Debugging

Index.js dosyasınında önemli olduğunu düşündüğüm kısımlarına breakpoint koyuyorum.

- addltem() fonksiyonunda:
 - const product = products.find(p => p.id === productId); => product array de var mı?
 - if (product.stock <= quantity) { => stok doğru şekilde artıyor veya azalıyor mu?

```
▼ Watch
              this.items = [];
this.total = 0;
this.discountApplied = false;
                                                                               ▼ Breakpoints

    Pause on uncaught exceptions

           addItem(productId, quantity = 1) { productId = 2, quant
                                                                                🕶 💶 index.js
                    const product = Dproducts.Dfind(p => p.Did ===
20

✓ const product = products.find(p => p.id === pro... 20

✓ if (product.stock <= quantity) { // < yerine <=.</p>
                    if (!product) { | product = {id: 2, name: 'Telefc
    throw new Error('Ürün bulunamadı!');
                                                                                ✓ const itemIndex = this.items.findIndex(item => ... 54

✓ return sum + item.price; // quantity çarpımı un... 79

✓ if (cartElement && totalElement) {
                    if (product.stock <= quantity) { // < yerine <=
                         throw new Error('Yetersiz stok!');
27
28
29
30
31
32
33
34
35
                                                                               ▼ Block
                    const existingItem = this.items.find(item => ite

▼ Local

                                                                                this: ShoppingCart
                    if (existingItem) {
                        existingItem.quantity += quantity;
                                                                                   quantity: 1
                                                                                → Script
                        this.items.push({
                                                                               → Global
                            productId,
name: product.name,
price: product.price
                                                                                ▼ Call Stack
```

removeltem() fonksiyonunda:

o product.stock += 1; sabit değer verilmesi doğru mu?

```
62
63
if (product) {
64
product.stock += 1; // item.quantity yerine sa
65
}
```

- calculateTotal() fonksiyonunda:
 - return sum + item.price; burada quantity ile çarpılması gerektiği unutulmuş.

Breakpoint'leri belirledikten sonra source sekmesinde uygulamayı test ediyorum. Sepete ekle, sil ve toplam fiyatı kontrol ediyorum. Hatalar ve düzenlenmiş halleri:

• **Stok Doğrulama Hatası:** Sepete eklenen miktarla stok aynıysa o ürünün eklenmesini önlüyor.

 Yanlış Toplam Hesaplama: Ürünün fiyatını miktarla çarpmadan ekliyor. Aynı üründen birden fazla ekleyince hesaplama da hata olşuyor.

 Ürün silme sırasında stoğa geri yükleme: Kaldılan miktar yerine her zaman +1 yüklüyor.

```
if (product) { product = {id: 4, nam product.stock += item.quantity; / } 

}

this.items.splice(itemIndex, 1); 
this.calculateTotal(); 
this.updateUI(); 
} catch (error) {

**Scope

**Block

**Item: {productId: 4, name: 'Kulaklık', price: 1000, quantity: 2} 
itemIndex: 0

**ProductId: 4, name: 'Kulaklık', price: 1000, stock: 15} 
**Local

**ItemIndex: 0

**ProductId: 4, name: 'Kulaklık', price: 1000, stock: 15} 
**Local

**ItemIndex: 0

**ProductId: 4, name: 'Kulaklık', price: 1000, stock: 15} 
**Local

**ProductId: 4, name: 'Kulaklık', price: 1000, stock: 15} 
**Local

**ProductId: 4, name: 'Kulaklık', price: 1000, stock: 15} 
**Local

**ProductId: 4, name: 'Kulaklık', price: 1000, stock: 15} 
**Local
```

Watch Paneli ile Breakpoint Testi

- addltem() fonksiyonu için düzeltmeden öncesi ve sonrası:
 - product.stock => doğru şekilde azalıyor mu?
 - this.items => doğru ürünü içeriyor mu?

```
throw new Error('Ürün bulunamadı!');
                                                            ▼ Watch
                                                            this.items: Array(0)
if (product.stock <= quantity) { // < yerine <= ku</pre>
    throw new Error('Yetersiz stok!');
                                                            ▼ Breakpoints

    Pause on uncaught exceptions

const existingItem = this.items.find(item => item.
                                                           Pause on caught exceptions

○ Paused on breakpoint

             (!product) { product = {id: 2, na
              throw new Error('Ürün bulunamadı!
                                                  ▼ Watch
          if (product.stock < quantity) { // <</pre>
                                                    ▶ 0: {productId: 2, name: 'Telefon', price: 8000, quantity: 1}
              throw new Error('Yetersiz stok!')
                                                    ▶ [[Prototype]]: Array(0)
                                                    product.stock: 10
```

- removeltem() fonksiyonu için düzeltmeden öncesi ve sonrası:
 - o product.stock => kaldırıldıktan sonra güncelleniyor mu?
 - this.items => doğru şekilde artıyor mu?

```
const item = this.items[itemIndex];
                                              ▼ Watch
const product = products.find(p => p.
                                              ▼ this.items: Array(2)
                                               > 0: {productId: 4, name: 'Kulaklık', price: 1000, quantity: 1}
> 1: {productId: 2, name: 'Telefon', price: 8000, quantity: 1}
if (product) { product = {id: 4, nam
    product<mark>.stock += 1;</mark> // item.quant
                                               [[Prototype]]: Array(0)
this.items.splice(itemIndex, 1);
                                               product.stock: 15
this.calculateTotal():
const item = this.items[itemIndex];
const product = products.find(p => p.

→ this.items: Array(1)

if (product) { product = {id: 2, nam
                                                ▶ 0: {productId: 2, name: 'Telefon', price: 8000, quantity: 2}
     product.stock += item.quantity; /
                                                > [[Prototype]]: Array(0)
                                               ▼ Breakpoints
this.items.splice(itemIndex, 1);
```