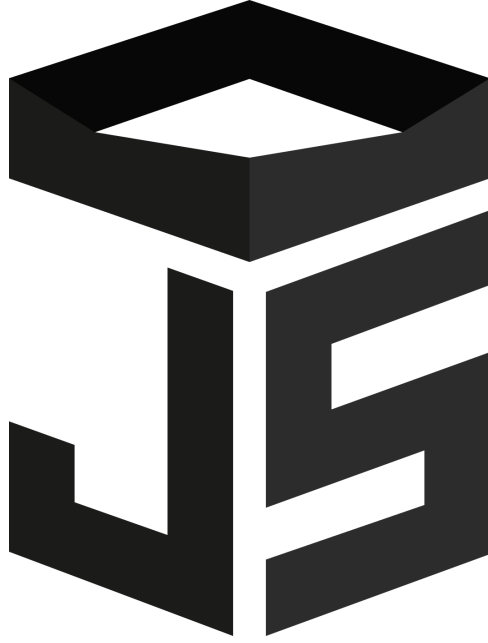


FE-1426P-1



Hedefimiz

Detaylı bir “Alışverişte Alınacaklar Listesi” uygulaması yapalım

Koşullar ve bilgilendirmeler

1. Projemizde arayüzü ayarlamak için “react-bootstrap” ve “styled-components” paketlerini kullanalım
2. Uygulamamız, alışverişe giderken hangi markette, hangi kategoride, hangi ürünü almak istediğimizi belirtip liste oluşturabilmemizi sağlamalı
 - Market örneği: “Migros”, “Teknosa”, “BİM” vs.
 - Kategori örneği: “Elektronik”, “Şarküteri”, “Oyuncak”, “Bakliyat”, “Fırın” vs.
 - Bu marketleri ve kategorileri oluşturmak bize kalmış, istediğimiz şekilde oluşturabiliriz
 - Market ve kategorileri iki ayrı obje array’i olarak tutalım

- Market objesi örneği:

```
{
  "id": 1,
  "name": "Migros"
}
```

- Kategori objesi örneği:

```
{
  "id": 1,
  "name": "Elektronik"
}
```

- id'leri kendi elimizle girebiliriz, sorun değil, otomatik oluşturmaya gerek yok
- Marketleri ve kategorileri React state olarak tutmamıza gerek yok, bu verileri değiştirmeyeceğiz
- Marketleri “shops”, kategorileri ise “categories” isminde array'lere kaydedelim
 - Yukarıdaki örnek objeleri array'lerin içine elimizle kod olarak gireceğiz sadece. Yani array'i oluştururken içi boş bir şekilde oluşturmayacağız, yukarıdaki örneklerdeki gibi objeleri elimizle array'in içine yazarak oluşturacağız.
 - Bu işlemlerin sonucunda elimizde “shops” isminde bir array'de market listesi, “categories” ismindeki başka bir array'de ise kategori listesi olmalı

3. Bir ürün ekleme alanı oluşturalım

- Form input'lar aracılığıyla kullanıcıdan yeni ürün bilgisi alacağız, ardından girilen o ürünleri bir “products” state'ine kaydedeceğiz
- Ürün eklerken alınacak bilgiler:
 - Ürünün adını yazdığımız bir yazı kutucuğu
 - Ürünün hangi marketten alınacağını seçtiğimiz bir select input'u
 - Ürünün kategorisini seçtiğimiz bir select input'u

- Kategori ve market seçtirirken ilgili kategorinin veya marketin “id” değerini seçtirtmeliyiz
 - Bu listeleri select input’unun içine basarken ilgili array’imizden döngüyle basalım, manuel girmeyelim
 - Döngü olarak basarken option’ların “value” kısmına kategorinin veya marketin (artık hangisinin döngüsünü o an yapıyor isek) id’sini basalım. Böylece seçtiğimizde id’yi seçmiş olacağız
- Eklediğimiz her bir ürünün ek olarak otomatik oluşturulan rastgele bir id’si olsun
 - Yani ürünün adı (name), hangi marketten alınacağı (shop) ve kategorisinin (category) yanında bir de id’si olsun
 - “products” state’inde array içerisine kaydederken name, shop ve category bilgilerinin yanında bu id ile beraber kaydedeceğiz
 - Yardımcı kaynaklar kısmına bakabiliriz

4. Eklenen ürünler tablo içerisinde bize gösterilsin

- Arayüzü oluşturmak için react-bootstrap’teki tabloları kullanabiliriz
- İlk önce bir tablo örneği oluşturalım, tablo görünümünü tarayıcımızda elde ettikten sonra verilerimizi gösterecek şekilde tabloyu düzenleyebiliriz
- Veri olarak “products” state’indeki array’i tablo oluşturmak için kullanalım
- “products.map(product ⇒)” yaparak tablonun satırlarını teker teker oluşturacağız

5. Ürünün üstüne tıklayınca satın alındı olarak işaretleyelim

- Satıra onClick verip hangi satırdaki ürüne tıklandıysa o ürüne “isBought: true” tarzı bir değer eklememiz lazım “products” state’imizde
- Satın alındı olarak işaretlenince ürünün yazısının üstü çizilsin
 - Yani o satırı temsil eden üründe “isBought” değeri true ise css ile satırın yazılarının üstünü çizelim
- Yardımcı kaynaklara bakabiliriz

6. Ürünler tablosunun en sağına ek bir sütun koyalım. Bu sütun, ürünü silme sütunu olsun

- Bu sütunun içerisine IconButton component'i hazırlayıp koyalım.
 - IconButton componenti, bir silme ikonu içeren kare şeklinde (dikdörtgen değil) bir buton olsun.
 - Butona tıklayınca ilgili ürünü listeden tamamen kaldırsın
- Ürünü products state'inden silmek için, hangi ürünü silmek istiyorsak o ürünün id'sini products.filter yaparak elemeliyiz ve çıkan yeni array'i products state'ine tekrar kaydetmeliyiz
 - Yardımcı kaynaklara bakabiliriz

7. Listedeki tüm ürünler satın alındığı zaman "Alışveriş Tamamlandı" şeklinde bir alert tetikleyelim

- Sadece listede ürün varken ve tüm ürünler satın alınmış bir durumdayken tetiklenmeli
- Alışverişin tamamlanması durumunu ayrı bir state olarak TUTMAYALIM. Sadece alışveriş listesi state'ini kullanıp oradan bu durumu hesaplayarak "Alışveriş Tamamlandı" uyarısını basalım
 - "products" state'i üzerinde array'lerdeki ".every()" özelliğini kullanarak her ürün satın alınmış mı diye kontrol edeceğiz
- Bu alert tetiklendikten sonra listeden bir ürün sildiğimizde tetiklenmemeli
 - Örneğin listede 8 ürün varsa, 7 ürün satın alınmışken 8. ürünün de üstüne tıklanarak satın alındığı zaman "Alışveriş Tamamlandı" alert'ı tetiklenmeli. Fakat 8 ürünün 8i de satın alınmışken bir ürünü listeden sildiğimizde yani 7/7 olduğunda "Alışveriş Tamamlandı" alert'ı tetiklenmemeli. Aynı şekilde bir ürün daha silinirse yani 6/6 olursa alert tekrar tetiklenmemeli
 - Yani bir ürün silinmeden önce eğer listedeki her ürün halihazırda satın alındıysa, ürün silindikten sonra da aynı şekilde satın alınmış olacağı için son liste durumunda tüm ürünler yine satın alınmış gözükeceğinden dolayı yeni bir alert bastırmamalıyız, zaten önceden de her ürün satın alınmıştı çünkü
 - Kontrollerimizi buna göre yazmalıyız yani, buglu olmamalı uygulamamız

- React'ta state'i güncellerken setState yaptığımız fonksiyonun içinde o state'in önceki haline erişebiliyorduk. Bunu kullanarak önceki hali güncellemeden hemen önce (state'in güncellenmesi için return etmeden hemen önce) alışverişin tamamlanması koşulu uyuyorsa alert patlatabiliriz
- Alışveriş tamamlandığı zaman alert ile birlikte bir konfeti patlatalım :)
 - Yardımcı kaynaklara bakabiliriz

Yardımcı Kaynaklar

- 🗝️ "npm nanoid"
- 🗝️ "react filter array of objects"
- 🗝️ "react updating objects inside arrays"
- 🗝️ "js confetti npm"