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

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

import { CartItem } from 'src/app/cart-item';

import { Product } from 'src/app/product';

import { CartService } from 'src/app/Services/cart.service';

import { UserService } from 'src/app/Services/user.service';

@Component({

selector: 'app-get-product',

templateUrl: './get-product.component.html',

styleUrls: ['./get-product.component.css']

})

export class GetProductComponent implements OnInit {

category!: string;

product!: Product[];

name!: string;

page: number = 1;

count: number = 0;

tableSize: number = 7;

constructor(private route: ActivatedRoute, private userService: UserService, private cartService: CartService, private router: Router) {

}

ngOnInit(): void {

this.category = this.route.snapshot.params['category'];

this.showMedicineByCategory();

}

onTableDataChange(event: any) {

this.page = event;

}

showMedicineByCategory() {

if (this.category == 'All-Medicines') {

this.userService.getAllMedicine().subscribe({

next: (data) => {

this.product = data;

this.product.forEach((p) => {

p.img = 'data:image/jpeg;base64,' + p.productImage.imageData;

})

}, error: (error) => {

console.log(error);

alert('No Medicines Found');

}

})

} else {

this.userService.getMedicineByCategory(this.category).subscribe({

next: (data) => {

this.product = data;

this.product.forEach((p) => {

p.img = 'data:image/jpeg;base64,' + p.productImage.imageData;

})

}, error: (error) => {

console.log(error);

alert('No Medicines Found');

}

})

}

}

onSearch(name: string) {

if (name != undefined) {

console.log('navigating to search url');

let url = "/user/search/product/" + name;

this.router.navigateByUrl(url);

} else {

console.log('please enter a name');

}

}

sortByPriceLowToHigh() {

this.product.sort((a, b) => a.price - b.price);

}

sortByPriceHighToLow() {

this.product.sort((a, b) => b.price - a.price);

}

sortByNameAscending() {

this.product.sort((a, b) => a.name.localeCompare(b.name));

}

sortByNameDescending() {

this.product.sort((a, b) => b.name.localeCompare(a.name));

}

addToCart(product: Product) {

const cartItem = new CartItem(product);

this.cartService.addToCart(cartItem);

}

}