

Alışveriş Sepeti Uygulaması Hata Analizi ve Çözüm Raporu

Giriş

Bu rapor, "Advanced-JS-and-Web-APIs(week4)" klasöründeki "debug-Odev3" projesinde bulunan alışveriş sepeti uygulamasında gerçekleştirdiğim debugging işlemlerini ve tespit ettiğim hataları detaylı olarak açıklamaktadır.

Debugging Metodolojisi

Debugging işlemlerini Chrome DevTools kullanarak gerçekleştirdim. Öncelikle uygulamanın genel davranışını gözlemleyerek hataları tespit ettim, ardından Sources panelinde ilgili kodlara breakpoint'ler ekleyerek adım adım çalışmayı inceledim.

Tespit Edilen Hatalar ve Çözümleri

Sorun 1: Sepete Ürün Eklenince Stok Azalmıyor

Kullanıcı sepete ürün eklediğinde, ürünün stok miktarı azalmıyor. Bu durum, aynı ürünün stok miktarından fazla sepete eklenmesine sebep oluyor ve ekleme işlemi hem durmuyor hem de sepete eklenen ürün miktarı stokta kalmadığı halde artmaya devam ediyor.

Debugging Süreci:

- Chrome DevTools'da *addItem* metoduna breakpoint ekleyerek, ürün ekleme işlemi sonrasında *product.stock* değerinin değişip değişmediği kontrol ettim.
- Sonuçta, *addItem* metodunda ürün stok azaltan kod satırının bulunmadığını fark ettim.

Çözüm: Aşağıdaki satır eklenmeli.

```
//SORUN1: Sepete ürün eklenince stok azalmıyor(çözüm satırı)|  
product.stock -= quantity;
```

Doğrulama: Değişiklik sonrası ürün eklediğimde, stok miktarının doğru şekilde azaldığını ve UI'da güncellendiğini ve hatta "Yetersiz stok!" uyarısının çıktığını gözlemledim.

Sorun 2: Stok Kontrolü Hatalı - Stok 0 Olmadan Yetersiz Stok Uyarısı Veriyor

Debugging Süreci:

1. Ürün stok miktarı 1 olduğunda "Yetersiz stok!" uyarısı verdiğini fark ettim.
2. addItem metodundaki stok kontrolü koşulunu inceledim.
3. Chrome DevTools'da breakpoint ekleyerek değişkenlerin değerlerini takip ettim.
4. Koşulda *product.stock <= quantity* ifadesinin kullanıldığını gördüm.

Sorunun Analizi: Bu koşul, stok miktarı eklenmek istenen miktar ile eşit olduğunda bile hata fırlatıyor. Örneğin, stokta 1 ürün varsa ve 1 ürün eklemek istediğimizde bile "Yetersiz stok!" hatası alıyoruz. Oysa stok miktarı kadar ürün ekleyebilmeliyiz.

Çözüm: Koşulu, stok miktarının eklenmek istenen miktardan küçük olması durumunda hata verecek şekilde değiştirdim:

```
//SORUN2:Stok Kontrolü Hatalı - Stok 0 Olmadan Yetersiz Stok Uyarısı Veriyor(çözüm)
if (product.stock < quantity) {
  throw new Error("Yetersiz stok!");
}
```

Doğrulama:

- Stok miktarı 1 olan bir ürün için 1 adet eklemeyi denedim.
- Önceki kod ile "Yetersiz stok!" hatası alırken, düzeltme sonrası ürünün başarıyla eklendiğini gördüm.
- Stok miktarı 0'a düştüğünde "Sepete Ekle" butonunun devre dışı kaldığını doğruladım.
- Stok miktarından fazla ürün eklemeye çalıştığımda hala "Yetersiz stok!" hatası aldığımı doğruladım.

Sorun 3: Toplam Fiyat Hesaplaması Hatalı

Debugging Süreci:

1. `calculateTotal` metoduna breakpoint ekledim.
2. Watch paneline `this.total`, `item.price` ve `item.quantity` değişkenlerini ekledim.
3. Sepete farklı miktarlarda ürünler ekleyerek toplam fiyatın nasıl hesaplandığını inceledim.
4. Toplam hesaplanırken ürün miktarının dikkate alınmadığını fark ettim.

Çözüm: Toplam hesaplanırken ürün miktarı ile fiyatın çarpılmasını ayarladım:

```
calculateTotal() {  
  //SORUN3:Toplam Fiyat Hesaplaması Hatalı(çözüm)  
  this.total = this.items.reduce((sum, item) => {  
    return sum + item.price * item.quantity;  
  }, 0);  
}
```

Doğrulama: Değişiklik sonrası, sepete 2 adet ürün eklediğimde toplam fiyatın doğru hesaplandığını (birim fiyat × miktar) Watch panelinde gözlemledim.

Sorun 4: İndirim Uygulaması Hatalı

Debugging Süreci:

1. `applyDiscount` ve `calculateTotal` metodlarına breakpoint ekledim.
2. İndirim kodunu uyguladım ve kodun çalışmasını adım adım takip ettim.
3. Watch paneline `this.total` ve `this.discountApplied` değişkenlerini ekledim.
4. İndirim uygulandığında toplam fiyatın %10 azalması yerine %90 azaldığını fark ettim.

Çözüm: İndirim hesaplamasını düzelttim:

```
if (this.discountApplied && this.total > 0) {  
  this.total *= 0.9; //SORUN4: %10 indirim için 0.9 ile çarpılmalı  
}
```

Doğrulama: Değişiklik sonrası, indirim kodunu uyguladığımda(INDIRIM10) toplam fiyatın %10 azaldığını Watch panelinde gözlemledim.

Sorun 5: Ürün sepetten kaldırılıp tekrar eklenince stok 1 artıyor

Debugging Süreci:

1. Sepetten ürün sildiğimde stok miktarının her zaman sadece 1 arttığını fark ettim.
2. `removeItem` metoduna breakpoint ekledim.
3. "Watch" panelini kullanarak `item.quantity` değerini izledim.
4. Sorun: Stok artırma kısmında gerçek miktar (`quantity`) yerine, sabit değer (1) kullanılıyor.

Çözüm: `removeItem` metodundaki stok artırma kodunu düzelttim:

```
if (product) {  
  //SORUN5:çözüm-sabit değer yerine item.quantity  
  product.stock += item.quantity;  
}
```

Sorun 6: Geçersiz indirim kodu hata mesajı birikimi

Debugging Süreci:

1. `showError` metoduna breakpoint ekledim.
2. Geçersiz bir indirim kodu girip "İndirimi Uygula" butonuna bastım.
3. DOM'da `errorElement.textContent` değerini inceledim.
4. Sorun: Mevcut mesaja ekleme yapılıyor (`+=`), üzerine yazılmıyor (`=`)

Çözüm: `showError` metodunu düzeltip ve timeout ekledim:

```
showError(message) {  
  const errorElement = document.getElementById("error");  
  if (errorElement) {  
    errorElement.textContent = message; // += yerine = kullan  
  
    // Mesajı belirli bir süre sonra temizle  
    setTimeout(() => {  
      errorElement.textContent = "";  
    }, 3000);  
  }  
}
```

Sorun7: Stok Güncelleme Sorunu

Sorun: Ürün sepete eklenip silindiğinde stok bilgisi otomatik olarak güncellenmemektedir. Stok güncellemesi sadece başka bir ürün sepete eklendiğinde gerçekleşmektedir.

Debugging Süreci:

1. *addItem* metodunda *product.stock -= quantity;* satırının yanına bir breakpoint ekledim.
2. *removeItem* metodunda *this.items.splice(itemIndex, 1);* satırının yanına tıklayarak bir breakpoint ekledim.
3. Herhangi bir ürün için "Sepete Ekle" butonuna tıkladım.
4. *this.updateUI();* satırından sonra kodun devam etmesine izin verdim.
5. Konsol panelinde *document.dispatchEvent(new Event("stockUpdate"));* komutunun çalıştırılmadığını gözlemledim.
6. *stockUpdate* olayının sadece *setUpEventHandlers* metodunda tanımlandığını, ancak *addItem* ve *removeItem* metotlarında tetiklenmediğini gördüm.

Çözüm: Her iki metoda da stok güncellemesi sonrası "*stockUpdate*" olayını tetikleyen kod eklenmelidir:

addItem:

```
this.calculateTotal();
this.updateUI();
//SORUN7:(1)
document.dispatchEvent(new Event("stockUpdate")); // Bu satır eklendi
```

removeItem:

```
this.calculateTotal();
this.updateUI();
//SORUN7:(2)
document.dispatchEvent(new Event("stockUpdate")); // Bu satır eklendi
catch (error) {
```