

        type: "bar",

        name: "companies",

        axisYType: "primary",

        color: "#0165",

        dataPoints: [

          { y: 25, label: "Country", color: "#ff8c91" },

          { y: 75, label: "Country", color: "#649dd2" },

          { y: 81, label: "Country", color: "#649dd2" },

          { y: 99, label: "Country", color: "#1075bd" },

          { y: 99, label: "Country", color: "#1075bd" },

        ]

      }]

    });

    chart.render();

  }

  selfHelp() {

    let chart = new CanvasJS.Chart("selfHelpOption", {

      animationEnabled: true,

      axisX: {

        interval: 1

      },

      axisY: {

        interval: 100

      },

      axisY2: {

        interlacedColor: "rgba(1,77,101,.2)",

        gridColor: "rgba(1,77,101,.1)",

        title: "Number of Companies"

      },

      dataPointWidth: 7,

      data: [{

        type: "bar",

        name: "companies",

        axisYType: "primary",

        color: "#0165",

        dataPoints: [

          { y: 25, label: "Country", color: "#ff8c91" },

          { y: 75, label: "Country", color: "#649dd2" },

          { y: 81, label: "Country", color: "#649dd2" },

          { y: 99, label: "Country", color: "#1075bd" },

          { y: 99, label: "Country", color: "#1075bd" },

        ]

      }]

    });

    chart.render();

  }

  cancelTech() {

    let chart = new CanvasJS.Chart("cancelTechVisit", {

      animationEnabled: true,

      axisX: {

        interval: 1

      },

      axisY: {

        interval: 100

      },

      axisY2: {

        interlacedColor: "rgba(1,77,101,.2)",

        gridColor: "rgba(1,77,101,.1)",

        title: "Number of Companies"

      },

      dataPointWidth: 7,

      data: [{

        type: "bar",

        name: "companies",

        axisYType: "primary",

        color: "#0165",

        dataPoints: [

          { y: 25, label: "Country", color: "#ff8c91" },

          { y: 75, label: "Country", color: "#649dd2" },

          { y: 81, label: "Country", color: "#649dd2" },

          { y: 99, label: "Country", color: "#1075bd" },

          { y: 99, label: "Country", color: "#1075bd" },

        ]

      }]

    });

    chart.render();

  }

  callClosing() {

    let chart = new CanvasJS.Chart("callClosingCompliance", {

      animationEnabled: true,

      axisX: {

        interval: 1

      },

      axisY: {

        interval: 100

      },

      axisY2: {

        interlacedColor: "rgba(1,77,101,.2)",

        gridColor: "rgba(1,77,101,.1)",

        title: "Number of Companies"

      },

      dataPointWidth: 7,

      data: [{

        type: "bar",

        name: "companies",

        axisYType: "primary",

        color: "#0165",

        dataPoints: [

          { y: 25, label: "Country", color: "#ff8c91" },

          { y: 75, label: "Country", color: "#649dd2" },

          { y: 81, label: "Country", color: "#649dd2" },

          { y: 99, label: "Country", color: "#1075bd" },

          { y: 99, label: "Country", color: "#1075bd" },

        ]

      }]

    });

    chart.render();

  }

  empathySympathy() {

    var chart = new CanvasJS.Chart("empathySympathy", {

      animationEnabled: true,

      theme: "light2",

      dataPointWidth: 7,

      axisX: {

        valueFormatString: "MMM"

      },

      axisY: {

        prefix: "$",

        labelFormatter: addSymbols

      },

      toolTip: {

        shared: true

      },

      legend: {

        cursor: "pointer",

        itemclick: toggleDataSeries

      },

      data: [

        {

          type: "column",

          name: "Actual Sales",

          showInLegend: true,

          xValueFormatString: "MMMM YYYY",

          yValueFormatString: "$#,##0",

          dataPoints: [

            { x: new Date(2016, 0), y: 20000, color: "#0e74bd" },

            { x: new Date(2016, 1), y: 30000, color: "#0e74bd" },

            { x: new Date(2016, 2), y: 25000, color: "#0e74bd" },

            { x: new Date(2016, 3), y: 70000, color: "#0e74bd" },

            { x: new Date(2016, 4), y: 50000, color: "#0e74bd" },

            { x: new Date(2016, 5), y: 35000, color: "#0e74bd" },

            { x: new Date(2016, 6), y: 30000, color: "#0e74bd" },

            { x: new Date(2016, 7), y: 43000, color: "#0e74bd" },

            { x: new Date(2016, 8), y: 35000, color: "#0e74bd" },

            { x: new Date(2016, 9), y: 30000, color: "#0e74bd" },

            { x: new Date(2016, 10), y: 40000, color: "#0e74bd" },

            { x: new Date(2016, 11), y: 50000, color: "#0e74bd" }

          ]

        },

      ]

    });

    chart.render();

    function addSymbols(e) {

      var suffixes = ["", "K", "M", "B"];

      var order = Math.max(Math.floor(Math.log(e.value) / Math.log(1000)), 0);

      if (order > suffixes.length - 1)

        order = suffixes.length - 1;

      var suffix = suffixes[order];

      return CanvasJS.formatNumber(e.value / Math.pow(1000, order)) + suffix;

    }

    function toggleDataSeries(e) {

      if (typeof (e.dataSeries.visible) === "undefined" || e.dataSeries.visible) {

        e.dataSeries.visible = false;

      } else {

        e.dataSeries.visible = true;

      }

      e.chart.render();

    }

  }

  overRollBehave() {

    var chart = new CanvasJS.Chart("overallBehaviouralSentiments", {

      animationEnabled: true,

      title: {

        horizontalAlign: "left"

      },

      data: [{

        type: "doughnut",

        dataPoints: [

          { y: 67, label: "Neutral", color: "#54d8fe" },

          { y: 28, label: "Positive", color: "#a3a0fb" },

          { y: 10, label: "Negative", color: "#ffda82" },

        ]

      }]

    });

    chart.render();

  }

}

**ci-nav-bar**

**html**

<nav class="navbar navbar-expand-lg navbar-light shadow fixed-top sticky-top">

  <button class="navbar-toggler navbar-toggler-right" type="button" data-toggle="collapse"

    data-target="#navbarTogglerDemo02" aria-controls="navbarTogglerDemo02" aria-expanded="false"

    aria-label="Toggle navigation">

    <span class="navbar-toggler-icon"></span>

  </button>

  <a class="navbar-brand"><img src="assets/Voice.ai-logo.png" alt="logo"></a>

  <div class="collapse navbar-collapse custom-navbar" id="navbarTogglerDemo02">

    <ul class="navbar-nav ml-md-auto">

      <li class="nav-item">

        <a class="nav-link" routerLink="/home" routerLinkActive="active">Home</a>

      </li>

      <li class="nav-item">

        <a class="nav-link" routerLink="/cioverview" routerLinkActive="active">Overview</a>

      </li>

      <li class="nav-item">

        <a class="nav-link" routerLink="/ciagent" routerLinkActive="active">Agent Compliance Monitoring</a>

      </li>

    </ul>

    <ul class="navbar-nav ml-auto right-nav">

      <li class="nav-item dropdown">

        <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-toggle="dropdown"

          aria-haspopup="true" aria-expanded="false">

          Johan deo

        </a>

        <div class="dropdown-menu" aria-labelledby="navbarDropdown">

          <a class="dropdown-item" routerLink="/login">logout</a>

        </div>

      </li>

      <li class="nav-item">

        <a class="nav-link" href="#"><img src="assets/user.png" alt="arrow" class="user-img"></a>

      </li>

    </ul>

  </div>

</nav>

**css**

.sticky-top{

    padding-right:50px;

    padding-left:50px;

    background-color: #ffffff !important;

}

img.user-img {

    width:40px;

}

a#navbarDropdown {

    margin-top: 10px;

    }

.navbar-nav {

    text-align: center;

}

@media only screen and (max-width: 768px) {

 .form {

     width: 60%;

 }

 .right-nav{

 text-align:center;

 }

 }

     li.nav-item.mt-2 {

    font-family: 'Open Sans', sans-serif;

    padding: 0px 25px;

    font-weight: 600;

}

.nav-custom-link.active {

    color: #1d5574 !important;

}

a.nav-custom-link {

    color: #b0bac9 !important;

}

**crm-overview component**

**html**

<app-crm-nav-bar></app-crm-nav-bar>

<section>

    <div class="container-fluid">

        <div class="box">

            <form class="pt-3">

              <div class="form-group pt-3 ">

                 <div class="row">

                        <div class="col-md-4">

                        <div class="form-group">

                          <label class="date-block-ACM ACM-label">Date Range</label>

                      <input class="form-control date-form" type="text" name="daterange" value="01/01/2020 - 01/15/2020" />

            <img class="icon-calendar" src="assets/Calendar Icon.png">

                            </div>

                        </div>

 <div class="col-md-3">

 <div class="form-group">

       <label class="ACM-label">Select Process</label>

      <select class="form-control" id="sel1" name="sellist1">

      <option>Process 01</option>

      <option>Process 02</option>

      <option>Process 03</option>

      <option>Process 04</option>

      </select>

     <img class="process-arrow" style="width: 10px;"

                            src="assets/Dropdown-arrow.png">

      </div>

 </div>

 <div class="col-md-3">

 <div class="form-group">

    <label class="ACM-label">Region</label>

      <select class="form-control" id="sel1" name="sellist1">

      <option>Chandigarh</option>

      <option>Pune</option>

      <option>Mumbai</option>

      </select>

      <img class="region-arrow" style="width: 10px;"

                            src="assets/Dropdown-arrow.png">

 </div>

 </div>

                <div class="col-md-2 submit-col">

                <button type="submit" class="btn btn-block btn-primary btn-search-row mb-2">Submit</button>

                </div>

</div>

</div>

</form>

            <div class="row custom-4-col-row">

                <div class="col-md-3">

                    <div class="card">

                        <div class="card-body">

                           <div class="row border-row">

                           <div class="col-md-8">

                           <p>CALL VOLUME</p>

                           <h3>123</h3>

                           </div>

                           <div class="col-md-4 text-right">

                           <img src="assets/Call volume.png" alt="call\_icon">

                           </div>

                           </div>

                            <div class="row card-row-2">

                                <div class="col-md-7 col-6">

                                    <p>Outbound</p>

                                </div>

                                <div class="col-md-5 col-6">

                                    <p>Inbound</p>

                                </div>

                            </div>

                            <div class="row">

                                <div class="col-md-7 col-6">

                                    <h3>23</h3> </div>

                                <div class="col-md-5 col-6">

                                    <h3>100</h3> </div>

                            </div>

                        </div>

                    </div>

                </div>

                <div class="col-md-3">

                    <div class="card">

                        <div class="card-body">

                          <div class="row border-row">

                           <div class="col-md-8">

                           <p>NUMBER OF CALLERS</p>

                           <h3>150</h3>

                           </div>

                           <div class="col-md-4 text-right">

                           <img src="assets/Number of Callers.png" alt="call\_icon">

                           </div>

                           </div>

                            <div class="row card-row-2">

                                <div class="col-md-7 col-6">

                                    <p>No. of Agents</p>

                                </div>

                                <div class="col-md-5 col-6">

                                    <p>Calls/Agent</p>

                                </div>

                            </div>

                            <div class="row">

                                <div class="col-md-7 col-6">

                                    <h3>25</h3> </div>

                                <div class="col-md-5 col-6">

                                    <h3>125</h3> </div>

                            </div>

                        </div>

                    </div>

                </div>

                <div class="col-md-3">

                    <div class="card">

                        <div class="card-body">

                           <div class="row">

                           <div class="col-md-8">

                           <p>FIRST CALL RESOLUTION</p>

                           <h3>55</h3>

                           </div>

                           <div class="col-md-4 text-right">

                           <img src="assets/First Call Resolution.png" alt="call\_icon">

                           </div>

                           </div>

                        </div>

                    </div>

                </div>

                <div class="col-md-3">

                    <div class="card">

                        <div class="card-body">

                           <div class="row border-row">

                           <div class="col-md-8">

                           <p>ABANDONED CALLs</p>

                           <h3>120</h3>

                           </div>

                           <div class="col-md-4 text-right">

                           <img src="assets/Abandoned Calls.png" alt="call\_icon">

                           </div>

                           </div>

                            <div class="row card-row-2">

                                <div class="col-md-6 col-6">

                                    <p>Avg Wait Time until Abandon</p>

                                </div>

                                <div class="col-md-6 col-6">

                                    <p>Avg IVR Wait Time</p>

                                </div>

                            </div>

                            <div class="row">

                                <div class="col-md-6 col-6">

                                    <h3>10m 20s</h3> </div>

                                <div class="col-md-6 col-6">

                                    <h3>15m 05s</h3> </div>

                            </div>

                        </div>

                    </div>

                </div>

            </div>

            <div class="row pt-4 pb-4 custom-2-col-row">

                <div class="col-md-6">

                    <div class="card">

                        <div class="card-body">

                            <p>TOTAL HANDLING TIME</p>

                            <div id="chartContainer" style="height: 300px; width: 100%;"></div>

                        </div>

                    </div>

                </div>

                <div class="col-md-6">

                    <div class="card">

                        <div class="card-body">

                            <p>PEAK HOUR TRAFFIC</p>

                            <div id="chartdiv" style="height: 300px;"></div>

                        </div>

                    </div>

                </div>

            </div>

        </div>

    </div>

</section>

**css**

body {

    background: #f2f2f2;

    background-repeat: no-repeat;

    background-size: cover;

    background-attachment: fixed;

    height: 100%;

    font-size: 13px;

    font-family: 'Open Sans', sans-serif;

}

a#navbarDropdown {

margin-top: 8px;

}

img.user-img {

width:40px;

}

.sticky-top{

padding-right:50px;

padding-left:50px;

background-color: #ffffff !important;

}

.box{

width: 95%;

margin: auto;

}

.card{

background-color: #ffffff;

}

.form{

border:none;

width:15%;

padding:4px;

}

.border-row{

border-bottom: 1px solid #e0e7ff;

padding-bottom: 10px;

margin-bottom: 15px;

}

/\*

hr {

  margin-top: 1rem;

  margin-bottom: 1rem;

  border: 0;

  border-top: 1px solid #e0e7ff;

}

.custom-4-col-row {

    text-align: center;

}\*/

.custom-4-col-row .card-body {

    height: 205px;

}

.form-row {

    margin: 0px 2px;

}

select {

  -webkit-appearance: none;

  -moz-appearance: none;

  appearance: none;

}

.dropdown-toggle::after {

    display: none;

}

.form-control {

    border: none !important;

    font-size: 14px;

}

.processdiv-col {

    border-left: 2px solid #8e8e8e;

    border-right: 2px solid #8e8e8e;

}

img.icon-calendar {

    position: absolute;

    top: 40px;

    right: 25px;

}

img.process-arrow {

    position: absolute;

    top: 46px;

    right: 26px

}

img.region-arrow {

    position: absolute;

    top: 46px;

    right: 26px

}

img.user-icon {

    position: relative;

    right: -4px;

}

button.btn-primary {

    background-color: #0f75bc!important;

    margin-top: 29px;

    font-size: 14px;

}

form {

    width: 75%;

}

label.ACM-label {

    font-size: 14px;

    font-weight: 600;

}

li.nav-item.mt-2 {

    font-family: 'Open Sans', sans-serif;

    padding: 0px 25px;

    font-weight: 600;

}

li.nav-item.mt-2 {

    font-family: 'Open Sans', sans-serif;

    padding: 0px 25px;

    font-weight: 600;

}

.nav-custom-link.active {

    color: #1d5574 !important;

}

a.nav-custom-link {

    color: #b0bac9 !important;

}

a.dropdown-item {

    font-size: 13px;

}

.card {

    border: none;

}

.border-row p {

    margin-bottom: 5px;

    font-weight: 600;

}

.custom-2-col-row p {

    font-weight: 600;

}

.card-row-2 p {

    margin-bottom: 5px;

    font-weight: 600;

}

.card-body h3 {

    font-weight: 600;

}

.custom-4-col-row .col-md-3 {

    padding-left: 10px;

    padding-right: 10px;

}

.custom-2-col-row .col-md-6 {

    padding-left: 10px;

    padding-right: 10px;

}

.form-group .col-md-4, .form-group .col-md-3, .form-group .col-md-2 {

    padding-left: 10px;

    padding-right: 10px;

}

.form-group {

    margin-bottom: 10px;

}

/\*\*\*\*\*MOBILE MEDIA QUERY\*\*\*\*\*/

   @media only screen and (max-width: 768px) {

.right-nav{

text-align:center;

}

}

**ts**

import { Component, OnInit, NgZone  } from '@angular/core';

import \* as CanvasJS from 'src/assets/canvasjs.min.js';

import \* as am4core from "@amcharts/amcharts4/core";

import \* as am4charts from "@amcharts/amcharts4/charts";

import am4themes\_animated from "@amcharts/amcharts4/themes/animated";

am4core.useTheme(am4themes\_animated);

@Component({

  selector: 'app-crm-overview',

  templateUrl: './crm-overview.component.html',

  styleUrls: ['./crm-overview.component.css']

})

export class CRMOverviewComponent implements OnInit {

  private chart: am4charts.XYChart;

  constructor(private zone: NgZone) {}

  ngOnInit() {

    this.abondandcall();

}

abondandcall(){

  let chart = new CanvasJS.Chart("chartContainer", {

    animationEnabled: true,

    data: [{

      type: "doughnut",

      startAngle: 60,

      //innerRadius: 60,

      indexLabelFontSize: 17,

      indexLabel: "{label} - #percent%",

      toolTipContent: "<b>{label}:</b> {y} (#percent%)",

      dataPoints: [

        { y: 67, label: "Inbox" },

        { y: 28, label: "Archives" },

        { y: 10, label: "Labels" },

        { y: 7, label: "Drafts"},

        { y: 15, label: "Trash"},

        { y: 6, label: "Spam"}

      ]

    }]

  });

  chart.render();

}

ngAfterViewInit() {

  let chart = am4core.create("chartdiv", am4charts.XYChart);

  chart.maskBullets = false;

  let xAxis = chart.xAxes.push(new am4charts.CategoryAxis());

  let yAxis = chart.yAxes.push(new am4charts.CategoryAxis());

  xAxis.dataFields.category = "weekday";

  yAxis.dataFields.category = "hour";

  xAxis.renderer.grid.template.disabled = true;

  xAxis.renderer.minGridDistance = 40;

  yAxis.renderer.grid.template.disabled = true;

  yAxis.renderer.inversed = true;

  yAxis.renderer.minGridDistance = 30;

  let series = chart.series.push(new am4charts.ColumnSeries());

  series.dataFields.categoryX = "weekday";

  series.dataFields.categoryY = "hour";

  series.dataFields.value = "value";

  series.sequencedInterpolation = true;

  series.defaultState.transitionDuration = 3000;

  let bgColor = new am4core.InterfaceColorSet().getFor("background");

  let columnTemplate = series.columns.template;

  columnTemplate.strokeWidth = 1;

  columnTemplate.strokeOpacity = 0.2;

  columnTemplate.stroke = bgColor;

  columnTemplate.tooltipText = "{weekday}, {hour}: {value.workingValue.formatNumber('#.')}";

  columnTemplate.width = am4core.percent(100);

  columnTemplate.height = am4core.percent(100);

  series.heatRules.push({

    target: columnTemplate,

    property: "fill",

    min: am4core.color(bgColor),

    max: chart.colors.getIndex(0)

  });

  // heat legend

  let heatLegend = chart.bottomAxesContainer.createChild(am4charts.HeatLegend);

  heatLegend.width = am4core.percent(100);

  heatLegend.series = series;

  heatLegend.valueAxis.renderer.labels.template.fontSize = 9;

  heatLegend.valueAxis.renderer.minGridDistance = 30;

  // heat legend behavior

  series.columns.template.events.on("over", function(event) {

    handleHover(event.target);

  })

  series.columns.template.events.on("hit", function(event) {

    handleHover(event.target);

  })

  function handleHover(column) {

    if (!isNaN(column.dataItem.value)) {

      heatLegend.valueAxis.showTooltipAt(column.dataItem.value)

    }

    else {

      heatLegend.valueAxis.hideTooltip();

    }

  }

  series.columns.template.events.on("out", function(event) {

    heatLegend.valueAxis.hideTooltip();

  })

  chart.data = [

    {

      "hour": "12pm",

      "weekday": "Sun",

      "value": 2990

    },

    {

      "hour": "1am",

      "weekday": "Sun",

      "value": 2520

    },

    {

      "hour": "2am",

      "weekday": "Sun",

      "value": 2334

    },

    {

      "hour": "3am",

      "weekday": "Sun",

      "value": 2230

    },

    {

      "hour": "10am",

      "weekday": "Sat",

      "value": 3287

    },

  ];

}

ngOnDestroy() {

  this.zone.runOutsideAngular(() => {

    if (this.chart) {

      this.chart.dispose();

    }

  });

}

  }

**crm-call-monitoring component**

**html**

<app-crm-nav-bar></app-crm-nav-bar>

<div class="container-fluid">

    <!-- form -->

    <div class="row">

        <div class="col-md-10">

            <br>

            <!--speedometer graphs  -->

            <div class="row">

                <div class="col-md-2">

                    <div id="callSetupSuccess"></div>

                </div>

                <div class="col-md-2">

                    <div id="interactionRouted"></div>

                </div>

                <div class="col-md-2">

                    <div id="repeatCall"></div>

                </div>

                <div class="col-md-2">

                    <div id="avoidableCall"></div>

                </div>

                <div class="col-md-2">

                    <div id="callResolved"></div>

                </div>

            </div>

            <br>

            <!-- call handling -->

            <div class="row">

                <div class="col-md-6">

                    <p>Call Handling Time</p>

                    <div id="callHandle1" style="height: 200px; width: 100%;"></div>

                </div>

                <div class="col-md-6">

                    <p>Call Handling Time</p>

                    <div id="callHandle2" style="height: 200px; width: 100%;"></div>

                </div>

            </div>

            <br>

            <!-- first 3 col row -->

            <div class="row">

                <div class="col-md-4">

                    <p>Abandoned Calls</p>

                    <div id="abandonedCalls" style="height: 200px; width: 100%;"></div>

                </div>

                <div class="col-md-4">

                    <p>Answered Calls-IVR WAIT TIME</p>

                    <div id="answeredCallIVR" style="height: 200px; width: 100%;"></div>

                </div>

                <div class="col-md-4">

                    <p>Call Duration</p>

                    <div id="callDuration" style="height: 200px; width: 100%;"></div>

                </div>

            </div>

            <br>

            <!-- second 3 col row -->

            <div class="row">

                <div class="col-md-4">

                    <p>Call Per Agent</p>

                    <div id="callPerAgent" style="height: 200px; width: 100%;"></div>

                </div>

                <div class="col-md-4">

                    <p>Average Speed Of Answer</p>

                    <div id="averageSpeedOfAnswer" style="height: 200px; width: 100%;"></div>

                </div>

                <div class="col-md-4">

                    <p>Cost Per Call</p>

                    <div id="costPerCall" style="height: 200px; width: 100%;"></div>

                </div>

            </div>

        </div>

        <div class="col-md-2">

            <br>

            <div class="row">

                <div class="col-md-4">

                    <p>PD 1</p>

                </div>

                <div class="col-md-2">

                    <div class="purple" style="background-color: rgb(183, 0, 255); border: rgb(183, 0, 255);"></div>

                </div>

                <div class="col-md-6">

                    <p>Work Order Details</p>

                </div>

            </div>

            <br>

            <div class="row">

                <div class="col-md-4">

                    <p>PD 3</p>

                </div>

                <div class="col-md-2">

                    <div class="blue" style="background-color: rgb(64, 11, 255); border: rgb(64, 11, 255);"></div>

                </div>

                <div class="col-md-6">

                    <p>Work Order Change</p>

                </div>

            </div>

            <br>

            <div class="row">

                <div class="col-md-4">

                    <p>PD 5</p>

                </div>

                <div class="col-md-2">

                    <div class="green" style="background-color: rgba(7, 92, 49, 0.966); border: rgba(7, 92, 49, 0.966);"></div>

                </div>

                <div class="col-md-6">

                    <p>Video Provisioning</p>

                </div>

            </div>

            <br>

            <div class="row">

                <div class="col-md-4">

                    <p>PD 7</p>

                </div>

                <div class="col-md-2">

                    <div class="navy" style="background-color: rgb(0, 102, 255); border: rgb(0, 102, 255);"></div>

                </div>

                <div class="col-md-6">

                    <p>Service Code Request</p>

                </div>

            </div>

            <br>

            <div class="row">

                <div class="col-md-4">

                    <p>PD 9</p>

                </div>

                <div class="col-md-2">

                    <div class="yellow" style="background-color: rgb(235, 255, 57); border: rgb(235, 255, 57);"></div>

                </div>

                <div class="col-md-6">

                    <p>Cancel Work Order</p>

                </div>

            </div>

            <br>

            <div class="row">

                <div class="col-md-4">

                    <p>PD 11</p>

                </div>

                <div class="col-md-2">

                    <div class="lightgreen" style="background-color: rgb(0, 255, 149); border: rgb(0, 255, 149);"></div>

                </div>

                <div class="col-md-6">

                    <p>Blank Call</p>

                </div>

            </div>

            <br>

            <div class="row">

                <div class="col-md-4">

                    <p>PD 13</p>

                </div>

                <div class="col-md-2">

                    <div class="pink" style="background-color: magenta; border: magenta;"></div>

                </div>

                <div class="col-md-6">

                    <p>Rescheduled Work Order</p>

                </div>

            </div>

        </div>

    </div>

</div>

**css**

.purple, .green, .yellow, .pink, .navy, .lightgreen, .blue{

    width: 8px;

    height: 8px;

    margin-top: 6px;

    border-radius: 20px;

}

**ts**

import { Component, OnInit, NgZone } from '@angular/core';

import \* as CanvasJS from 'src/assets/canvasjs.min.js';

import \* as am4core from "@amcharts/amcharts4/core";

import \* as am4charts from "@amcharts/amcharts4/charts";

import am4themes\_animated from "@amcharts/amcharts4/themes/animated";

am4core.useTheme(am4themes\_animated);

@Component({

  selector: 'app-crm-call-monitoring',

  templateUrl: './crm-call-monitoring.component.html',

  styleUrls: ['./crm-call-monitoring.component.css']

})

export class CRMCallMonitoringComponent implements OnInit {

  private chart: am4charts.XYChart;

  constructor(private zone: NgZone) { }

  ngAfterViewInit() {

    this.callSetupSuccess();

    this.interactionRouted();

    this.repeatCalls();

    this.avoidableCall();

    this.callResolved();

  }

  callSetupSuccess() {

    let chart = am4core.create("callSetupSuccess", am4charts.GaugeChart);

    chart.hiddenState.properties.opacity = 0; // this makes initial fade in effect

    chart.innerRadius = -15;

    let axis = chart.xAxes.push(new am4charts.ValueAxis<am4charts.AxisRendererCircular>());

    axis.min = 0;

    axis.max = 100;

    axis.strictMinMax = true;

    axis.renderer.grid.template.stroke = new am4core.InterfaceColorSet().getFor("background");

    axis.renderer.grid.template.strokeOpacity = 1;

    let colorSet = new am4core.ColorSet();

    let range0 = axis.axisRanges.create();

    range0.value = 0;

    range0.endValue = 100;

    range0.axisFill.fillOpacity = 1;

    range0.axisFill.fill = colorSet.getIndex(0);

    range0.axisFill.zIndex = - 1;

    let hand = chart.hands.push(new am4charts.ClockHand());

    // using chart.setTimeout method as the timeout will be disposed together with a chart

    chart.setTimeout(randomValue, 1000);

    function randomValue() {

      hand.showValue(20, 1000, am4core.ease.cubicOut);

    }

  }

  interactionRouted() {

    let chart = am4core.create("interactionRouted", am4charts.GaugeChart);

    chart.hiddenState.properties.opacity = 0; // this makes initial fade in effect

    chart.innerRadius = -15;

    let axis = chart.xAxes.push(new am4charts.ValueAxis<am4charts.AxisRendererCircular>());

    axis.min = 0;

    axis.max = 100;

    axis.strictMinMax = true;

    axis.renderer.grid.template.stroke = new am4core.InterfaceColorSet().getFor("background");

    axis.renderer.grid.template.strokeOpacity = 1;

    let colorSet = new am4core.ColorSet();

    let range0 = axis.axisRanges.create();

    range0.value = 0;

    range0.endValue = 100;

    range0.axisFill.fillOpacity = 1;

    range0.axisFill.fill = colorSet.getIndex(0);

    range0.axisFill.zIndex = - 1;

    let hand = chart.hands.push(new am4charts.ClockHand());

    // using chart.setTimeout method as the timeout will be disposed together with a chart

    chart.setTimeout(randomValue, 1000);

    function randomValue() {

      hand.showValue(20, 1000, am4core.ease.cubicOut);

    }

  }

  repeatCalls() {

    let chart = am4core.create("repeatCall", am4charts.GaugeChart);

    chart.hiddenState.properties.opacity = 0; // this makes initial fade in effect

    chart.innerRadius = -15;

    let axis = chart.xAxes.push(new am4charts.ValueAxis<am4charts.AxisRendererCircular>());

    axis.min = 0;

    axis.max = 100;

    axis.strictMinMax = true;

    axis.renderer.grid.template.stroke = new am4core.InterfaceColorSet().getFor("background");

    axis.renderer.grid.template.strokeOpacity = 1;

    let colorSet = new am4core.ColorSet();

    let range0 = axis.axisRanges.create();

    range0.value = 0;

    range0.endValue = 100;

    range0.axisFill.fillOpacity = 1;

    range0.axisFill.fill = colorSet.getIndex(0);

    range0.axisFill.zIndex = - 1;

    let hand = chart.hands.push(new am4charts.ClockHand());

    // using chart.setTimeout method as the timeout will be disposed together with a chart

    chart.setTimeout(randomValue, 1000);

    function randomValue() {

      hand.showValue(20, 1000, am4core.ease.cubicOut);

    }

  }

  avoidableCall(){

    let chart = am4core.create("avoidableCall", am4charts.GaugeChart);

    chart.hiddenState.properties.opacity = 0; // this makes initial fade in effect

    chart.innerRadius = -15;

    let axis = chart.xAxes.push(new am4charts.ValueAxis<am4charts.AxisRendererCircular>());

    axis.min = 0;

    axis.max = 100;

    axis.strictMinMax = true;

    axis.renderer.grid.template.stroke = new am4core.InterfaceColorSet().getFor("background");

    axis.renderer.grid.template.strokeOpacity = 1;

    let colorSet = new am4core.ColorSet();

    let range0 = axis.axisRanges.create();

    range0.value = 0;

    range0.endValue = 100;

    range0.axisFill.fillOpacity = 1;

    range0.axisFill.fill = colorSet.getIndex(0);

    range0.axisFill.zIndex = - 1;

    let hand = chart.hands.push(new am4charts.ClockHand());

    // using chart.setTimeout method as the timeout will be disposed together with a chart

    chart.setTimeout(randomValue, 1000);

    function randomValue() {

      hand.showValue(20, 1000, am4core.ease.cubicOut);

    }

  }

  callResolved(){

    let chart = am4core.create("callResolved", am4charts.GaugeChart);

    chart.hiddenState.properties.opacity = 0; // this makes initial fade in effect

    chart.innerRadius = -15;

    let axis = chart.xAxes.push(new am4charts.ValueAxis<am4charts.AxisRendererCircular>());

    axis.min = 0;

    axis.max = 100;

    axis.strictMinMax = true;

    axis.renderer.grid.template.stroke = new am4core.InterfaceColorSet().getFor("background");

    axis.renderer.grid.template.strokeOpacity = 1;

    let colorSet = new am4core.ColorSet();

    let range0 = axis.axisRanges.create();

    range0.value = 0;

    range0.endValue = 100;

    range0.axisFill.fillOpacity = 1;

    range0.axisFill.fill = colorSet.getIndex(0);

    range0.axisFill.zIndex = - 1;

    let hand = chart.hands.push(new am4charts.ClockHand());

    // using chart.setTimeout method as the timeout will be disposed together with a chart

    chart.setTimeout(randomValue, 1000);

    function randomValue() {

      hand.showValue(20, 1000, am4core.ease.cubicOut);

    }

  }

  ngOnDestroy() {

    this.zone.runOutsideAngular(() => {

      if (this.chart) {

        this.chart.dispose();

      }

    });

  }

  ngOnInit() {

    this.callHandling1();

    this.callHandling2()

    this.abandonedCall();

    this.answeredCallIVR();

    this.callDuration();

    this.callPerAgent();

    this.averageSpeedOfAns();

    this.costPerCall();

  }

  callHandling1() {

    var chart = new CanvasJS.Chart("callHandle1", {

      animationEnabled: true,

      axisY: {

        title: "Billions of Barrels",

        titleFontColor: "#4F81BC",

        lineColor: "#4F81BC",

        labelFontColor: "#4F81BC",

        tickColor: "#4F81BC"

      },

      axisY2: {

        title: "Millions of Barrels/day",

        titleFontColor: "#C0504E",

        lineColor: "#C0504E",

        labelFontColor: "#C0504E",

        tickColor: "#C0504E"

      },

      toolTip: {

        shared: true

      },

      legend: {

        cursor: "pointer",

        itemclick: toggleDataSeries

      },

      data: [{

        type: "column",

        name: "Proven Oil Reserves (bn)",

        legendText: "Proven Oil Reserves",

        showInLegend: true,

        dataPoints: [

          { label: "Saudi", y: 266.21 },

          { label: "Venezuela", y: 302.25 },

          { label: "Iran", y: 157.20 },

          { label: "Iraq", y: 148.77 },

          { label: "Kuwait", y: 101.50 },

          { label: "UAE", y: 97.8 }

        ]

      },

      {

        type: "column",

        name: "Oil Production (million/day)",

        legendText: "Oil Production",

        axisYType: "secondary",

        showInLegend: true,

        dataPoints: [

          { label: "Saudi", y: 10.46 },

          { label: "Venezuela", y: 2.27 },

          { label: "Iran", y: 3.99 },

          { label: "Iraq", y: 4.45 },

          { label: "Kuwait", y: 2.92 },

          { label: "UAE", y: 3.1 }

        ]

      }]

    });

    chart.render();

    function toggleDataSeries(e) {

      if (typeof (e.dataSeries.visible) === "undefined" || e.dataSeries.visible) {

        e.dataSeries.visible = false;

      }

      else {

        e.dataSeries.visible = true;

      }

      chart.render();

    }

  }

  callHandling2() {

    var chart = new CanvasJS.Chart("callHandle2", {

      animationEnabled: true,

      axisY: {

        title: "Billions of Barrels",

        titleFontColor: "#4F81BC",

        lineColor: "#4F81BC",

        labelFontColor: "#4F81BC",

        tickColor: "#4F81BC"

      },

      axisY2: {

        title: "Millions of Barrels/day",

        titleFontColor: "#C0504E",

        lineColor: "#C0504E",

        labelFontColor: "#C0504E",

        tickColor: "#C0504E"

      },

      toolTip: {

        shared: true

      },

      legend: {

        cursor: "pointer",

        itemclick: toggleDataSeries

      },

      data: [{

        type: "column",

        name: "Proven Oil Reserves (bn)",

        legendText: "Proven Oil Reserves",

        showInLegend: true,

        dataPoints: [

          { label: "Saudi", y: 266.21 },

          { label: "Venezuela", y: 302.25 },

          { label: "Iran", y: 157.20 },

          { label: "Iraq", y: 148.77 },

          { label: "Kuwait", y: 101.50 },

          { label: "UAE", y: 97.8 }

        ]

      },

      {

        type: "column",

        name: "Oil Production (million/day)",

        legendText: "Oil Production",

        axisYType: "secondary",

        showInLegend: true,

        dataPoints: [

          { label: "Saudi", y: 10.46 },

          { label: "Venezuela", y: 2.27 },

          { label: "Iran", y: 3.99 },

          { label: "Iraq", y: 4.45 },

          { label: "Kuwait", y: 2.92 },

          { label: "UAE", y: 3.1 }

        ]

      }]

    });

    chart.render();

    function toggleDataSeries(e) {

      if (typeof (e.dataSeries.visible) === "undefined" || e.dataSeries.visible) {

        e.dataSeries.visible = false;

      }

      else {

        e.dataSeries.visible = true;

      }

      chart.render();

    }

  }

  abandonedCall() {

    var chart = new CanvasJS.Chart("abandonedCalls", {

      animationEnabled: true,

      theme: "light2", // "light1", "light2", "dark1", "dark2"

      axisY: {

        title: "Reserves(MMbbl)"

      },

      data: [{

        type: "column",

        showInLegend: true,

        legendMarkerColor: "grey",

        legendText: "MMbbl = one million barrels",

        dataPoints: [

          { y: 300878, label: "Venezuela" },

          { y: 266455, label: "Saudi" },

          { y: 169709, label: "Canada" },

          { y: 158400, label: "Iran" },

          { y: 142503, label: "Iraq" },

          { y: 101500, label: "Kuwait" },

          { y: 97800, label: "UAE" },

          { y: 80000, label: "Russia" }

        ]

      }]

    });

    chart.render();

  }

  answeredCallIVR() {

    var chart = new CanvasJS.Chart("answeredCallIVR", {

      animationEnabled: true,

      theme: "light2", // "light1", "light2", "dark1", "dark2"

      axisY: {

        title: "Reserves(MMbbl)"

      },

      data: [{

        type: "column",

        showInLegend: true,

        legendMarkerColor: "grey",

        legendText: "MMbbl = one million barrels",

        dataPoints: [

          { y: 300878, label: "Venezuela" },

          { y: 266455, label: "Saudi" },

          { y: 169709, label: "Canada" },

          { y: 158400, label: "Iran" },

          { y: 142503, label: "Iraq" },

          { y: 101500, label: "Kuwait" },

          { y: 97800, label: "UAE" },

          { y: 80000, label: "Russia" }

        ]

      }]

    });

    chart.render();

  }

  callDuration() {

    var chart = new CanvasJS.Chart("callDuration", {

      theme: "light2",

      animationEnabled: true,

      axisY: {

        includeZero: false,

        title: "Number of Viewers",

        suffix: "mn"

      },

      toolTip: {

        shared: "true"

      },

      legend: {

        cursor: "pointer",

        itemclick: toggleDataSeries

      },

      data: [{

        type: "spline",

        visible: false,

        showInLegend: true,

        yValueFormatString: "##.00mn",

        name: "Season 1",

        dataPoints: [

          { label: "Ep. 1", y: 2.22 },

          { label: "Ep. 2", y: 2.20 },

          { label: "Ep. 3", y: 2.44 },

          { label: "Ep. 4", y: 2.45 },

          { label: "Ep. 5", y: 2.58 },

          { label: "Ep. 6", y: 2.44 },

          { label: "Ep. 7", y: 2.40 },

          { label: "Ep. 8", y: 2.72 },

          { label: "Ep. 9", y: 2.66 },

          { label: "Ep. 10", y: 3.04 }

        ]

      },

      {

        type: "spline",

        showInLegend: true,

        visible: false,

        yValueFormatString: "##.00mn",

        name: "Season 2",

        dataPoints: [

          { label: "Ep. 1", y: 3.86 },

          { label: "Ep. 2", y: 3.76 },

          { label: "Ep. 3", y: 3.77 },

          { label: "Ep. 4", y: 3.65 },

          { label: "Ep. 5", y: 3.90 },

          { label: "Ep. 6", y: 3.88 },

          { label: "Ep. 7", y: 3.69 },

          { label: "Ep. 8", y: 3.86 },

          { label: "Ep. 9", y: 3.38 },

          { label: "Ep. 10", y: 4.20 }

        ]

      },

      {

        type: "spline",

        showInLegend: true,

        yValueFormatString: "##.00mn",

        name: "Season 6",

        dataPoints: [

          { label: "Ep. 1", y: 7.94 },

          { label: "Ep. 2", y: 7.29 },

          { label: "Ep. 3", y: 7.28 },

          { label: "Ep. 4", y: 7.82 },

          { label: "Ep. 5", y: 7.89 },

          { label: "Ep. 6", y: 6.71 },

          { label: "Ep. 7", y: 7.80 },

          { label: "Ep. 8", y: 7.60 },

          { label: "Ep. 9", y: 7.66 },

          { label: "Ep. 10", y: 8.89 }

        ]

      },

      {

        type: "spline",

        showInLegend: true,

        yValueFormatString: "##.00mn",

        name: "Season 7",

        dataPoints: [

          { label: "Ep. 1", y: 10.11 },

          { label: "Ep. 2", y: 9.27 },

          { label: "Ep. 3", y: 9.25 },

          { label: "Ep. 4", y: 10.17 },

          { label: "Ep. 5", y: 10.72 },

          { label: "Ep. 6", y: 10.24 },

          { label: "Ep. 7", y: 12.07 }

        ]

      }]

    });

    chart.render();

    function toggleDataSeries(e) {

      if (typeof (e.dataSeries.visible) === "undefined" || e.dataSeries.visible) {

        e.dataSeries.visible = false;

      } else {

        e.dataSeries.visible = true;

      }

      chart.render();

    }

  }

  callPerAgent() { }

  averageSpeedOfAns() {

    var chart = new CanvasJS.Chart("averageSpeedOfAnswer", {

      animationEnabled: true,

      axisY: {

        title: "Billions of Barrels",

        titleFontColor: "#4F81BC",

        lineColor: "#4F81BC",

        labelFontColor: "#4F81BC",

        tickColor: "#4F81BC"

      },

      axisY2: {

        title: "Millions of Barrels/day",

        titleFontColor: "#C0504E",

        lineColor: "#C0504E",

        labelFontColor: "#C0504E",

        tickColor: "#C0504E"

      },

      toolTip: {

        shared: true

      },

      legend: {

        cursor: "pointer",

        itemclick: toggleDataSeries

      },

      data: [{

        type: "column",

        name: "Proven Oil Reserves (bn)",

        legendText: "Proven Oil Reserves",

        showInLegend: true,

        dataPoints: [

          { label: "Saudi", y: 266.21 },

          { label: "Venezuela", y: 302.25 },

          { label: "Iran", y: 157.20 },

          { label: "Iraq", y: 148.77 },

          { label: "Kuwait", y: 101.50 },

          { label: "UAE", y: 97.8 }

        ]

      },

      {

        type: "column",

        name: "Oil Production (million/day)",

        legendText: "Oil Production",

        axisYType: "secondary",

        showInLegend: true,

        dataPoints: [

          { label: "Saudi", y: 10.46 },

          { label: "Venezuela", y: 2.27 },

          { label: "Iran", y: 3.99 },

          { label: "Iraq", y: 4.45 },

          { label: "Kuwait", y: 2.92 },

          { label: "UAE", y: 3.1 }

        ]

      }]

    });

    chart.render();

    function toggleDataSeries(e) {

      if (typeof (e.dataSeries.visible) === "undefined" || e.dataSeries.visible) {

        e.dataSeries.visible = false;

      }

      else {

        e.dataSeries.visible = true;

      }

      chart.render();

    }

  }

  costPerCall() {

    var chart = new CanvasJS.Chart("costPerCall", {

      animationEnabled: true,

      axisY: {

        title: "Billions of Barrels",

        titleFontColor: "#4F81BC",

        lineColor: "#4F81BC",

        labelFontColor: "#4F81BC",

        tickColor: "#4F81BC"

      },

      axisY2: {

        title: "Millions of Barrels/day",

        titleFontColor: "#C0504E",

        lineColor: "#C0504E",

        labelFontColor: "#C0504E",

        tickColor: "#C0504E"

      },

      toolTip: {

        shared: true

      },

      legend: {

        cursor: "pointer",

        itemclick: toggleDataSeries

      },

      data: [{

        type: "column",

        name: "Proven Oil Reserves (bn)",

        legendText: "Proven Oil Reserves",

        showInLegend: true,

        dataPoints: [

          { label: "Saudi", y: 266.21 },

          { label: "Venezuela", y: 302.25 },

          { label: "Iran", y: 157.20 },

          { label: "Iraq", y: 148.77 },

          { label: "Kuwait", y: 101.50 },

          { label: "UAE", y: 97.8 }

        ]

      },

      {

        type: "column",

        name: "Oil Production (million/day)",

        legendText: "Oil Production",

        axisYType: "secondary",

        showInLegend: true,

        dataPoints: [

          { label: "Saudi", y: 10.46 },

          { label: "Venezuela", y: 2.27 },

          { label: "Iran", y: 3.99 },

          { label: "Iraq", y: 4.45 },

          { label: "Kuwait", y: 2.92 },

          { label: "UAE", y: 3.1 }

        ]

      }]

    });

    chart.render();

    function toggleDataSeries(e) {

      if (typeof (e.dataSeries.visible) === "undefined" || e.dataSeries.visible) {

        e.dataSeries.visible = false;

      }

      else {

        e.dataSeries.visible = true;

      }

      chart.render();

    }

  }

}

**crm-nav-bar component**

**html**

<nav class="navbar navbar-expand-lg navbar-light shadow fixed-top sticky-top">

  <button class="navbar-toggler navbar-toggler-right" type="button" data-toggle="collapse"

    data-target="#navbarTogglerDemo02" aria-controls="navbarTogglerDemo02" aria-expanded="false"

    aria-label="Toggle navigation">

    <span class="navbar-toggler-icon"></span>

  </button>

  <a class="navbar-brand"><img src="assets/Voice.ai-logo.png" alt="logo"></a>

  <div class="collapse navbar-collapse custom-navbar" id="navbarTogglerDemo02">

    <ul class="navbar-nav ml-md-auto">

      <li class="nav-item">

        <a class="nav-link" routerLink="/home" routerLinkActive="active">Home</a>

      </li>

      <li class="nav-item">

        <a class="nav-link" routerLink="/crmoverview" routerLinkActive="active">Overview</a>

      </li>

      <li class="nav-item">

        <a class="nav-link" routerLink="/crmcall" routerLinkActive="active">Call Monitoring</a>

      </li>

    </ul>

    <ul class="navbar-nav ml-auto right-nav">

      <li class="nav-item dropdown">

        <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-toggle="dropdown"

          aria-haspopup="true" aria-expanded="false">

          Johan deo

        </a>

        <div class="dropdown-menu" aria-labelledby="navbarDropdown">

          <a class="dropdown-item" routerLink="/login">logout</a>

        </div>

      </li>

      <li class="nav-item">

        <a class="nav-link" href="#"><img src="assets/user.png" alt="arrow" class="user-img"></a>

      </li>

    </ul>

  </div>

</nav>

**css**

.sticky-top{

    padding-right:50px;

    padding-left:50px;

    background-color: #ffffff !important;

}

img.user-img {

    width:40px;

}

a#navbarDropdown {

    margin-top: 10px;

    }

.navbar-nav {

    text-align: center;

}

 /\*\*\*\*\*MOBILE MEDIA QUERY\*\*\*\*\*/

@media only screen and (max-width: 768px) {

 .form {

     width: 60%;

 }

 .right-nav{

 text-align:center;

 }

 }