

Insider Final Project (Debugging)

Furkan Boztepe

- 26. kod satırında `product.stock <= quantity` olan kısımda ürünün stoğu adet ile eşit olduğunda yetersiz stok hatası atıyor fakat bu küçük eşittir yerine küçüktür işareti olmalı ki eşit olduğunda stok var olsun.

```
17  
18 addItem(productId, quantity = 1) { productId = 2, quantity = 1  
19   try {  
20     const product = products.find(p => p.id === productId); product = {id: 2, name: 'Telefon', price: 8000, stock: 10}, productId = 2  
21  
22     if (!product) { product = {id: 2, name: 'Telefon', price: 8000, stock: 10}  
23       throw new Error('Ürün bulunamadı!');  
24     }  
25  
26     if (product.stock <= quantity) { // < yerine <= kullanıldı product = {id: 2, name: 'Telefon', price: 8000, stock: 10}, quantity = 1  
27       throw new Error('Yetersiz stok!');  
28     }  
29   }
```

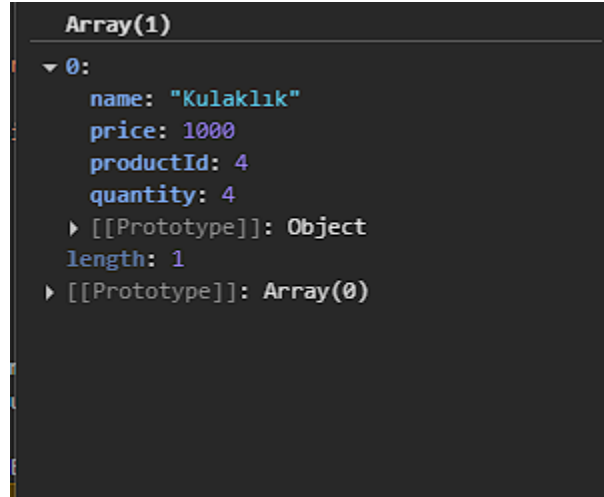
- 78. kod satırındaki `calculateTotal` metodunda sepetteki ürünlerin toplam fiyatını hesaplarken miktar çarpımı yapmıyor bu yüzden toplam fiyat yanlış hesaplanıyordu. Bunu düzeltmek için toplamı fiyat ve miktar ile çarpmak gerekli.

```
78 calculateTotal() {  
79   this.total = this.products.reduce((sum, item) => { sum = 0, item = {productId: 2, name: 'Telefon', price: 8000, quantity: 4}  
80     debugger;  
81     return sum + item.price; // quantity çarpımı unutuldu  
82   }, 0);
```

- 84. kod satırında eğer %10 indirim uygulanmışsa toplam fiyatı 0.1 ile çarpıyor yani bu da toplam fiyatı %10'una indiriyor, indirimler fiyattan düşürüldüğü için 0.9 ile çarpılması gerekli. Bu hatayı arayüzdeki fiyat değişiminden anladım ve debugger ile bir 1000 TL'lik bir ürün aldığım zaman onun total fiyatını 100'e düşürüyordu.

```
82   }, 100  
83   if (this.discountApplied && this.total > 0) {  
84     this.total *= 0.1;  
85     debugger;  
86   }  
87 }  
88  
89 applyDiscount(code) {  
90   if (code === 'INDIRIM10' && !this.discountApplied) {  
91     debugger;  
92     this.discountApplied = true;  
93     this.calculateTotal();  
94     this.updateUI();  
95     this.showMessage('İndirim uygulandı!');  
96   } else {  
97     this.showError('Geçersiz indirim kodu!');  
98   }  
99 }
```

- UI tarafında farkettiğim bir hata sepete ürün eklerken adetinin kaç olduğunu görmeden ürün kartında sabit fiyat gösteriyor. Bunu engellememiz için 115. kod satırında $\${item.price * item.quantity}$ olmalı ki fiyatı adet ile çarpıp doğru toplamı alalım.



- 68. kod satırında ürünü silme işlemi yaparken stoğunu 1 arttırıyor fakat sepet içerisinde kaç adet ürün olduğunu baz almıyor. Bu hatayı düzeltmek için `product.stock += 1` yerine `product.stock += item.quantity` olarak yazmalıyız.

```
52 removeItem(productId) { productId = 4
53   debugger;
54   try {
55     const itemIndex = this.items.findIndex( itemIndex = 0
56       (item) => item.productId === productId item = {productId: 4, name: 'Kulaklık', price: 1000, quantity: 11}, productId = 4
57     );
58
59     if (itemIndex === -1) { itemIndex = 0
60       throw new Error("Ürün sepette bulunamadı!");
61     }
62
63     const item = this.items[itemIndex]; item = {productId: 4, name: 'Kulaklık', price: 1000, quantity: 11}, itemIndex = 0
64     const product = products.find((p) => p.id === productId); product = {id: 4, name: 'Kulaklık', price: 1000, stock: 15}, productId = 4
65
66     if (product) { product = {id: 4, name: 'Kulaklık', price: 1000, stock: 15}
67       debugger;
68       product.stock += 1; // item.quantity yerine sabit değer
69     }
```

- Geçersiz bir indirim kuponu girdiğimiz zaman sayfada sürekli geçersiz indirim kuponu yazdırıyordu bu hatanın önüne geçmek için 126. kod satırındaki `errorElement`'in `textContent`'ini hata mesajına eşitledik.

```
123 showError(message) {
124   debugger;
125   const errorElement = document.getElementById("error");
126   if (errorElement) {
127     debugger;
128     errorElement.textContent = message;
129   }
130 }
```

- 194. kod satırındaki addToCart metodu içerisinde 196. satırda parametre içerisindeki undefined tanımlı hatalı bir tanımdır ve kodun okunabilirliğini netliğini azaltıyor. Bunun yerine parametre içerisinde undefined yerine quantity göndeririz ve ürünün adetini her sepete eklendiğinde 1 arttırırız.

```
194     addToCart(productId, quantity = 1) { productId = 4, quantity = 1
195     debugger;
196     window.cart.addItem(productId, quantity);
197     document.dispatchEvent(new Event("stockUpdate"));
198   }
199 }
200
```

- 125. satırdaki showError metodunda kullanıcı yanlış bir indirim kuponu girdiğinde Geçersiz indirim kodu sayfada sürekli kalıyordu bunu engellemek için de metod içerisinde setTimeout ekledim.

```
125     showError(message) {
126       const errorElement = document.getElementById("error");
127       if (errorElement) {
128         errorElement.textContent = message;
129         setTimeout(() => {
130           errorElement.textContent = "";
131         }, 3000);
132       }
133     }
134
```

- Bir ürünü sepete ekle dedikten sonra stoğundaki adet sayısı eksilmiyor ve sepete bu üründen sınırsız şekilde ekleme yapıyorduk. addItem metoduna product.stock -= quantity; eklemesi yapıldığında artık ürünün stoğu kadar adet eklemesi yapılıyor.

Laptop Fiyat: 15000.00 TL Stok: 5 Sepete Ekle	Telefon Fiyat: 8000.00 TL Stok: 10 Sepete Ekle	Tablet Fiyat: 5000.00 TL Stok: 0 Sepete Ekle	Kulaklık Fiyat: 1000.00 TL Stok: 15 Sepete Ekle	Mouse Fiyat: 500.00 TL Stok: 20 Sepete Ekle
---	--	--	---	---

Sepetim

Tablet	8 adet	40000 TL	Sil
--------	--------	----------	-----

- Kullanıcı deneyimi için sepete eklenen ürünlerin adedi kaç olursa olsun sil butonuna tıklandığında adetini 1 eksiltmiyor direkt tamamını siliyor. Daha iyi bir kullanıcı deneyimi için bunu şu şekilde düzelttim.

```
54 removeItem(productId) { productId = 4
55   debugger;
56   try {
57     const itemIndex = this.items.findIndex( itemIndex = 0
58       (item) => item.productId === productId item = {productId: 4, name: 'Kulaklık', price: 1000, quantity: 3}, productId = 4
59     );
60
61     if (itemIndex === -1) { itemIndex = 0
62       throw new Error("Ürün sepette bulunamadı!");
63     }
64
65     const item = this.items[itemIndex]; item = {productId: 4, name: 'Kulaklık', price: 1000, quantity: 3}, itemIndex = 0
66     const product = products.find((p) => p.id === productId); product = {id: 4, name: 'Kulaklık', price: 1000, stock: 12}, productId = 4
67
68     if (product) { product = {id: 4, name: 'Kulaklık', price: 1000, stock: 12}
69       debugger;
70       if (item.quantity > 1) {
71         item.quantity -= 1;
72         product.stock += 1;
73       } else {
74         product.stock += item.quantity;
75         this.items.splice(itemIndex, 1);
76       }
77     }
```