**Services**

**auth-service**

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

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

@Injectable({

  providedIn: 'root'

})

export class AuthServiceService {

  constructor(private loginService: LoginServiceService,) { }

  // authenticateUser(user : LoginModel){

  //   return this.loginService.http.post('users/login', user).map(res => res.json());

  // }

}

**ci-agent-com**

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

import { HttpClient, HttpHeaders } from '@angular/common/http';

import { Observable } from 'rxjs';

import { map } from 'rxjs/operators';

@Injectable({

  providedIn: 'root'

})

export class CiAgentComServiceService {

  constructor(private http: HttpClient) { }

  dateRangeFilterFun(dateSelectValue) {

    // let headers = new HttpHeaders();

    // headers.append('Content-Type', '/application/json');

    // return this.http.post("",userauth,{headers:headers}).pipe(map(res => res));

    return this.http.post("",dateSelectValue).pipe(map(res => res));

  }

  selectCountryFun(countryValue) {

    return this.http.post("",countryValue).pipe(map(res => res));

  }

  minPercentFun(minValue) {

    return this.http.post("",minValue).pipe(map(res => res));

  }

  maxPercentFun(maxValue) {

    return this.http.post("",maxValue).pipe(map(res => res));

  }

}

**ci-overview**

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

import { HttpClient, HttpHeaders } from '@angular/common/http';

import { Observable } from 'rxjs';

import { map } from 'rxjs/operators';

@Injectable({

  providedIn: 'root'

})

export class CiOverviewServiceService {

  constructor(private http: HttpClient) { }

  // adding folder function

  addfolderfun(folderValues) {

    return this.http.post("", folderValues).pipe(map(res => res));

  }

  // search folder

  searchfolderfun(folderValues) {

    return this.http.post("", folderValues).pipe(map(res => res));

  }

  // search folder

  selectfolderfun(selectfoldval) {

    return this.http.post("", selectfoldval).pipe(map(res => res));

  }

  // search folder

  datestartendfun(datevalue) {

    return this.http.post("", datevalue).pipe(map(res => res));

  }

  // search folder

  callIDsearchfun(callIdsearchval) {

    return this.http.post("", callIdsearchval).pipe(map(res => res));

  }

  // search folder

  callDurLesfun(calldurlessval) {

    return this.http.post("", calldurlessval).pipe(map(res => res));

  }

  // search folder

  callDurGreatfun(calldurgrval) {

    return this.http.post("", calldurgrval).pipe(map(res => res));

  }

  // search folder

  overtalkserfun(overtalkVal) {

    return this.http.post("", overtalkVal).pipe(map(res => res));

  }

  // search folder

  callsilenceserfun(callsilVal) {

    return this.http.post("", callsilVal).pipe(map(res => res));

  }

  initalApiCall(){

    return this.http.get("http://localhost:3000/data")

  }

}

**login-service**

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

import { HttpClient, HttpHeaders } from '@angular/common/http';

import { Observable } from 'rxjs';

import { map } from 'rxjs/operators';

@Injectable({

  providedIn: 'root'

})

export class LoginServiceService {

  constructor(private http: HttpClient) { }

  loginServiceFun(userauth): Observable<any>{

    let headers = new HttpHeaders();

    headers.append('Content-Type', '/application/json');

    return this.http.post("",userauth,{headers:headers}).pipe(map(res => res));

  }

}

**Index.html**

<!doctype html>

<html lang="en">

<head>

  <meta charset="utf-8">

  <title>VoiceAI</title>

  <base href="/">

  <meta name="viewport" content="width=device-width, initial-scale=1">

  <link rel="icon" type="image/x-icon" href="favicon.ico">

  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">

  <link href="https://fonts.googleapis.com/css?family=Open+Sans&display=swap" rel="stylesheet">

  <link href="https://fonts.googleapis.com/css?family=Roboto:300,400,500&display=swap" rel="stylesheet">

  <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">

  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">

  <link rel="stylesheet" href="https://cdn.anychart.com/releases/8.7.1/css/anychart-ui.min.css" />

</head>

<body>

  <app-root></app-root>

  <script src="https://code.jquery.com/jquery-3.4.1.slim.min.js" integrity="sha384-J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ+n" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js" integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo" crossorigin="anonymous"></script>

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js" integrity="sha384-wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifwB6" crossorigin="anonymous"></script>

  <script src="https://cdn.anychart.com/releases/8.7.1/js/anychart-bundle.min.js"></script>

  <script src="https://cdn.anychart.com/releases/8.7.1/js/anychart-base.min.js"></script>

  <script src="https://cdn.anychart.com/releases/8.7.1/js/anychart-exports.min.js"></script>

  <script src="https://cdn.anychart.com/releases/8.7.1/js/anychart-ui.min.js"></script>

  <script>

    function stopsong() {

    var player = document.getElementById('Player');

    player.pause();

    player.currentTime = 0;/\*\*tried also with audio.currentTime here. Didn't worked \*\*/

}

function forwardAudio() {

    var player = document.getElementById('Player');

    player.currentTime += 10.0; /\*\*tried also with audio.currentTime here. Didn't worked \*\*/

}

function PreviousAudio() {

    var player = document.getElementById('Player');

    player.currentTime -= 10.0; /\*\*tried also with audio.currentTime here. Didn't worked \*\*/

}

  </script>

</body>

</html>

**Angular.json**

{

  "$schema": "./node\_modules/@angular/cli/lib/config/schema.json",

  "version": 1,

  "newProjectRoot": "projects",

  "projects": {

    "VoiceAI": {

      "projectType": "application",

      "schematics": {},

      "root": "",

      "sourceRoot": "src",

      "prefix": "app",

      "architect": {

        "build": {

          "builder": "@angular-devkit/build-angular:browser",

          "options": {

            "outputPath": "dist/VoiceAI",

            "index": "src/index.html",

            "main": "src/main.ts",

            "polyfills": "src/polyfills.ts",

            "tsConfig": "tsconfig.app.json",

            "aot": false,

            "assets": [

              "src/favicon.ico",

              "src/assets"

            ],

            "styles": [

              "./node\_modules/@angular/material/prebuilt-themes/indigo-pink.css",

              "src/styles.css",

              "node\_modules/ng2-daterangepicker/assets/daterangepicker.css"

            ],

            "scripts": [

              "node\_modules/jquery/dist/jquery.min.js"

            ]

          },

          "configurations": {

            "production": {

              "fileReplacements": [

                {

                  "replace": "src/environments/environment.ts",

                  "with": "src/environments/environment.prod.ts"

                }

              ],

              "optimization": true,

              "outputHashing": "all",

              "sourceMap": false,

              "extractCss": true,

              "namedChunks": false,

              "aot": true,

              "extractLicenses": true,

              "vendorChunk": false,

              "buildOptimizer": true,

              "budgets": [

                {

                  "type": "initial",

                  "maximumWarning": "2mb",

                  "maximumError": "5mb"

                },

                {

                  "type": "anyComponentStyle",

                  "maximumWarning": "6kb",

                  "maximumError": "10kb"

                }

              ]

            }

          }

        },

        "serve": {

          "builder": "@angular-devkit/build-angular:dev-server",

          "options": {

            "browserTarget": "VoiceAI:build"

          },

          "configurations": {

            "production": {

              "browserTarget": "VoiceAI:build:production"

            }

          }

        },

        "extract-i18n": {

          "builder": "@angular-devkit/build-angular:extract-i18n",

          "options": {

            "browserTarget": "VoiceAI:build"

          }

        },

        "test": {

          "builder": "@angular-devkit/build-angular:karma",

          "options": {

            "main": "src/test.ts",

            "polyfills": "src/polyfills.ts",

            "tsConfig": "tsconfig.spec.json",

            "karmaConfig": "karma.conf.js",

            "assets": [

              "src/favicon.ico",

              "src/assets"

            ],

            "styles": [

              "./node\_modules/@angular/material/prebuilt-themes/indigo-pink.css",

              "src/styles.css"

            ],

            "scripts": []

          }

        },

        "lint": {

          "builder": "@angular-devkit/build-angular:tslint",

          "options": {

            "tsConfig": [

              "tsconfig.app.json",

              "tsconfig.spec.json",

              "e2e/tsconfig.json"

            ],

            "exclude": [

              "\*\*/node\_modules/\*\*"

            ]

          }

        },

        "e2e": {

          "builder": "@angular-devkit/build-angular:protractor",

          "options": {

            "protractorConfig": "e2e/protractor.conf.js",

            "devServerTarget": "VoiceAI:serve"

          },

          "configurations": {

            "production": {

              "devServerTarget": "VoiceAI:serve:production"

            }

          }

        }

      }

    }

  },

  "defaultProject": "VoiceAI"

}

**Tsconfig.json**

{

  "compileOnSave": false,

  "compilerOptions": {

    "baseUrl": "./",

    "outDir": "./dist/out-tsc",

    "sourceMap": true,

    "declaration": false,

    "downlevelIteration": true,

    "experimentalDecorators": true,

    "module": "esnext",

    "moduleResolution": "node",

    "importHelpers": true,

    "target": "es2015",

    "typeRoots": [

      "node\_modules/@types"

    ],

    "lib": [

      "es2018",

      "dom"

    ]

  },

  "angularCompilerOptions": {

    "fullTemplateTypeCheck": true,

    "strictInjectionParameters": true,

    "allowSyntheticDefaultImports": true

  }

}