router.beforeEach((to, from, next) => {

  const user=JSON.parse(sessionStorage.getItem('user'));

  const token=user ? user.token: null;

  const role=user ? user.role: null;

  if (user) {

      store.commit('setUser',user);

      store.commit('setRole',user.role);

      store.commit('setLoggedIn',true);

    }

  if(to.meta.requiresLogin){

    if(!token){

      next({path:'/login'})

    }else{

      if (role === 'admin') {

        // If the user is an admin, redirect to the admin dashboard

        if (to.path !== '/admin/dashboard') {

          next({ path: '/admin/dashboard' });

        } else {

          next();  // Allow navigation if already at the correct page

        }

      } else if (role === 'professional') {

        // If the user is a professional, redirect to the professional dashboard

        if (to.path !== '/professional/dashboard') {

          next({ path: '/professional/dashboard' });

        } else {

          next();  // Allow navigation if already at the correct page

        }

      } else if (role === 'customer') {

        // If the user is a customer, redirect to the customer dashboard

        if (to.path !== '/customer/dashboard') {

          next({ path: '/customer/dashboard' });

        } else {

          next();  // Allow navigation if already at the correct page

        }

      } else {

        // In case the role is undefined or invalid, redirect to the login page

        next({ path: '/login' });

      }

    }

  }else{next()}

})

async createServiceRequest(id){

            try{

                const url=window.location.origin;

                const result=await fetch(url + `/createServiceRequest/${id}`,{

                    method: "POST",

                    headers: { "Content-Type": "application/json" },

                    body: JSON.stringify({professionalId: this.professionalId}),

                    credentials: 'same-origin',

                });

                if(result.ok){

                    const data=await result.json();

                    console.log(data);

                    this.errorMessage=data.message;

                }else {

                    const errorMsg = await result.json();

                    console.log("Failed to create Service request : ", errorMsg);

                    this.errormessage="Failed to create Service request";

                  }

            }catch (error) {

                console.log("Fetch error:", error);

                this.errormessage="Fetch error:", error;

              }

        },