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

import { Subject } from 'rxjs';

import { CartItem } from '../cart-item';

@Injectable({

providedIn: 'root'

})

export class CartService {

cartItems: CartItem[] = [];

totalPrice: Subject<number> = new Subject<number>();

totalQuantity: Subject<number> = new Subject<number>();

constructor() { }

addToCart(theCartItem: CartItem) {

//check if the cart item is already in the cart

let alreadyInCart: boolean = false;

let existingCartItem: CartItem | undefined = undefined;

if (this.cartItems.length > 0) {

//find the cart item based on the id

existingCartItem = this.cartItems.find((item) => item.pid === theCartItem.pid);

//check if the cart item is found

alreadyInCart = (existingCartItem != undefined);

}

if (alreadyInCart) {

//increase the quantity

existingCartItem!.quantity++;

} else {

this.cartItems.push(theCartItem);

console.log(this.cartItems);

}

this.calculateTotalPrice();

}

calculateTotalPrice() {

let totalPriceValue: number = 0;

let totalQuantityValue: number = 0;

for (let currentCartItem of this.cartItems) {

totalPriceValue += currentCartItem.quantity \* currentCartItem.price;

totalQuantityValue += currentCartItem.quantity;

}

console.log(`Total price: ${totalPriceValue}, Total quantity: ${totalQuantityValue}`);

this.totalPrice.next(totalPriceValue);

this.totalQuantity.next(totalQuantityValue);

}

decrementQuantity(cartItem: CartItem) {

cartItem.quantity--;

if (cartItem.quantity === 0) {

this.remove(cartItem);

} else {

this.calculateTotalPrice();

}

}

remove(cartItem: CartItem) {

const itemIndex = this.cartItems.findIndex(tempCartItem => tempCartItem.pid === cartItem.pid);

if (itemIndex > -1) {

this.cartItems.splice(itemIndex, 1);

this.calculateTotalPrice();

}

}

}