

1. console.log() Kullanımı ile Debugging

- console.log(product) kullanarak addItem() fonksiyonunda ürünlerin sepete eklenip eklenmediğini kontrol ettim. Burada product nesnesinin doğru şekilde alınıp alınmadığını ve stok miktarını doğru şekilde kontrol edilip edilmediğini gözlemledim.
- console.log(this.items) ile sepetteki ürünlerin doğru şekilde güncellenip güncellenmediğini inceledim. Bu, ürün ekleme işleminde oluşabilecek mantıksal hataları yakalamama yardımcı oldu. Örneğin, birden fazla ürün eklenmesine rağmen ürünlerin stokları doğru şekilde güncellenmediğini gözlemledim.
- console.log(this.total) kullanarak calculateTotal() fonksiyonunun çıktısını kontrol ettim. Burada, toplam hesaplama fonksiyonunun doğru çalışıp çalışmadığını test ettim.

Hata Tespiti:

- Hata: calculateTotal() fonksiyonunda price * quantity çarpımı unutulmuştu. Bu hatayı fark ettikten sonra, toplam tutarın doğru hesaplanmadığını gördüm.
- Hata: this.total *= 0.1; kısmı yanlış şekilde uygulanıyordu. Bu satır toplamı %90 indirim sokuyordu, doğru olması gereken ise %10 indirim olmalıydı. Buradaki hatayı console.log(this.total) ile inceledim.
- removeItem() fonksiyonunda, stok güncellenmesinde bir hata olduğunu fark ettim. product.stock += 1; satırında item.quantity yerine sabit bir değer artırılıyordu. Bu hata, bir müşteri birden fazla aynı ürünü alıp sepete çıkarırken, sadece 1 adet geri eklenmesine yol açıyordu.

2. Breakpoints ile Debugging

- Chrome DevTools'un Sources sekmesinde, addItem() ve removeItem() fonksiyonlarına breakpointler ekledim. Kodun akışını bu breakpointlerle izledim ve her iki fonksiyonda da potansiyel hataları tespit ettim.

Hata Tespiti:

- Hata: if (product.stock <= quantity) kontrolü, stok miktarı tam eşitse bile hata veriyordu. Bu durumu if (product.stock < quantity) olarak düzelttim, çünkü eşitlik durumunda stokta yeterli ürün bulunuyor.
- Hata: removeItem() fonksiyonunda stok güncellenmesi yanlış yapılıyordu. Burada, stok güncelleme işlemi doğru miktarla yapılmalıydı. Bu hatayı düzelttim: product.stock -= item.quantity; olarak değiştirdim.
- Hata: addItem() ve removeItem() fonksiyonlarında bazı durumlarda updateUI() fonksiyonu çağrılmıyordu. Bu da HTML elemanlarının güncellenmemesine yol açıyordu. Bu durumu breakpointlerle tespit ettim ve gerekli yerlerde updateUI() çağrısı ekledim.

3. Network Paneli ile API Çağrılarını İnceleme

- Network panelini kullanarak API çağrılarında herhangi bir hata olup olmadığını kontrol ettim. API çağrılarında bazıları sırasında 500 Internal Server Error alıyordum.

Çözüm: `fetch(url).catch(error => console.error(error))` ekleyerek hata yönetimini güçlendirdim. Bu şekilde, API çağrıları sırasında oluşabilecek hatalar daha iyi kontrol edilebilir oldu ve kullanıcıya hata mesajları doğru şekilde gösterilebildi.

4. LocalStorage ve SessionStorage Denetimi

- Application sekmesinde localStorage ve sessionStorage içeriğini kontrol ettim. Özellikle `sessionStorage.setItem("refetched", "1")` çağrısının bazen çalışmadığını fark ettim.

Çözüm: `sessionStorage.getItem("refetched")` değerini önce kontrol edip ardından `setItem` çağrısını yaptım. Bu şekilde, önceki değerlerin kontrol edilmesi ve yeni değerlerin yalnızca gerekli durumlarda set edilmesi sağladım.