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

import { ActivatedRouteSnapshot, CanActivate, CanLoad, Route, Router, RouterStateSnapshot, UrlSegment, UrlTree } from '@angular/router';

import { Observable } from 'rxjs';

import { TokenService } from '../token/token.service';

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

@Injectable({

providedIn: 'root'

})

export class AuthGuard implements CanLoad, CanActivate {

constructor(

private router: Router,

private authServ: AuthService

) {

}

async canLoad(

route: Route,

segments: UrlSegment[]): Promise<boolean> {

const isGoingToLoginPage = route.path === 'login' || 'signup';

if (!this.authServ.isLoggedIn() && !isGoingToLoginPage) {

this.router.navigate(['/login']);

return false;

}

if (this.authServ.isLoggedIn() && isGoingToLoginPage) {

// redirecting user to respective home pages if they try to navigate to login page

if (this.authServ.isAdmin()) {

this.router.navigate(['/admin']);

} else {

this.router.navigate(['/']);

}

return false;

}

if (this.authServ.isLoggedIn() && this.authServ.isAdmin()) {

this.router.navigate(['admin']);

return false;

}

return true;

}

async canActivate(

route: ActivatedRouteSnapshot,

state: RouterStateSnapshot

): | Promise<boolean | UrlTree> {

const isGoingToLoginPage = route.url.some(data => data.path.includes('login'));

if (!this.authServ.isLoggedIn() && !isGoingToLoginPage) {

this.router.navigate(['/login']);

return false;

}

if (this.authServ.isLoggedIn() && isGoingToLoginPage) {

// redirecting user to respective home pages if they try to navigate to login page

if (this.authServ.isAdmin()) {

this.router.navigate(['/admin']);

} else {

this.router.navigate(['/']);

}

return false;

}

if (this.authServ.isLoggedIn() && this.authServ.isAdmin()) {

this.router.navigate(['admin']);

return false;

}

return true;

}

}