

KOCAELİ ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

BİTİRME TEZİ

BUSINESS TO CONSUMER SİSTEMLER (B2C) İÇİN MAKİNE
ÖĞRENMESİ YÖNTEMİ İLE ÖNERİ SİSTEMİ

YAZARLAR: SAMET YAVUZ , BUĞRA BURAK ÖNAL
DANIŞMAN: PROF. DR. KEREM KÜÇÜK

KOCAELİ 2023

İÇİNDEKİLER

ŞEKİLLER DİZİNİ	3
GİRİŞ	4
1.FREQUENT PATTERN GROWTH (FP-GROWTH).....	5
1.1 FP AĞAÇLARI	7
2. FP AĞACI ALGORİTMASI	8
3. İMPLEMENTASYON.....	9
3.1. VERİ SETİ.....	10
3.2 VERİ ÖNİŞLEME	11
4. PROJE.....	12
4.1 PROJENİN İŞLEYİŞİ	13
4.1.1 Giriş Ekranı.....	13
4.1.2 Ana Ekran	14
4.1.3 Sepet Ekranı	16
4.1.4 Sipariş Alındı Ekranı.....	18
5. KAYNAKÇA	19

ŞEKİLLER DİZİNİ

Şekil 2.1 Fp Ağacı Pseudocode	8
Şekil 3.1 İmplementasyon akışı	9
Şekil 3.2 Model Sonuç	9
Şekil 4.1 Kullanıcı Giriş Ekranı	13
Şekil 4.2 Önerilen Ürün	14
Şekil 4.3 Önerilmeyen Ürün	14
Şekil 4.4 Ana Ekran	15
Şekil 4.5 Sepet Ekranı	16
Şekil 4.6 Sepetteki Ürünlerin Çıkarılması Sonrası Sepet Ekranı	17
Şekil 4.7 Siparisler.json Dosyası	17
Şekil 4.8 Sipariş Alındı Ekranı	18

GİRİŞ

B2C (Business-to-Consumer), bir işletme ile tüketici arasındaki ticaret ilişkisini ifade eden bir terimdir. B2C sistemleri ise bu ticaret ilişkisini kolaylaştıran çevrimiçi platformları ifade eder. Bu platformlar, işletmelerin ürün veya hizmetlerini doğrudan tüketicilere sunmalarını sağlar. B2C sistemleri, geleneksel perakende satış yöntemlerine alternatif olarak ortaya çıkmıştır ve özellikle internetin yaygınlaşmasıyla birlikte büyük bir popülerlik kazanmıştır. Bu sistemler, işletmelerin geniş bir ürün yelpazesini çevrimiçi olarak sergilemelerine, tüketicilerin ürünleri incelemelerine ve satın almalarına olanak tanır. B2C sistemlerinin temel amacı, işletmelerin tüketiciye doğrudan erişim sağlamasını ve satışlarını arttırmasını sağlamaktır. Tüketiciler, b2c sistemleri aracılığıyla ürünler hakkında bilgi edinebilir, ürünleri karşılaştırabilir, fiyatları kontrol edebilir ve satın alma işlemlerini gerçekleştirebilir. Bu sistemler, tüketicilere kolaylık, çeşitlilik ve erişilebilirlik sunarak alışveriş deneyimini geliştirir. B2C sistemleri, işletmelere bir dizi avantaj sağlar. İşletmeler, çevrimiçi platformlar üzerinden geniş bir kitleye ulaşabilir ve uluslararası pazarlara açılabilir. Aynı zamanda, işletmeler müşteri davranışlarını ve tercihlerini daha iyi anlamak için veri analitiği ve kişiselleştirme stratejilerini kullanabilirler. B2C sistemleri, işletmelere müşteri ilişkilerini güçlendirmek, marka sadakatini artırmak ve pazarlama stratejilerini daha etkili bir şekilde uygulamak için birçok fırsat sunar. Tüketicilerin güvenini kazanmak ve korumak, b2c sistemlerinin başarısı için kritik öneme sahiptir. Bu projemizde Türkiye'deki bir şirketin bize verdiği veri setleri ile çalışma yapıp, makine öğrenme teknikleri kullanarak bir öneri sistemi yapmayı amaçladık. Bu teknikleri bir web sitesi üzerinde yaptık.

1.FREQUENT PATTERN GROWTH (FP-GROWTH)

Sık Örüntü Madenciliği, işlemsel verilerde birlikte ortaya çıkan örüntüleri bulma sürecini ifade eder. En önde gelen uygulamalardan biri pazar sepeti analizidir. Perakendeciler, hedefe yönelik promosyonlar, reklamlar yapmak ve ayrıca mağazalarındaki ürünleri fiziksel olarak düzenlemek için müşteriler tarafından sıklıkla birlikte satın alınan ürünleri bulurlar. Bununla birlikte, kümeleme, öneriler oluşturma ve verilerdeki özellikleri bulma konusunda çok çeşitli başka uygulamalar da vardır.

Bu tezde Örüntü Büyümesi adı verilen bir böl ve fethet metodolojisini tanımlıyoruz ve özellikle FP-Büyüme Algoritmasını anlamaya ve uygulamaya odaklanıyoruz.

İlişkilendirme tekniği, Han tarafından ortaya atılan FP-Büyüme Algoritmasına yol açmıştır. Madenciliğin, örüntüler tarafından sık kullanılan örüntülerin tam bir kümesi üzerinde genişletilmiş bir önek ağacı yapısı ile yapıldığı verimli bir yöntemdir.

parça büyümesi. Ağaç yapısı, sık kullanılan örüntüler hakkında sıkıştırılmış bilgileri depolar. Han çalışmasında, Böl ve Fethet yöntemi ve diğer yöntemler sayesinde Bu algoritmayı apriori algoritması gibi sık örüntü madenciliği için kullanılan diğer popüler yöntemlerden daha verimli yapmıştır.

Bu algoritma, girdi veritabanını sıkıştırarak başlar ve böylece bir sık örüntü ağacı örneği geliştirir. Sıkıştırılmış veritabanı daha sonra birkaç koşullu veritabanına bölünür, burada her veritabanı bir benzersiz sık örüntü. Son olarak, her veritabanının madenciliği ayrı ayrı gerçekleştirilir. Bu nedenle, arama maliyetleri önemli ölçüde azalır ve iyi bir seçicilik sunar. FP Growth algoritmasının daha verimli olmasının nedenleri diğer algoritmalara göre:

1. Böl ve Fethet: Madencilik verileri alt veri kümelerine ayrıştırılır tanımlanan sık kalıplara göre. Daha küçük veri tabanlarında daha odaklı arama yapılmasını sağlar.
2. Aday üretimi yoktur. Sonuç olarak hiçbir aday testi gereklidir.
3. Tüm veritabanında tekrarlanan taramalar yok

1.1 FP AĞAÇLARI

Bir sık örüntü ağacı, etiketlenmiş bir kökten oluşur null olarak, öge-önek alt ağaçları kümesi alt ögeler olarak ve sık kullanılan öge başlığı tablosundan oluşur.

Öge-önek alt ağacındaki her bir düğüm şunlardan oluşur üç alan: item-name, count ve node-link, buradan item-name düğümün hangi ögeyi temsil ettiğini kaydeder,count işlem sayısını kaydeder,yolun o noktaya ulaşan kısmı tarafından temsil edilir.düğümü ve FP'deki bir sonraki düğüme düğüm-bağlantı bağlantıları Ağaç, aynı öge adını taşıyan veya varsa null hiçbiridir.

Sık kullanılan öge başlığı tablosundaki her giriş iki alandan oluşur: bir öge adı ve bir düğüm bağlantısı.

2. FP AĞACI ALGORİTMASI

Procedure $FPgrowth^*(T)$

Input: A conditional FP-tree T

Output: The complete set of all FI's corresponding to T .

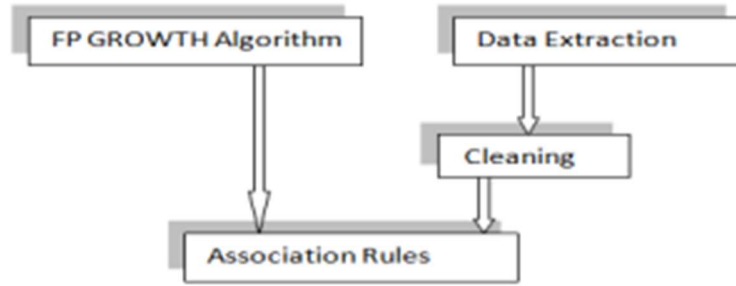
Method:

1. **if** T only contains a single branch B
2. **for each** subset Y of the set of items in B
3. output itemset $Y \cup T.base$ with count = smallest count of nodes in Y ;
4. **else for each** i in $T.header$ **do begin**
5. output $Y = T.base \cup \{i\}$ with $i.count$;
6. **if** $T.FP-array$ is defined
7. construct a new header table for Y 's FP-tree from $T.FP-array$
8. **else** construct a new header table from T ;
9. construct Y 's conditional FP-tree T_Y and possibly its FP-array A_Y ;
10. **if** $T_Y \neq \emptyset$
11. call $FPgrowth^*(T_Y)$;
12. **end**

Şekil 2.1 Fp Ağacı Pseudocode

3. İMPLEMENTASYON

Şekil 3.1.' de implematasyon akışına yer verilmiştir.



Şekil 3.1 İmplementasyon akışı

Şekil 3.1. uygulama, elde edilen kullanıcı geri bildirim veri kümesi ile başlar çevrimiçi ve bir dizi nitelikten oluşmaktadır. Bu veri kümesi daha sonra eksiklikler giderilerek ve çözülerek temizlenir . O zaman mevcut FP Büyüme algoritması temiz veri kümesi üzerinde uygulanır ve analiz için gerekli birliktelik kuralları uygulanır

	0100101	0100102	0100103	0100303	0100304	0100500	0100305	0100028	0100045	0100046	0100251	...	7203003	7203005	7204009	7200010	7400114	7600101	7800103	8400228	9990221	9990222	9990223
0	False	False	False	True	True	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False	True
1	False	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	True	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	True	False	False
3	False	False	False	True	True	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	True	False
4	False	False	False	False	False	False	True	False	False	False	False	...	False	False	True	True	True	True	True	True	True	False	False
5	False	True	True	False	False	True	False	False	False	False	False	...	False	False	False	False	False	False	False	False	True	False	False
6	False	True	True	False	False	False	False	True	False	False	False	...	False	False	False	False	False	False	False	False	True	False	False
7	True	True	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	True	False	False
8	False	False	False	False	False	False	False	False	False	False	True	...	False	False	False	False	False	False	False	False	True	False	False
9	False	False	False	True	True	False	False	False	False	True	True	...	True	True	False	False	False	False	False	False	True	True	False
10	False	False	True	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	True	False	False
11	True	True	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	True	False	False

[12 rows x 231 columns]

Şekil 3.2 Model Sonuç

3.1. VERİ SETİ

Veri kümemiz stok_hareketleri.xml bizlere türkiyede bulunan bir anonim şirket tarafından kullanımımıza sunuldu. İçinde bu özellikler vardır. Status : Faturanın türünü belirtir, Fyear : Yıl belirtir, Fmon: Ay belirtir, Evrak_Tarihi : Fatura Tarihi belirtir, Rec_Id : Faturanın kimliği'ni belirtir, Cari_Kod : Kullanıcı ID'sini belirtir, Ürün_kodu : Ürüne özel id belirtir, Ürün_grubu:Ürünün hangi tür gıda grubunda olduğunu belirtir, Miktar :Üründen kaç adet alındığını belirtir. Elimizde ayrıca kendimiz oluşturduğumuz users.xml ve urun.xml veri setleri vardır. users.xml'de user_id : Sisteme giriş yapan kullanıcının kimlik numarasıdır, Password: Geçici bir numerik şifredir, Price_factor:Kullanıcının geçmişte şirketle yaptığı işlemlere itafen şirketin o kullanıcıya sunacağı fiyatları belirleyen bir çarpım katsayısıdır. urunler.xml'de ise ürün_kodu, ürün grubu ve ürünün fiyatı ayrı ayrı listelenmiştir.

3.2 VERİ ÖNİŞLEME

Veri seti içinde status alanı sadece SATIS , Fyear alanı sadece 2022 yılı olan kayıtlar alındı ve gerisi silindi ,bunun nedeni veri setinin boyutunun makinelerimizde çok fazla işlem gücü harcayacağını öngörmemizdir.Veritinde birbirini tekrar eden data kümeleri kaldırıldı . Veritinde özellik alanı boş olan yerler silindi.

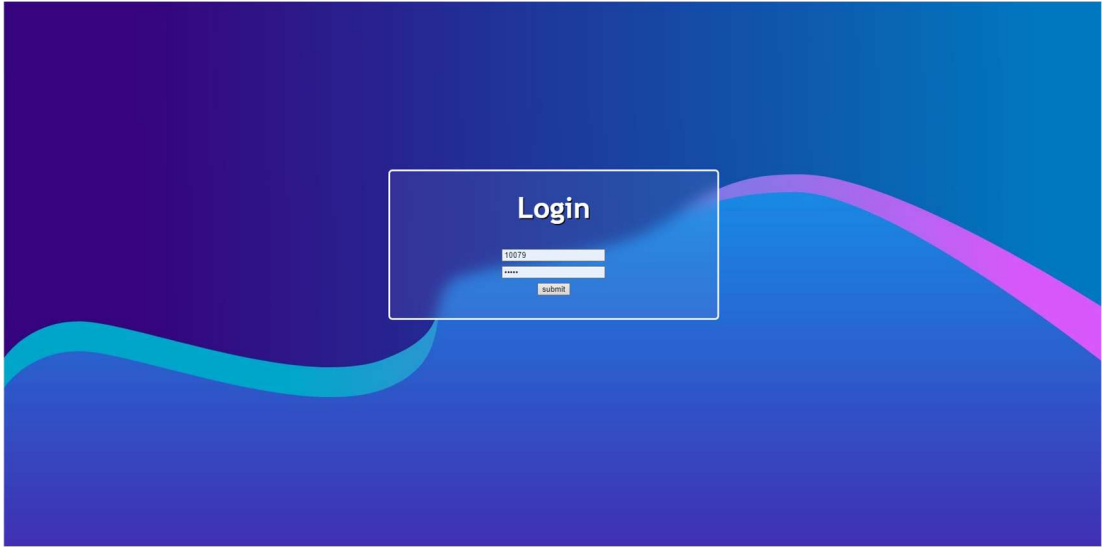
4. PROJE

Amacımız bir tedarikçi firmaya B2C(Business to Customer) Stok yazılım arayüzü geliştirmektir. Bu kapsamda Django ile çalışmaya karar verdik. Bize sağlanan veri setindeki bilgilerin kısıtlı olması ve veri boyutunun fazla olmasından dolayı bu veri setinden iki yeni veri setleri türettik. Bu veri setleri urun.xml ve users.xml. urun.xml ürünlerin, ürün kodu, ürün grubu ve fiyat bilgilerini içerirken, users.xml ise kullanıcı ID'si ve veri setinde bulunmadığından dolayı bizim tarafımızdan atanmış olan şifre ve kişiye özel fiyatlandırma çarpanı bilgilerini içermektedir.

4.1 PROJENİN İŞLEYİŞİ

4.1.1 Giriş Ekranı

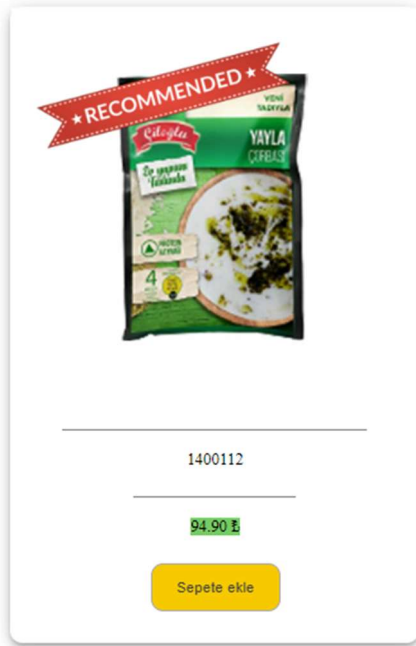
Uygulamaya Login ekranından giriş yapılır. Kullanıcıya ID ve şifre bilgisinin firma tarafından verildikten sonra kullanıcı, kendi ID'si ve şifresiyle sisteme giriş yapar.



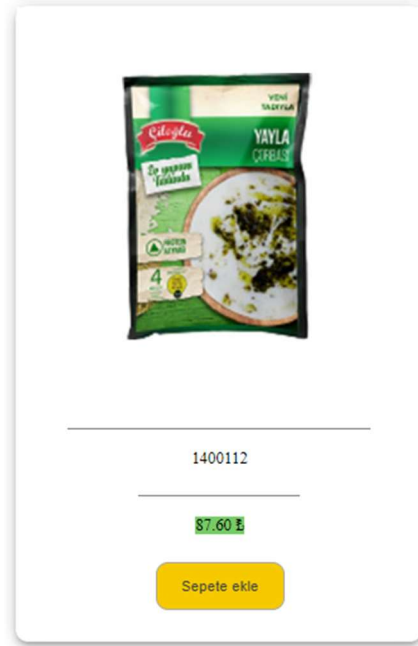
Şekil 4.1 Kullanıcı Giriş Ekranı

4.1.2 Ana Ekran

Giriş yapıldıktan sonra ana ekranda bulunan menüde ilk önce giriş yapan kişinin kullanıcı ID'si yazmakta ve hemen yanında varsayılan olarak “Tüm Ürünler” seçeneği seçili olacak şekilde ayarlanmıştır. Bu sayfada tüm ürünler gösterilmek ile birlikte kullanıcının önceki faturaları, FP-Growth algoritması kullanılarak analiz edilir ve yine kullanıcının en az satın aldığı 5 ürünün görseli üzerine “Recommended” görseli eklenerek en üstte gösterilir. Her biri ürün ayrı bir kutucuk içinde sergilenir. Her kutucuğun içinde ise ürün görseli, ürün kodu, ürün fiyatı ve sepete ekle butonu yer alır. Ürün fiyatı, kullanıcıya özel fiyatlandırma çarpanı ile çarpılarak hesaplanır. Bu değer 0,5 ile 1,5 arasındadır.



Şekil 4.2 Önerilen Ürün



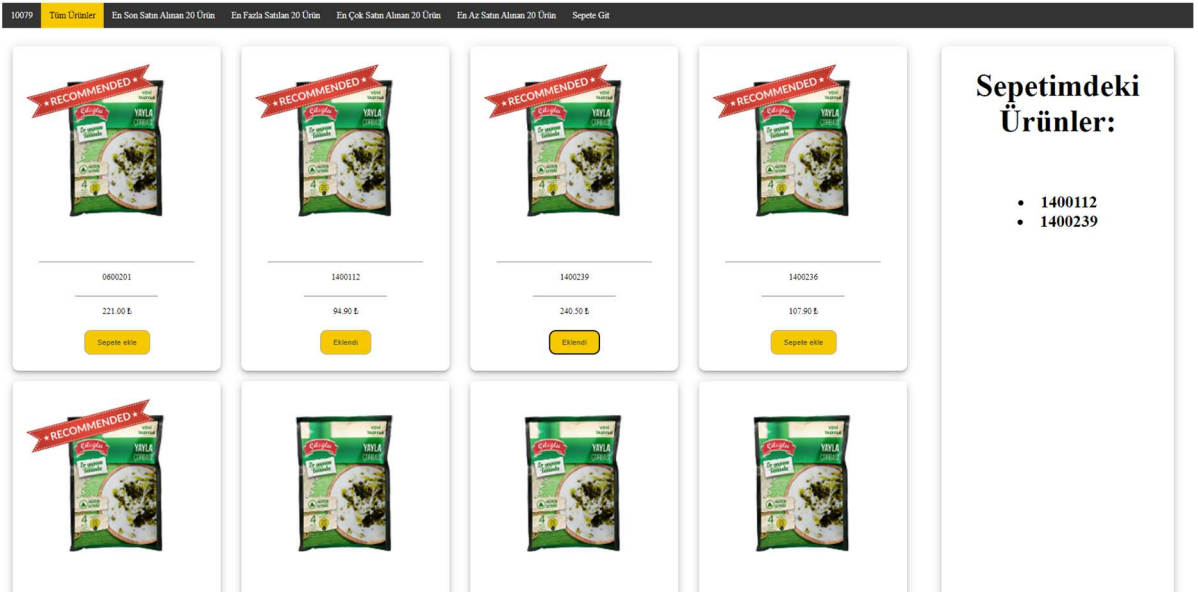
Şekil 4.3 Önerilmeyen Ürün

Şekil 4.2. ve Şekil 4.3. te görüldüğü gibi farklı kullanıcılara farklı fiyatlandırma ve farklı ürün önerileri sunulmaktadır.

Diğer sekmeleri açıklamak gerekirse; “En Son Satın Alınan 20 Ürün” sekmesinde, kullanıcının en son satın aldığı 20 ürünü, “En Fazla Satılan 20 Ürün” sekmesinde, tüm kullanıcılar tarafından en çok satın alınmış 20 ürünü, “En Çok Satın Alınan 20 Ürün” sekmesinde, kullanıcının en çok satın aldığı 20 ürünü, “En Az Satın Alınan 20 Ürün” sekmesinde ise kullanıcının en az satın aldığı 20 ürün listelenmektedir.

Ürün kutucuğu içerisinde bulunan “Sepete Ekle” butonuna basıldığında butonun adı “Eklendi” olarak değiştirilir ve ürün kodu, ekranın sağ tarafında bulunan “Sepetimdeki Ürünler” başlığı altında listelenir. Bu ekranda ürün sepetten çıkarılmak isteniyorsa “Eklendi” butonuna basılır ve ürün sepetten çıkarılır.

Alışverişin tamamlanmasının ardından menüde bulunan en son sekme olan “Sepete Git” seçeneğine tıklanır ve sepet ekranına gidilir.



Şekil 4.4 Ana Ekran





4.1.3 Sepet Ekranı

Bu ekranda, ana ekranda sepete eklenen ürünler gösterilir ve bu ürünlerin adetlerinin girişi burada sağlanır. Ürünün birim fiyatı ile adet bilgisi çarpılır ve tüm sepet toplanarak ekranın sağ tarafında bulunan “Ara Toplam” başlığının altına yazılır. Adet bilgisi her güncellendiğinde “Ara Toplam” bilgisi de güncellenir ve toplam tutar kullanıcıya dinamik olarak sunulur.

Ayrıca kullanıcı isterse bu ekranda da sepetten ürün çıkarabilir. Bu işlem sonrasında da “Ara Toplam” bilgisi güncellenir.

15079

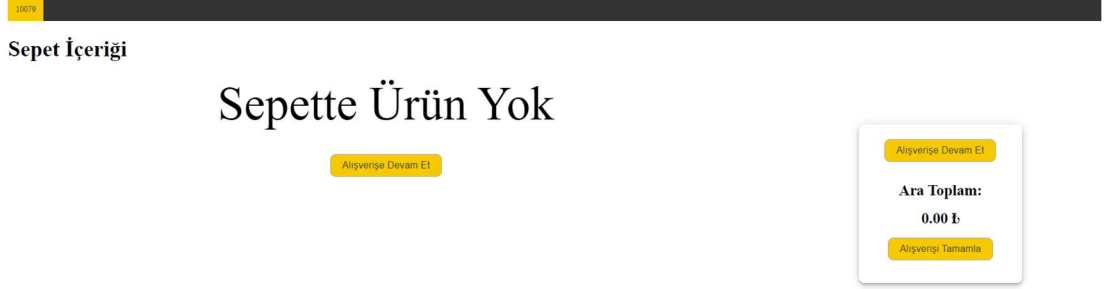
Sepet İçeriği

	1400239 240.50 ₺	Adet: <input type="text" value="1"/>	Sepetten Çıkar
	0600201 221.00 ₺	Adet: <input type="text" value="1"/>	Sepetten Çıkar
	1400112 94.90 ₺	Adet: <input type="text" value="1"/>	Sepetten Çıkar
	1400236 107.90 ₺	Adet: <input type="text" value="1"/>	Sepetten Çıkar

[Alışverişe Devam Et](#)
Ara Toplam:
664.30 ₺
[Alışveriş Tamamla](#)

Şekil 4.5 Sepet Ekranı

Bu ekranda kullanıcı tüm ürünleri sepetten çıkarırsa “Sepette Ürün Yok” yazısıyla ve “Alışverişi Devam Et” butonu ile karşılaşır.



Şekil 4.6 Sepetteki Ürünlerin Çıkarılması Sonrası Sepet Ekranı

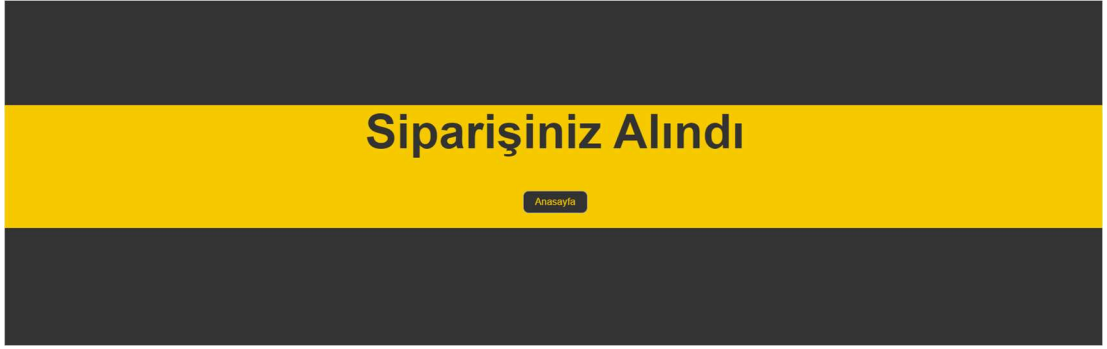
Alışverişi tamamlamak için ise kullanıcının “Alışverişi Tamamla” butonuna basması gerekmektedir. Bu butona basıldığında kullanıcının siparişi siparisler adında bir JSON dosyasına kaydedilir ve kullanıcı “Tamamlandı” ekranına yönlendirilir.

```
{
  "orders": [
    {
      "user_id": "18028",
      "order": [
        {
          "urun_kodu": "7102117",
          "miktar": "1"
        },
        {
          "urun_kodu": "7102109",
          "miktar": "1"
        }
      ]
    }
  ]
}
```

Şekil 4.7 Siparisler.json Dosyası

4.1.4 Sipariş Alındı Ekranı

Bu ekranda kullanıcı ana sayfaya yönlendirilir.



Şekil 4.8 Sipariş Alındı Ekranı

5. KAYNAKÇA

Tünel, mücahit & Gulcu, Saban. (2020). Social Campus Application with Machine Learning for Mobile Devices.

Wang, K., Tang, L., Han, J., Liu, J. (2002). Top Down FP-Growth for Association Rule Mining. In: Chen, MS., Yu, P.S., Liu, B. (eds) Advances in Knowledge Discovery and Data Mining. PAKDD 2002. Lecture Notes in Computer Science(), vol 2336. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-47887-6_34

Bashir Alam. (2023, January 25). *Implementation of FP-growth algorithm using Python 2023*. Hands-On.Cloud. <https://hands-on.cloud/implementation-of-fp-growth-algorithm-using-python/>

Goethals, Bart & Nijssen, Siegfried. (2005). Knowledge Discovery and Data Mining Edited.

Sidhu, S., Meena, U. K., Nawani, A., Gupta, H., & Thakur, N. (2014). FP Growth Algorithm Implementation. Retrieved from <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=ef40f70587ba1c653389c08cd21e96e387c5e6b8>

Raschka, S. (n.d.). *Fpgrowth*. mlxtend. https://rasbt.github.io/mlxtend/user_guide/frequent_patterns/fpgrowth/